Handbook of

Discrete and Computational Geometry

The CRC Press Series on

DISCRETE MATHEMATICS AND ITS APPLICATIONS

Series Editor

Kenneth H. Rosen, Ph.D.

AT&T Bell Laboratories

Charles J. Colbourn and Jeffrey H. Dinitz,
The CRC Handbook of Combinatorial Designs

Steven Furino, Ying Miao, and Jianxing Yin,
Frames and Resolvable Designs: Uses, Constructions, and Existence

Daryl D. Harms, Miroslav Kraetzl, Charles J. Colbourn, and John S. Devitt,
Network Reliability: Experiments with A Symbolic Algebra Environment

Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone,
Handbook of Applied Cryptography

Richard A. Mollin, Quadratics

Douglas R. Stinson, Cryptography: Theory and Practice

Handbook of

Discrete and Computational Geometry

edited by

Jacob E. Goodman
City College, CUNY

Joseph O'Rourke
Smith College

CRC Press
Boca Raton New York

Library of Congress Cataloging-in-Publication Data

Catalog record is available from the Library of Congress.

International Standard Book Number 0-8493-8524-5
Printed in the United States of America 3 4 5 6 7 8 9 0
Printed on acid-free paper

ADVISORY EDITORIAL BOARD

PREFACE

While books and journals of high quality have proliferated in discrete and computational geometry during recent years, there has been to date no single reference work fully accessible to the nonspecialist as well as to the specialist, covering all the major aspects of both fields. The *Handbook of Discrete and Computational Geometry* is intended to do exactly that: to make the most important results and methods in these areas of geometry readily accessible to those who use them in their everyday work, both in the academic world—as researchers in mathematics and computer science—and in the professional world—as practitioners in fields as diverse as operations research, molecular biology, and robotics.

A significant part of the growth that discrete mathematics as a whole has experienced in recent years has consisted of a substantial development in discrete geometry. This has been fueled partly by the advent of powerful computers and by the recent explosion of activity in the relatively young field of computational geometry. This synthesis between discrete and computational geometry, in which the methods and insights of each field have stimulated new understanding of the other, lies at the heart of this Handbook.

The phrase "discrete geometry," which at one time stood mainly for the areas of packing, covering, and tiling, has gradually grown to include in addition such areas as combinatorial geometry, convex polytopes, and arrangements of points, lines, planes, circles, and other geometric objects in the plane and in higher dimensions. Similarly, "computational geometry," which referred not long ago to simply the design and analysis of geometric algorithms, has in recent years broadened its scope, and now means the study of geometric problems from a computational point of view, including also computational convexity, computational topology, and questions involving the combinatorial complexity of arrangements and polyhedra. It is clear from this that there is now a significant overlap between these two fields, and in fact this overlap has become one of practice as well, as mathematicians and computer scientists have found themselves working on the same geometric problems and have forged successful collaborations as a result.

At the same time, a growing list of areas in which the results of this work are applicable has been developing. It includes areas as widely divergent as engineering, crystallography, computer-aided design, manufacturing, operations research, geographic information systems, robotics, error-correcting codes, tomography, geometric modeling, computer graphics, combinatorial optimization, computer vision, pattern recognition, and solid modeling.

With this in mind, it has become clear that a handbook encompassing the most important results of discrete and computational geometry would benefit not only the workers in these two fields, or in related areas such as combinatorics, graph theory, geometric probability, and real algebraic geometry, but also the *users* of this body of results, both industrial and academic. This Handbook is designed to fill that role. We believe it will prove an indispensable working tool both for researchers in geometry and geometric computing and for professionals who use geometric tools in their work.

The Handbook covers a broad range of topics in both discrete and computational geometry, as well as in a number of applied areas. These include geometric data structures, polytopes and polyhedra, convex hull and triangulation algorithms, packing and covering, Voronoi diagrams, combinatorial geometric questions, com-

putational convexity, shortest paths and networks, computational real algebraic geometry, geometric arrangements and their complexity, geometric reconstruction problems, randomization and de-randomization techniques, ray shooting, parallel computation in geometry, oriented matroids, computational topology, mathematical programming, motion planning, sphere packing, computer graphics, robotics, crystallography, and many others. A final chapter is devoted to a list of available software. Results are presented in the form of theorems, algorithms, and tables, with every technical term carefully defined in a glossary that precedes the section in which the term is first used. There are numerous examples and figures to illustrate the ideas discussed, as well as a large number of unsolved problems.

The main body of the volume is divided into six parts. The first two, on combinatorial and discrete geometry and on polytopes and polyhedra, deal with fundamental geometric objects such as planar arrangements, lattices, and convex polytopes. The next section, on algorithms and geometric complexity, discusses these basic geometric objects from a computational point of view. The fourth and fifth sections, on data structures and computational techniques, discuss various computational methods that cut across the spectrum of geometric objects, such as randomization and de-randomization, and parallel algorithms in geometry, as well as efficient data structures for searching and for point location. The sixth section, which is the longest in the volume, contains chapters on fourteen applications areas of both discrete and computational geometry, including low-dimensional linear programming, combinatorial optimization, motion planning, robotics, computer graphics, pattern recognition, graph drawing, splines, manufacturing, solid modeling, rigidity of frameworks, scene analysis, error-correcting codes, and crystallography. It concludes with a fifteenth chapter, an up-to-the-minute compilation of available software relating to the various areas covered in the volume. A comprehensive index follows, which includes proper names as well as all of the terms defined in the main body of the Handbook.

A word about references. Because it would have been prohibitive to provide complete references to all of the many thousands of results included in the Handbook, we have to a large extent restricted ourselves to references for either the most important results, or for those too recent to have been included in earlier survey books or articles; for the rest we have provided annotated references to easily accessible surveys of the individual subjects covered in the Handbook, which themselves contain extensive bibliographies. In this way, the reader who wishes to pursue an older result to its source will be able to do so.

On behalf of the sixty-one contributors and ourselves, we would like to express our appreciation to all those whose comments were of great value to the authors of the various chapters: Pankaj K. Agarwal, Boris Aronov, Noga Alon, Saugata Basu, Margaret Bayer, Louis Billera, Martin Blümlinger, Jürgen Bokowski, B.F. Caviness, Bernard Chazelle, Danny Chen, Xiangping Chen, Yi-Jen Chiang, Edmund M. Clarke, Kenneth Clarkson, Robert Connelly, Henry Crapo, Isabel Cruz, Mark de Berg, Jesús de Loera, Giuseppe Di Battista, Michael Drmota, Peter Eades, Jürgen Eckhoff, Noam D. Elkies, Eva Maria Feichtner, Ioannis Fudos, Branko Grünbaum, Dan Halperin, Eszter Hargittai, Ulli Hund, Jürg Hüsler, Peter Johansson, Norman Johnson, Amy Josefczyk, Gil Kalai, Gyula Károlyi, Kevin Klenk, Włodzimierz Kuperberg, Endre Makai, Jr., Jiří Matoušek, Peter McMullen, Hans Melissen, Bengt Nilsson, Michel Pocchiola, Richard Pollack, Jörg Rambau, Jürgen Richter-Gebert, Allen D. Rogers, Marie-Françoise Roy, Egon Schulte, Dana Scott, Jürgen Sellen, Micha Sharir, Peter Shor, Maxim Michailovich Skriganov, Neil J.A. Sloane, Richard

P. Stanley, Géza Tóth, Ioannis Tollis, Laureen Treacy, Alexander Vardy, Gert Vegter, Pamela Vermeer, Siniša Vrećica, Kevin Weiler, Asia Ivić Weiss, Neil White, Chee-Keng Yap, and Günter M. Ziegler.

In addition, we would like to convey our thanks to the editors of CRC Press for having the vision to commission this Handbook as part of their *Discrete Mathematics and Its Applications* series; to the CRC staff, for their help with the various stages of the project; and in particular to Nora Konopka, with whom we found it a pleasure to work from the inception of the volume.

Finally, we want to express our sincere gratitude to our families: Josy, Rachel, and Naomi Goodman, and Marylynn Salmon and Nell and Russell O'Rourke, for their patience and forbearance while we were in the throes of this project.

Jacob E. Goodman
Joseph O'Rourke

TABLE OF CONTENTS

GEOMETRIC DATA STRUCTURES AND SEARCHING

COMPUTATIONAL TECHNIQUES

APPLICATIONS OF DISCRETE AND COMPUTATIONAL GEOMETRY

CONTRIBUTORS

Pankaj K. Agarwal
Computer Science Department
Duke University
Durham, North Carolina 27708
e-mail: pankaj@cs.duke.edu

John Ralph Alexander, Jr.
Department of Mathematics
University of Illinois
Urbana, Illinois 61801
e-mail: jralex@symcom.math.uiuc.edu

Nina Amenta
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, California 94304
e-mail: amenta@parc.xerox.com

Chanderjit L. Bajaj
Computer Science Department
Purdue University
West Lafayette, Indiana 47907
e-mail: bajaj@cs.purdue.edu

Alexander I. Barvinok
Department of Mathematics
University of Michigan
Ann Arbor, Michigan 48109
e-mail: barvinok@math.lsa.umich.edu

József Beck
Department of Mathematics
Rutgers University
New Brunswick, New Jersey 08903
e-mail: jbeck@aramis.rutgers.edu

Marshall Bern
Xerox Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, California 94304
e-mail: bern@parc.xerox.com

Louis J. Billera
Department of Mathematics
Cornell University
Ithaca, New York 14853
e-mail: billera@math.cornell.edu

Anders Björner
Department of Mathematics
Royal Institute of Technology
S-100 44 Stockholm, Sweden
e-mail: bjorner@math.kth.se

Ulrich Brehm
Institut für Geometrie
Technische Universität Dresden
01062 Dresden, Germany
e-mail: brehm@math.tu-dresden.de

William W.L. Chen
School of Mathematics and Physics
Macquarie University
New South Wales 2109, Australia
e-mail: wchen@macadam.mpce.mq.edu.au

David P. Dobkin
Department of Computer Science
Princeton University
Princeton, New Jersey 08544
e-mail: dpd@cs.princeton.edu

Martin Dyer
School of Computer Studies
University of Leeds
Leeds LS2 9JT, United Kingdom
e-mail: dyer@scs.leeds.ac.uk

Susan Evans
Computer Science Department
Purdue University
West Lafayette, Indiana 47907
e-mail: evans@cs.purdue.edu

Gábor Fejes Tóth
Mathematical Institute of the
Hungarian Academy of Sciences
1364 Budapest, Pf. 127, Hungary
e-mail: gfejes@circle.math-inst.hu

Steven Fortune
Bell Laboratories
700 Mountain Ave
Murray Hill, New Jersey 07974
e-mail: sjf@research.bell-labs.com

Jacob E. Goodman
Department of Mathematics
City College, CUNY
New York, New York 10031
e-mail: jegcc@cunyvm.cuny.edu

Michael T. Goodrich
Department of Computer Science
Johns Hopkins University
Baltimore, Maryland 21218
e-mail: goodrich@cs.jhu.edu

Ronald L. Graham
AT&T Research
600 Mountain Avenue
Murray Hill, New Jersey 07974
e-mail: rlg@research.att.com

Peter Gritzmann
Fachbereich IV, Abteilung Mathematik
Universität Trier
54286 Trier, Germany
Email: gritzman@dm1.uni-trier.de

Dan Halperin
Department of Computer Science
Tel Aviv University
Tel Aviv 69978, Israel
e-mail: danha@math.tau.ac.il

Martin Henk
Fachbereich Mathematik, Sekr. 6-1
Technische Universität Berlin
10623 Berlin, Germany
e-mail: henk@math.tu-berlin.de

Christoph M. Hoffmann
Computer Science Department
Purdue University
West Lafayette, Indiana 47907
e-mail: hoffmann@cs.purdue.edu

Ravi Janardan
Department of Computer Science
University of Minnesota
Minneapolis, Minnesota 55455
email: janardan@cs.umn.edu

Gil Kalai
Institute of Mathematics
Hebrew University
Jerusalem, Israel
e-mail: kalai@cs.huji.ac.il

Lydia Kavraki
Department of Computer Science
Rice University
Houston, Texas 77005
e-mail: kavraki@cs.rice.edu

David A. Klarner
Department of Computer Science
University of Nebraska
Lincoln, Nebraska 68588
e-mail: klarner@swine.unl.edu

Victor Klee
Department of Mathematics
University of Washington
Seattle, Washington 98195
e-mail: klee@math.washington.edu

Jean-Claude Latombe
Robotics Laboratory
Department of Computer Science
Stanford University
Stanford, California 94305
e-mail: latombe@cs.stanford.edu

Carl Lee
Department of Mathematics
University of Kentucky
Lexington, Kentucky 40506
e-mail: lee@ms.uky.edu

Nimrod Megiddo
IBM Almaden Research Center
650 Harry Road
San Jose, California 95120
e-mail: megiddo@almaden.ibm.com

Bhubaneswar Mishra
Courant Institute, NYU
251 Mercer street
New York, New York 10012
e-mail: mishra@cs.nyu.edu

Joseph S.B. Mitchell
Applied Mathematics Department
SUNY at Stony Brook
Stony Brook, New York 11794
e-mail: jsbm@ams.sunysb.edu

David Mount
Department of Computer Science
University of Maryland
College Park, Maryland 20742
e-mail: mount@cs.umd.edu

Ketan Mulmuley
Department of Computer Science
The University of Chicago
Chicago, Illinois 60637
e-mail: mulmuley@cs.uchicago.edu

Joseph O'Rourke
Department of Computer Science
Smith College
Northampton, Massachusetts 01063
e-mail: orourke@cs.smith.edu

János Pach
Department of Computer Science
City College, CUNY
New York, New York 10031
e-mail: pach@math-inst.hu

Marco Pellegrini
IMC-CNR
Via Santa Maria 46
Pisa 56126, Italy
e-mail: pellegrini@imc.pi.cnr.it

Jürgen Richter-Gebert
Fachbereich Mathematik, Sekr. 6-1
Technische Universität Berlin
10623 Berlin, Germany
e-mail: richter@math.tu-berlin.de

Jason Rush
Department of Mathematics
University of Washington
Seattle, Washington 98195
e-mail: jar@math.washington.edu

Jeffrey Salowe
Cadence Design Systems, Inc.
555 River Oaks Parkway, MS 2B1
San Jose, California 95134
e-mail: jsalowe@cadence.com

Doris Schattschneider
Department of Mathematics
Moravian College
Bethlehem, Pennsylvania 18018
e-mail: schattdo@moravian.edu

Rolf Schneider
Mathematisches Institut
Albert-Ludwigs-Universität
79104 Freiburg i. Br., Germany
e-mail: rschnei@ruf.uni-freiburg.de

Egon Schulte
Department of Mathematics
Northeastern University
Boston, Massachusetts 02115
e-mail: schulte@neu.edu

Otfried Schwarzkopf
Department of Computer Science
Pohang University of
Science and Technology
Hyoja-Dong, Pohang 790-784
South Korea
email: otfried@postech.ac.kr

Raimund Seidel
Fachbereich 14–Informatik
Universität des Saarlandes
66041 Saarbrücken, Germany
e-mail: seidel@cs.uni-sb.de

Marjorie Senechal
Department of Mathematics
Smith College
Northampton, Massachusetts 01063
e-mail: senechal@minkowski.smith.edu

Micha Sharir
Department of Computer Science
Tel Aviv University
Tel Aviv 69978, Israel
e-mail: sharir@math.tau.ac.il

Steven S. Skiena
Department of Computer Science
SUNY at Stony Brook
Stony Brook, New York 11794
e-mail: skiena@cs.sunysb.edu

Jack Snoeyink
Department of Computer Science
University of British Columbia
Vancouver V6T 1Z4
British Columbia, Canada
e-mail snoeyink@cs.ubc.ca

Subhash Suri
Department of Computer Science
Washington University
Saint Louis, Missouri 63130
e-mail: suri@cs.wustl.edu

Roberto Tamassia
Department of Computer Science
Brown University
Providence, Rhode Island 02912
e-mail: rt@cs.brown.edu

Seth Teller
Synthetic Imagery Group
MIT Laboratory for Computer Science
Cambridge, Massachusetts 02139
e-mail: seth@theory.lcs.mit.edu

Michael J. Todd
School of Operations Research
and Industrial Engineering
Cornell University
Ithaca, New York 14853
e-mail: miketodd@cs.cornell.edu

Godfried T. Toussaint
School of Computer Science
McGill University
Montréal, Québec H3A 2K6, Canada
e-mail: godfried@opus.cs.mcgill.ca

Gert Vegter
Department of Mathematics
and Computer Science
University of Groningen
9700 AV Groningen, The Netherlands
e-mail: gert@cs.rug.nl

Rephael Wenger
Department of Computer Science
Ohio State University
Columbus, Ohio 43210
e-mail: wenger@cis.ohio-state.edu

Neil White
Department of Mathematics
University of Florida
Gainesville, Florida 32611
e-mail: white@math.ufl.edu

Walter Whiteley
Department of Mathematics
and Statistics
York University
North York, Ontario M3J 1P3, Canada
e-mail: whiteley@mathstat.yorku.ca

Tony C. Woo
Department of Industrial Engineering
University of Washington
Seattle, Washington 98195
e-mail: twoo@u.washington.edu

Chee K. Yap
Courant Institute, NYU
251 Mercer Street
New York, New York 10012
e-mail: yap@cs.nyu.edu

Günter M. Ziegler
Fachbereich Mathematik, Sekr. 6-1
Technische Universität Berlin
10623 Berlin, Germany
e-mail: ziegler@math.tu-berlin.de

Rade Živaljević
Mathematicki Institut
Knez Mihailova 35/1
11001 Beograd, Yugoslavia
e-mail: ezivalje@ubbg.etf.bg.ac.yu

COMBINATORIAL
AND
DISCRETE GEOMETRY

1 FINITE POINT CONFIGURATIONS

János Pach

INTRODUCTION

The study of combinatorial properties of finite point configurations is a vast area of research in geometry, whose origins go back at least to the ancient Greeks. Since it includes virtually all problems starting with "consider a set of n points in space," space limitations impose the necessity of making choices. As a result, we will restrict our attention to Euclidean spaces and will discuss problems that we find particularly important. The chapter is partitioned into incidence problems (Section 1.1), metric problems (Section 1.2), and coloring problems (Section 1.3).

1.1 INCIDENCE PROBLEMS

In this section we will be concerned mainly with the structure of incidences between a finite point configuration P and a set of finitely many lines (or, more generally, k-dimensional flats, spheres, etc.). Sometimes this set consists of all lines connecting the elements of P. The prototype of such a question was raised by Sylvester more than one hundred years ago: Is it true that for any configuration of finitely many points in the plane, not all on a line, there is a line passing through exactly two points? The affirmative answer to this question was first given by Gallai. Generalizations for circles and conic sections in place of lines were established by Motzkin and Wilson-Wiseman, respectively.

GLOSSARY

Incidence: A point of configuration P lies on an element of a given collection of lines (k-flats, spheres, etc.).

Simple crossing: A point incident with exactly two elements of a given collection of lines.

Ordinary line: A line passing through exactly two elements of a given point configuration.

Ordinary hyperplane: A $(d-1)$-dimensional flat passing through exactly d elements of a point configuration in Euclidean d-space.

Motzkin hyperplane: A hyperplane whose intersection with a given d-dimensional point configuration lies—with the exception of exactly one point—in a $(d-2)$-dimensional flat.

Regular family of curves: A family Γ of curves in the xy-plane defined in terms of D real parameters satisfying the following properties. There is an integer s such that (a) the dependence of the curves on x, y, and the parameters

is algebraic of degree at most s; (b) no two distinct curves of Γ intersect in more than s points; (c) for any D points of the plane, there are at most s curves in Γ passing through all of them.

Degrees of freedom: The smallest number D of real parameters defining a regular family of curves.

Spanning tree: A tree whose vertex set is a given set of points and whose edges are line segments.

Spanning path: A spanning tree that is a polygonal path.

Convex position: P forms the vertex set of a convex polygon or polytope.

k-set: A k-element subset of P that can be obtained by intersecting P with an open halfspace.

Halving plane: A hyperplane with $\lfloor |P|/2 \rfloor$ points of P on each side.

SYLVESTER-TYPE RESULTS

1. Gallai theorem (dual version): Any set of lines in the plane, not all of which pass through the same point, determines a simple crossing.

2. Motzkin-Hansen theorem: For any finite set of points in Euclidean d-space, not all of which lie on a hyperplane, there exists a Motzkin hyperplane. We obtain as a corollary that n points in d-space, not all of which lie on a hyperplane, determine at least n distinct hyperplanes. (A hyperplane is *determined* by a point set P if its intersection with P is not contained in a $(d-2)$-flat.) Putting the points on two skew lines in 3-space shows that the existence of an ordinary hyperplane cannot be guaranteed for $d > 2$.

   If $n > 8$ is sufficiently large, then any set of n noncocircular points in the plane determines at least $\binom{n-1}{2}$ distinct circles, and this bound is best possible [Ell67]. The number of ordinary circles determined by n noncocircular points is known to be at least $11n(n-1)/247$.

3. Csima-Sawyer theorem: Any set of n noncollinear points in the plane determines at least $6n/13$ ordinary lines ($n > 7$). This bound is sharp for $n = 13$ and false for $n = 7$ (see Figure 1.1.1). In 3-space, any set of n noncoplanar points determines at least $2n/5$ Motzkin hyperplanes.

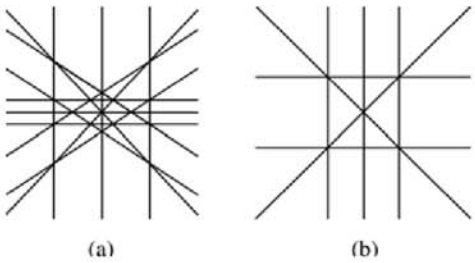


FIGURE 1.1.1
Extremal examples for the (dual) Csima-Sawyer theorem:
(a) 13 *lines (including the line at infinity) determining only* 6 *simple points;*
(b) 7 *lines determining only* 3 *simple points.*

4. Orchard problem: What is the maximum number of collinear triples determined by n points in the plane, no four on a line? There are several construc-

tions showing that this number is at least $n^2/6 - O(n)$, which is asymptotically best possible. (See Figure 1.1.2.)

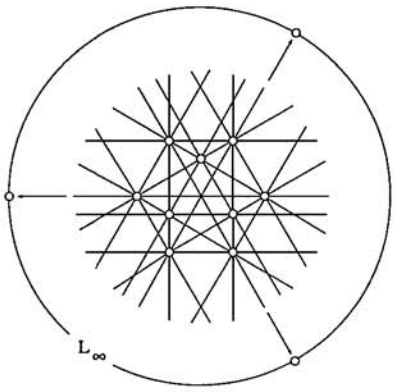


FIGURE 1.1.2
12 *points and* 19 *lines, each passing through exactly* 3 *points.*

5. Dirac's problem: Is it true that—with six exceptions listed in [Grü72]—any set of n points in the plane, not all on a line, has an element incident to at least $n/2$ connecting lines? If true, this result is best possible, as is shown by the example of n points distributed as evenly as possible on two intersecting lines. It is known that there is a positive constant c such that one can find a point incident to at least cn connecting lines. A useful equivalent formulation of this statement is that any set of n points in the plane, no more than $n-k$ of which are on the same line, determines at least $c'kn$ distinct connecting lines, for a suitable constant $c' > 0$. Note that according to the $d = 2$ special case of the Motzkin-Hansen theorem, due to Erdős (see No. 2 above), for $k = 1$ the number of distinct connecting lines is at least n. For $k = 2$, the corresponding bound is $2n - 4$, $(n \geq 10)$.

6. Ungar's theorem: n noncollinear points in the plane always determine at least $2\lfloor n/2 \rfloor$ lines of different slopes (see Figure 1.1.3); this proves Scott's conjecture. Furthermore, any set of n points in the plane, not all on a line, permits a spanning tree, all of whose $n-1$ edges have different slopes [Jam87].

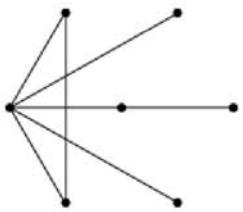


FIGURE 1.1.3
7 *points determining* 6 *distinct slopes.*

UPPER BOUNDS ON THE NUMBER OF INCIDENCES

Given a set P of n points and a family Γ of m curves or surfaces, the number of

incidences between them can be obtained by summing over all $p \in P$ the number of elements of Γ passing through p. If the elements of Γ are taken from a regular family of curves with D degrees of freedom, the maximum number of incidences between P and Γ is $O(n^{D/(2D-1)}m^{(2D-2)/(2D-1)} + n + m)$. In the most important applications, Γ is a family of straight lines or unit circles in the plane ($D = 2$), or it consists of circles of arbitrary radii ($D = 3$). The best upper bounds known for the number of incidences are summarized in Table 1.1.1. It follows from the first line of the table that for any set P of n points in the plane, the number of distinct straight lines containing at least k elements of P is $O(n^2/k^3 + n/k)$, and this bound cannot be improved (Szemerédi-Trotter). In the sixth line of the table, $\beta(n, m)$ is an extremely slowly growing function, which is certainly $o(n^\epsilon m^\epsilon)$ for every $\epsilon > 0$. A collection of spheres in 3-space is said to be in *general position* here if no three of them pass through the same circle.

TABLE 1.1.1 Maximum number of incidences between n points of P and m elements of Γ. [CEG+90]

POINT SET P	FAMILY Γ	BOUND	TIGHT
Planar	lines	$O(n^{2/3}m^{2/3} + n + m)$	yes
Planar	pseudolines	$O(n^{2/3}m^{2/3} + n + m)$	yes
Planar	unit circles	$O(n^{2/3}m^{2/3} + n + m)$	?
Planar	any circles	$O(n^{3/5}m^{4/5} + n + m)$	?
Planar	pseudocircles	$O(n^{3/5}m^{4/5} + n + m)$	?
3-dimensional	spheres	$O(n^{4/7}m^{9/7}\beta(n, m) + n^2)$	?
3-dimensional	spheres in gen. position	$O(n^{3/4}m^{3/4} + n + m)$	?

MIXED PROBLEMS

Many problems about finite point configurations involve some notions that cannot be defined in terms of incidences: convex position, midpoint of a segment, etc. Below we list a few questions of this type. They are discussed in this part of the chapter, and not in Section 1.2 which deals with metric questions, because we can disregard most aspects of the Euclidean metrics in their formulation. For example, convex position can be defined by requiring that some sets should lie on one side of certain hyperplanes. This is essentially equivalent to introducing an order along each straight line.

1. Erdős-Klein-Szekeres problem: What is the maximum number of points that can be chosen in the plane so that no three are on a line and no k are in convex position ($k > 3$)? Denoting this number by $c(k)$, it is known that

$$2^{k-2} \le c(k) \le \binom{2n-4}{n-2}.$$

Let $e(k)$ denote the maximum size of a planar point set P that has no three elements on a line and no k elements that form the vertex set of an "empty"

convex polygon, i.e., a convex k-gon whose interior is disjoint from P. We have $e(3) = 2$, $e(4) = 4$, $e(5) = 9$, and Horton showed that $e(k)$ is infinite for all $k \geq 7$. It is an outstanding open problem to decide whether $e(6)$ is finite.

2. The number of empty k-gons: Let $H_k^d(n)$ $(n \geq k \geq d+1)$ denote the minimum number of k-tuples that induce an empty convex polytope of k vertices in a set of n points in d-space, no $d+1$ of which lie on a hyperplane. Clearly, $H_2^1(n) = n-1$ and $H_k^1(n) = 0$ for $k > 2$. For $k = d+1$, we have

$$\frac{1}{d!} \leq \lim_{n\to\infty} H_k^d(n)/n^d \leq \frac{2}{(d-1)!},$$

[Val95]. For $d = 2$, the best estimates known for $H_k^2 = \lim_{n\to\infty} H_k^2(n)/n^2$ are

$$1 \leq H_3^2 \leq 1.68,\ 1/2 \leq H_4^2 \leq 2.42,\ 0 \leq H_5^2 \leq 1.46,$$

$$0 \leq H_6^2 \leq 1/3,\ H_7^2 = H_8^2 = \ldots = 0.$$

3. The number of k-sets: Let $N_k^d(n)$ denote the maximum number of k-sets in a set of n points in d-space, no $d+1$ of which lie on the same hyperplane. In other words, $N_k^d(n)$ is the maximum number of different ways in which k points of an n-element set can be separated from the others by a hyperplane. It is known that

$$\Omega(n \log k) \leq N_k^2(n) \leq O\left(n\sqrt{k}/\log^* k\right),$$

where $\log^* k$ denotes the iterated logarithm of k. For the number of halving planes, $N_{\lfloor n/2 \rfloor}^3(n) = O(n^{8/3})$, and

$$\Omega(n^{d-1} \log n) \leq N_{\lfloor n/2 \rfloor}^d(n) = o(n^d).$$

4. The number of midpoints: Let $M(n)$ denote the minimum number of different midpoints of the $\binom{n}{2}$ line segments determined by n points in convex position in the plane. One might guess that $M(n) \geq (1 - o(1))\binom{n}{2}$, but it was shown in [EFF91] that

$$\binom{n}{2} - \left\lfloor \frac{n(n+1)(1-e^{-1/2})}{4} \right\rfloor \leq M(n) \leq \binom{n}{2} - \left\lfloor \frac{n^2-2n+12}{20} \right\rfloor.$$

5. Midpoint-free subsets: As a partial answer to a question proposed in [MP], it was proved by V. Bálint et al. that if $m(n)$ denotes the largest number m such that every set of n points in the plane has a midpoint-free subset of size m, then

$$\left\lceil \frac{-1+\sqrt{8n+1}}{2} \right\rceil \leq m(n) \leq o(n)$$

OPEN PROBLEMS

Here we give six problems from the multitude of interesting questions that remain open.

1. Motzkin-Dirac conjecture: Any set of n noncollinear points in the plane determines at least $n/2$ ordinary lines ($n > 13$).

2. Generalized orchard problem (Erdős): What is the maximum number of collinear k-tuples determined by n points in the plane, no $k+1$ of which are on a line ($k \geq 3$)? In particular, show that it is $o(n^2)$ for $k = 4$. The best lower bound known is $\Omega(n^{1+1/(k-2)})$.

3. Maximum independent subset problem (Erdős): Determine the largest number $\alpha(n)$ such that any set of n points in the plane, no four on a line, has an $\alpha(n)$-element subset with no collinear triples. Füredi has shown that $\Omega(\sqrt{n \log n}) \leq \alpha(n) \leq o(n)$.

4. Slope problem (Jamison): Is it true that every set of n points in the plane, not all on a line, permits a spanning path, all of whose $n-1$ edges have different slopes?

5. Empty triangle problem (Bárány): Is it true that every set of n points in the plane, no three on a line, determines at least $t(n)$ empty triangles that share a side, where $t(n)$ is a suitable function tending to infinity?

6. Balanced partition problem (Kupitz): Does there exist an integer k with the property that for every planar point set P, there is a connecting line such that the difference between the number of elements of P on its left side and right side does not exceed k? Several examples show that this assertion is not true with $k = 1$.

1.2 METRIC PROBLEMS

The systematic study of the distribution of the $\binom{n}{2}$ distances determined by n points was initiated by Erdős in 1946. Given a point configuration $P = \{p_1, p_2, \ldots, p_n\}$, let $g(P)$ denote the number of distinct distances determined by P, and let $f(P)$ denote the number of times that the unit distance occurs between two elements of P. That is, $f(P)$ is the number of pairs $p_i p_j$ $(i<j)$ such that $|p_i - p_j| = 1$. What is the minimum of $g(P)$ and what is the maximum of $f(P)$ over all n-element subsets of Euclidean d-space? These questions have raised deep number-theoretic and combinatorial problems, and have contributed richly to many recent developments in these fields.

GLOSSARY

Unit distance graph: A graph whose vertex set is a given point configuration P, in which two points are connected by an edge if and only if their distance is one.

Diameter: The maximum distance between two points of P.

General position in the plane: No three points of P are on a line, and no four on a circle.

Separated set: The distance between any two elements is at least one.

Nearest neighbor of $p \in P$: A point $q \in P$, whose distance from p is minimum.

Farthest neighbor of $p \in P$: A point $q \in P$, whose distance from p is maximum.

Homothetic sets: Similar sets in parallel position.

REPEATED DISTANCES

Extremal graph theory has played an important role in this area. For example, it is easy to see that the unit distance graph assigned to an n-element planar point set P cannot contain $K_{2,3}$, a complete bipartite graph with 2 and 3 vertices in its classes. Thus, by a well-known graph-theoretic result, $f(P)$, the number of edges in this graph, is at most $O(n^{3/2})$. This bound can be improved to $O(n^{4/3})$ by using more sophisticated combinatorial techniques (apply line 3 of Table 1.1.1 with $m = n$); but we are still far from knowing what the best upper bound is.

In Table 1.2.1, we summarize the best currently known estimates on the maximum number of times the unit distance can occur among n points in the plane, under various restrictions on their position. In the first line of the table—and throughout this chapter—c denotes (unrelated) positive constants. The second and third lines show how many times the minimum distance and the maximum distance, resp., can occur among n arbitrary points in the plane. Table 1.2.2 contains some analogous results in higher dimensions. In the first line, $\beta(n)$ is an extremely slowly growing function, closely related to the functional inverse of the Ackermann function.

TABLE 1.2.1 Estimates for the maximum number of unit distances determined by an n-element planar point set P.

POINT SET P	LOWER BOUND	UPPER BOUND	SOURCE
Arbitrary	$n^{1+c/\log\log n}$	$O(n^{4/3})$	Erdős, Spencer et al.
Separated	$\lfloor 3n - \sqrt{12n-3} \rfloor$	$\lfloor 3n - \sqrt{12n-3} \rfloor$	Reutter, Harborth
Of diameter 1	n	n	Hopf-Pannwitz
In convex position	$2n-7$	$O(n \log n)$	Edelsbrunner-Hajnal, Füredi
No 3 collinear	$\Omega(n \log n)$	$O(n^{4/3})$	Kárteszi
Separated, no 3 coll.	$(2+5/16-o(1))n$	$(2+3/7)n$	[Tót95]

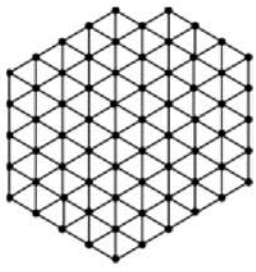


FIGURE 1.2.1
A separated point set with $\lfloor 3n - (12n-3)^{1/2} \rfloor$ unit distances ($n = 69$). All such sets have been characterized by Kupitz.

TABLE 1.2.2 Estimates for the maximum number of unit distances determined by an n-element point set P in d-space.

POINT SET P	LOWER BOUND	UPPER BOUND	SOURCE
$d=3$, arbitrary	$\Omega(n^{4/3}\log\log n)$	$O(n^{3/2}\beta(n))$	Clarkson et al.
$d=3$, separated	$6n-O(n^{2/3})$	$6n-\Omega(n^{2/3})$	Newton
$d=3$, diameter 1	$2n-2$	$2n-2$	Grünbaum, Heppes
$d=3$, on sphere (rad. $1/\sqrt{2}$)	$\Omega(n^{4/3})$	$O(n^{4/3})$	Erdős-Hickerson-Pach
$d=3$, on sphere (rad. $r\neq 1/\sqrt{2}$)	$\Omega(n\log^* n)$	$O(n^{4/3})$	Erdős-Hickerson-Pach
$d>3$ even, arb.	$\frac{n^2}{2}\left(1-\frac{1}{\lfloor d/2\rfloor}\right)+n-O(d)$	$\frac{n^2}{2}\left(1-\frac{1}{\lfloor d/2\rfloor}\right)+n-\Omega(d)$	Erdős
$d>3$ odd, arb.	$\frac{n^2}{2}\left(1-\frac{1}{\lfloor d/2\rfloor}\right)+\Omega(n^{4/3})$	$\frac{n^2}{2}\left(1-\frac{1}{\lfloor d/2\rfloor}\right)+O(n^{4/3})$	Erdős-Pach

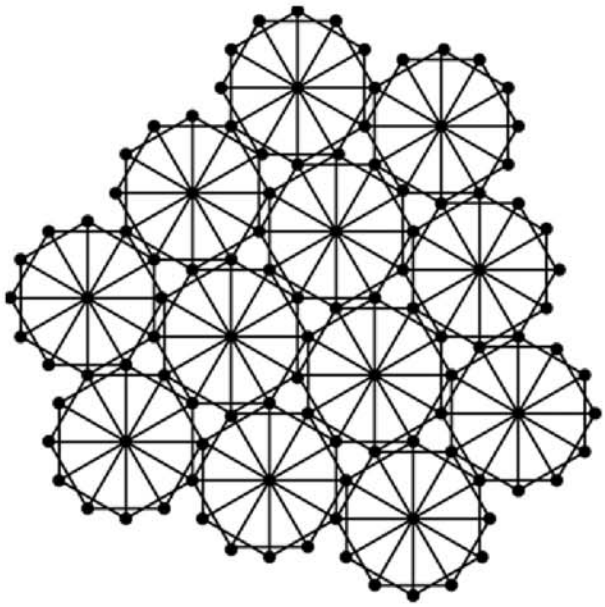


FIGURE 1.2.2
n points, among which the second-smallest distance occurs $(\frac{24}{7}+o(1))n$ times.

The second line of Table 1.2.2 can be extended by showing that the smallest distance cannot occur more than $3n-2k+4$ times between points of an n-element set in the plane whose convex hull has k vertices. The maximum number of occurrences of the second-smallest and second-largest distance is $(24/7+o(1))n$ and $3n/2$ (if n is even), respectively (Brass, Vesztergombi).

Given any point configuration P, let $\Phi(P)$ denote the sum of the numbers of farthest neighbors for every element $p\in P$. Table 1.2.3 contains tight upper bounds on $\Phi(P)$ in the plane and in 3-space, and asymptotically tight ones for higher dimensions [ES89], [Csi95], [EP90].

TABLE 1.2.3 Upper bounds on $\Phi(P)$, the total number of farthest neighbors of all points of an n-element set P.

POINT SET P	UPPER BOUND
Planar, n is even	$3n-3$
Planar, n is odd	$3n-4$
Planar, in convex position	$2n$
3-dimensional, $n \equiv 0 \pmod 2$	$n^2/4+3n/2+3$
3-dimensional, $n \equiv 1 \pmod 4$	$n^2/4+3n/2+9/4$
3-dimensional, $n \equiv 3 \pmod 4$	$n^2/4+3n/2+13/4$
d-dimensional ($d>3$)	$n^2(1-1/\lfloor d/2\rfloor+o(1))$

DISTINCT DISTANCES

It is obvious that if all distances between pairs of points of a d-dimensional set P are the same, then $|P| \le d+1$. If P determines at most g distinct distances, we have that $|P| \le \binom{d+g}{d}$; see [BBS83]. This implies that if d is fixed and n tends to infinity, then the minimum number of distinct distances determined by n points in d-space is at least $\Omega(n^{1/d})$. Denoting this minimum by $g_d(n)$, for $d \ge 3$ we have the following results:

$$\Omega(n^{1/(d-1)}/2^{c\alpha^2(n)}) \le g_d(n) \le O(n^{2/d}),$$

where $\alpha(n)$ is the (extremely slowly growing) functional inverse of Ackermann's function. In Table 1.2.4, we list some lower and upper bounds on the minimum number of distinct distances determined by an n-element point set P, under various assumptions on its structure.

TABLE 1.2.4 Estimates for the minimum number of distinct distances determined by an n-element point set P in the plane.

POINT SET P	LOWER BOUND	UPPER BOUND	SOURCE
Arbitrary	$\Omega(n^{4/5})$	$O(n/\sqrt{\log n})$	Székely [Szé95]
In convex position	$\lfloor n/2\rfloor$	$\lfloor n/2\rfloor$	Altman
No 3 collinear	$\lceil (n-1)/3\rceil$	$\lfloor n/2\rfloor$	Szemerédi
In general position	$\Omega(n)$	$O(n^{1+c/\sqrt{\log n}})$	Erdős, Füredi et al.

RELATED RESULTS

1. Integer distances: There are arbitrarily large, noncollinear finite point sets in the plane such that all distances determined by them are integers, but there exists no infinite set with this property.

2. Generic subsets: Any set of n points in the plane contains $\Omega(n^{1/4})$ points such that all distances between them are distinct. This bound could perhaps be improved to about $n^{1/3}$; see [LT95].

3. Borsuk's problem: It was conjectured that every (finite) d-dimensional point set P can be partitioned into $d+1$ parts of smaller diameter. It follows from the results quoted in the third lines of Tables 1.2.1 and 1.2.2 that this is true for $d = 2$ and 3. Surprisingly, Kahn and Kalai proved that there exist sets P that cannot be partitioned into fewer than $(1.2)^{\sqrt{d}}$ parts of smaller diameter. In particular, the conjecture is false for $d = 946$. On the other hand, it is known that for large d, every d-dimensional set can be partitioned into $(\sqrt{3/2} + o(1))^d$ parts of smaller diameter [Sch88].

4. Nearly equal distances: Two numbers are said to be nearly equal if their difference is at most one. If n is sufficiently large, then the maximum number of times that nearly the same distance occurs among n separated points in the plane is $\lfloor n^2/4 \rfloor$. The maximum number of pairs in a separated set of n points in the plane, whose distance is nearly equal to any one of k arbitrarily chosen numbers, is $\frac{n^2}{2}(1 - \frac{1}{k+1} + o(1))$, as n tends to infinity [EMP93].

5. Repeated angles: In an n-element planar point set, the maximum number of noncollinear triples that determine the same angle is $O(n^2 \log n)$, and this bound is asymptotically tight (Pach-Sharir). The corresponding maximum in 3-space is at most $O(n^{8/3})$, but in 4-space the angle $\pi/2$ can occur $\Omega(n^3)$ times (Croft, Purdy).

6. Repeated triangles: Let $t_d(n)$ denote the maximum number of triples in an n-element point set in d-space that induce a unit area triangle. It is known that $\Omega(n^2 \log\log n) \leq t_2(n) \leq O(n^{7/3})$, $t_5(n) = o(n^3)$, and $t_6(n) = \Theta(n^3)$ (Pach-Sharir, Purdy). In the plane, the maximum number of triples that determine a triangle of unit perimeter, or an isosceles triangle, is also $O(n^{7/3})$.

7. Similar triangles: There exists a positive constant c such that for any triangle T and any $n \geq 3$, there is an n-element point set in the plane with at least cn^2 triples that induce triangles similar to T. For most quadrilaterals Q, the maximum number of 4-tuples of an n-element set that induce quadrilaterals similar to Q is $o(n^2)$. The maximum number of pairwise homothetic triples in a set of n points in the plane is $O(n^{3/2})$, and this bound is asymptotically tight [EE94].

CONJECTURES OF ERDŐS

1. The number of times the unit distance can occur among n points in the plane does not exceed $n^{1+c/\log\log n}$.

2. Any set of n points in the plane determines at least $\Omega(n/\sqrt{\log n})$ distinct distances.

3. Any set of n points in convex position in the plane has a point from which there are at least $\lfloor n/2 \rfloor$ distinct distances.

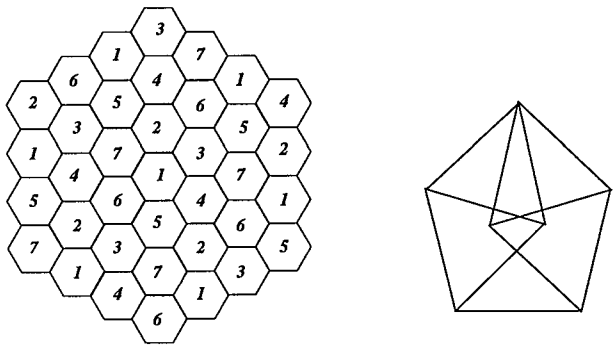
4. There is an integer $k \geq 4$ such that any finite set in convex position in the plane has a point from which there are no k points at the same distance.

5. Any set of n points in the plane, not all on a line, contains at least $n - 2$ triples that determine distinct angles (Corrádi, Erdős, Hajnal).

6. The diameter of any set of n points in the plane with the property that the set of all distances determined by them is separated (on the line) is at least $\Omega(n)$. Perhaps it is at least $n - 1$, with equality when the points are collinear.

1.3 COLORING PROBLEMS

If we partition a space into a small number of parts (i.e., we color its points with a small number of colors), at least one of these parts must contain certain "unavoidable" point configurations. In the simplest case, the configuration consists of a pair of points at a given distance. The prototype of such a question is the Hadwiger-Nelson problem: What is the minimum number of colors needed for coloring the plane so that no two points at unit distance receive the same color? The answer is known to be between 4 and 7.



(i) (ii)

FIGURE 1.3.1
The chromatic number of the plane is (i) *at most* 7 *and* (ii) *at least* 4.

GLOSSARY

Chromatic number of a graph: The minimum number of colors, $\chi(G)$, needed to color all the vertices of G so that no two vertices of the same color are adjacent.

List-chromatic number of a graph: The minimum number k such that for any assignment of a list of k colors to every vertex of the graph, for each vertex it is possible to choose a single color from its list so that no two vertices adjacent to each other receive the same color.

Chromatic number of a metric space: The chromatic number of the unit distance graph of the space, i.e., the minimum number of colors needed to color all points of the space so that no two points of the same color are at unit distance.

Polychromatic number of metric space: The minimum number of colors, χ, needed to color all points of the space so that for each color class C_i, there is a distance d_i such that no two points of C_i are at distance d_i. A sequence of "forbidden" distances, $(d_1, \ldots, d_\chi)$, is called a ***type*** of the coloring. (The same coloring may have several types.)

Girth of a graph: The length of the shortest cycle in the graph.

A point configuration P is ***k-Ramsey*** in d-space if, for any coloring of the points of d-space with k colors, at least one of the color classes contains a congruent copy of P.

A point configuration P is ***Ramsey*** if, for every k, there exists $d(k)$ such that P is k-Ramsey in $d(k)$-space.

Brick: The vertex set of a right parallepiped.

FORBIDDEN DISTANCES

Table 1.3.1 contains the best bounds we know for the chromatic numbers of various spaces. All lower bounds can be established by showing that the corresponding unit distance graphs have some *finite* subgraphs of large chromatic number. $S^{d-1}(r)$ denotes the sphere of radius r in d-space, where the distance between two points is the length of the chord connecting them.

TABLE 1.3.1 Estimates for the chromatic numbers of metric spaces.

SPACE	LOWER BOUND	UPPER BOUND	SOURCE
Line	2	2	
Plane	4	7	Nelson, Isbell
Rational points of plane	2	2	Woodall
3-space	5	21	Raĭskiĭ
Rational points of 3-space	2	2	Benda, Perles
$S^2(r), \frac{1}{2} \le r \le \frac{\sqrt{3-\sqrt{3}}}{2}$	3	4	Simmons
$S^2(r), \frac{\sqrt{3-\sqrt{3}}}{2} \le r \le \frac{1}{\sqrt{3}}$	3	5	Straus
$S^2(r), r \ge \frac{1}{\sqrt{3}}$	4	7	Simmons
$S^2\left(\frac{1}{\sqrt{2}}\right)$	4	4	Simmons
Rational points of 4-space	4	4	Benda, Perles
Rational points of 5-space	6	?	Chilakamarri
d-space	$(1+o(1))(1.2)^d$	$(3+o(1))^d$	Frankl-Wilson, Larman-Rogers
$S^{d-1}(r), r \ge \frac{1}{2}$	d	?	Lovász

Next we list several problems and results strongly related to the Hadwiger-Nelson problem (quoted in the introduction to this section).

1. Polychromatic number: Stechkin and Woodall showed that the polychromatic number of the plane is between 4 and 6. It is known that for any $r \in [\sqrt{2} - 1, 1/\sqrt{5}]$, there is a coloring of type $(1, 1, 1, 1, 1, r)$ [Soi94]. However, the list-chromatic number of the unit distance graph of the plane, which is at least as large as its polychromatic number, is infinite.

2. Dense sets realizing no unit distance: The *lower* (resp. *upper*) *density* of an unbounded set in the plane is the lim inf (resp. lim sup) of the ratio of the Lebesgue measure of its intersection with a disk of radius r around the origin to $r^2\pi$, as $r \to \infty$. If these two numbers coincide, their common value is called the ***density*** of the set. Let δ^d denote the maximum density of a planar set, no pair of points of which is at unit distance. Croft showed that $0.2293 \leq \delta^2 \leq 0.2857$.

3. The graph of large distances: Let $G_i(P)$ denote the graph whose vertex set is a finite point set P, with two vertices connected by an edge if and only if their distance is one of the i largest distances determined by P. In the plane, $\chi(G_1(P)) \leq 3$ for every P; see Borsuk's problem in the preceding section. It is also known that for any finite planar set, $G_i(P)$ has a vertex with fewer than $3i$ neighbors (Erdős-Lovász-Vesztergombi). Thus, $G_i(P)$ has fewer than $3in$ edges, and its chromatic number is at most $3i$. However, if $n > ci^2$ for a suitable constant $c > 0$, we have $\chi(G_i(P)) \leq 7$.

EUCLIDEAN RAMSEY THEORY

According to an old result of Gallai, for any finite d-dimensional point configuration P and for any coloring of d-space with finitely many colors, at least one of the color classes will contain a homothetic copy of P. The corresponding statement is false if, instead of a homothet, we want to find a *translate*, or even a *congruent copy*, of P. Nevertheless, for some special configurations, one can establish interesting positive results, provided that we color a sufficiently high-dimensional space with a sufficiently small number of colors. The Hadwiger-Nelson-type results discussed in the preceding subsection can also be regarded as very special cases of this problem, in which P consists of only two points. The field, known as "Euclidean Ramsey theory", was started by a series of papers by Erdős, Graham, Montgomery, Rothschild, Spencer, and Straus.

For details, see Chapter 8 of this Handbook.

OPEN PROBLEMS

1. (Erdős, Simmons) Is it true that the chromatic number of $S^{d-1}(r)$, the sphere of radius r in d-space, is equal to $d+1$, for every $r > 1/2$? In particular, does this hold for $d = 3$ and $r = 1/\sqrt{3}$?

2. (Erdős) Does there exist an integer g such that the chromatic number of any unit distance graph in the plane whose girth is at least g does not exceed 3? It is known that if such an integer exists, it must be at least 5 [Wor79].

3. (Sachs) What is the minimum number of colors, $\chi(d)$, sufficient to color any system of nonoverlapping unit balls in d-space so that no two balls that are tangent to each other receive the same color? Equivalently, what is the maximum chromatic number of a unit distance graph induced by a d-dimensional separated point set? It is easy to see that $\chi(2) = 4$, and we also know that $5 \leq \chi(3) \leq 9$.

4. (Ringel) Does there exist any finite upper bound on the number of colors needed to color any system of (possibly overlapping) disks (of not necessarily equal radii) in the plane so that no two disks that are tangent to each other receive the same color, provided that no three disks touch one another at the same point? If such a number exists, it must be at least 5.

5. (Graham) Is it true that any 3-element point set P that does not induce an equilateral triangle is 2-Ramsey in the plane? This is known to be false for equilateral triangles, and correct for right triangles (Shader). Is every 3-element point set P 3-Ramsey in 3-space? The answer is again in the affirmative for right triangles (Bóna and Tóth).

1.4 SOURCES AND RELATED MATERIAL

SURVEYS

All results not given an explicit reference above may be traced in these surveys.

[PA95]: A monograph devoted to combinatorial geometry.

[Pac93]: A collection of essays covering a large area of discrete and computational geometry, mostly of some combinatorial flavor.

[HDK64]: A classical treatise of problems and exercises in combinatorial geometry, complete with solutions.

[KW91]: A collection of beautiful open questions in geometry and number theory, together with some partial answers organized into challenging exercises.

[EP95]: A survey full of original problems raised by the "founding father" of combinatorial geometry.

[JT95]: A collection of more than two hundred unsolved problems about graph colorings, with an extensive list of refererences to related results.

[Grü72]: A monograph containing many results and conjectures on configurations and arrangements.

RELATED CHAPTERS

Chapter 4: Helly-type theorems and geometric transversals
Chapter 5: Pseudoline arrangements

Chapter 8: Euclidean Ramsey theory
Chapter 10: Geometric discrepancy theory and uniform distribution
Chapter 11: Topological methods
Chapter 21: Arrangements

REFERENCES

[BBS83] Ei. Bannai, Et. Bannai, and D. Stanton. An upper bound on the cardinality of an s-distance subset in real Euclidean space II. *Combinatorica*, 3:147–152, 1983.

[CEG+90] K. Clarkson, H. Edelsbrunner, L. Guibas, M. Sharir, and E. Welzl. Combinatorial complexity bounds for arrangements of curves and surfaces. *Discrete Comput. Geom.*, 5:99–160, 1990.

[Csi95] G. Csizmadia. Furthest neighbors in space. Report 95-23, Math. Inst., Hungarian Acad. Sciences, Budapest, 1995. *Discrete Math.*, to appear.

[EE94] G. Elekes and P. Erdős. Similar configurations and pseudogrids. In K. Böröczky and G. Fejes Tóth, editors, *Intuitive Geometry*, pages 85–104. North-Holland, Amsterdam, 1994.

[EFF91] P. Erdős, P. Fishburn, and Z. Füredi. Midpoints of diagonals of convex n-gons. *SIAM J. Discrete Math.*, 4:329–341, 1991.

[Ell67] P.D.T.A. Elliott. On the number of circles determined by n points. *Acta Math. Acad. Sci. Hungar.*, 18:181–188, 1967.

[EMP93] P. Erdős, E. Makai, and J. Pach. Nearly equal distances in the plane. *Combin. Probab. Comput.*, 2:401–408, 1993.

[EP90] P. Erdős and J. Pach. Variations on the theme of repeated distances. *Combinatorica*, 10:261–269, 1990.

[EP95] P. Erdős and G. Purdy. Combinatorics of geometric configurations. In R. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, pages 809–874. North Holland, Amsterdam, 1995.

[ES89] H. Edelsbrunner and S. Skiena. The maximum number of unit distances in a convex n-gon. *Amer. Math. Monthly*, 96:614–618, 1989.

[Grü72] B. Grünbaum. *Arrangements and Spreads, CBMS Regional Conf. Ser. in Math., No. 10.* Amer. Math. Soc., Providence, 1972.

[HDK64] H. Hadwiger, H. Debrunner, and V. Klee. *Combinatorial Geometry in the Plane.* Holt, Rinehart and Winston, New York, 1964.

[Jam87] R. Jamison. Direction trees. *Discrete Comput. Geom.*, 2:249–254, 1987.

[JT95] T.R. Jensen and B. Toft. *Graph Coloring Problems.* Wiley-Interscience, New York, 1995.

[KW91] V. Klee and S. Wagon. *Old and New Unsolved Problems in Plane Geometry and Number Theory.* Math. Assoc. Amer., Washington, 1991.

[LT95] H. Lefmann and T. Thiele. Point sets with distinct distances. *Combinatorica*, 15:379–408, 1995.

[MP] W.O.J. Moser and J. Pach. *100 Research Problems in Discrete Geometry.* Manuscript.

[PA95] J. Pach and P.K. Agarwal. *Combinatorial Geometry.* Wiley-Interscience, New York, 1995.

[Pac93] J. Pach, editor. *New Trends in Discrete and Computational Geometry*. Springer-Verlag, Berlin, 1993.

[Sch88] O. Schramm. Illuminating sets of constant width. *Mathematika*, 35:180–199, 1988.

[Soi94] A. Soifer. Six-realizable set x_6. *Geombinatorics*, III:140–145, 1994.

[Szé95] L.A. Székely. Crossing numbers and hard Erdős problems in discrete geometry. Manuscript, 1995.

[Tót95] G. Tóth. The shortest distance among points in general position. Report 95-24, Math. Inst., Hungarian Academy of Sciences, Budapest, 1995. *Comput. Geom. Theory Appl.*, to appear.

[Val95] P. Valtr. On the minimum number of polygons in planar point sets. *Studia Sci. Math. Hungar.*, 30:155–163, 1995.

[Wor79] N.C. Wormald. A 4-chromatic graph with a special drawing. *J. Austral. Math. Soc. Ser. A*, 28:1–8, 1979.

2 PACKING AND COVERING

Gábor Fejes Tóth

INTRODUCTION

The basic problems in the classical theory of packings and coverings, the development of which was strongly influenced by the geometry of numbers and by crystallography, are the determination of the densest packing and the thinnest covering with congruent copies of a given body K. Roughly speaking, the density of an arrangement is the ratio between the total volume of the members of the arrangement and the volume of the whole space. In Section 2.1 we define this notion rigorously and give an account of the known density bounds.

In Section 2.2 we consider packings in, and coverings of, bounded domains. Section 2.3 is devoted to multiple arrangements and their decomposability. In Section 2.4 we make a detour to spherical and hyperbolic spaces. In Section 2.5 we discuss problems concerning the number of neighbors in a packing, while in Section 2.6 we investigate some selected problems concerning lattice arrangements. We close in Section 2.7 with problems concerning packing and covering with sequences of convex sets.

2.1 DENSITY BOUNDS FOR ARRANGEMENTS IN E^d

GLOSSARY

Convex body: A compact convex set with nonempty interior. A convex body in the plane is called a ***convex disk***. The collection of all convex bodies in d-dimensional Euclidean space $\mathbb{E}^d$ is denoted by $\mathcal{K}(\mathbb{E}^d)$. The subfamily of $\mathcal{K}(\mathbb{E}^d)$ consisting of centrally symmetric bodies is denoted by $\mathcal{K}^*(\mathbb{E}^d)$.

Operations on $\mathcal{K}(\mathbb{E}^d)$: For a set A and a real number λ we set $\lambda A = \{x \mid x = \lambda a,\ a \in A\}$. λA is called a ***homothetic copy*** of A. The ***Minkowski sum*** $A + B$ of the sets A and B consists of all points $a + b$, $a \in A$, $b \in B$. The set $A - A = A + (-A)$ is called the ***difference body*** of A. B^d denotes the unit ball centered at the origin, and $A + rB^d$ is called the ***parallel body*** of A at distance r ($r > 0$). If $A \subset \mathbb{E}^d$ is a convex body with the origin in its interior, then the ***polar body*** A^* of A is $\{x \in \mathbb{E}^d \mid \langle x, a\rangle \leq 1 \text{ for all } a \in A\}$.

The ***Hausdorff distance*** between the sets A and B is defined by

$$d(A, B) = \inf\{\varrho \mid A \subset B + \varrho B^d, B \subset A + \varrho B^d\}.$$

Lattice: The set of all integer linear combinations of a particular basis of $\mathbb{E}^d$.

Lattice arrangement: The set of translates of a given set in $\mathbb{E}^d$ by all vectors of a lattice.

Packing: A family of sets whose interiors are mutually disjoint.

Covering: A family of sets whose union is the whole space.

The ***volume*** (Lebesgue measure) of a measurable set A is denoted by $V(A)$. In the case of the plane we use the term ***area*** and the notation $a(A)$.

Density of an arrangement relative to a set: Let $\mathcal{A}$ be an arrangement (a family of sets each having finite volume) and D a set with finite volume. The ***inner density*** $d_{\text{inn}}(\mathcal{A}|D)$, ***outer density*** $d_{\text{out}}(\mathcal{A}|D)$, and ***density*** $d(\mathcal{A}|D)$ of $\mathcal{A}$ relative to D are defined by

$$d_{\text{inn}}(\mathcal{A}|D) = \frac{1}{V(D)} \sum_{A\in\mathcal{A}, A\subset D} V(A),$$

$$d_{\text{out}}(\mathcal{A}|D) = \frac{1}{V(D)} \sum_{A\in\mathcal{A},\, A\cap D\neq\emptyset} V(A),$$

and

$$d(\mathcal{A}|D) = \frac{1}{V(D)} \sum_{A\in\mathcal{A}} V(A\cap D).$$

(If one of the sums on the right side is divergent, then the corresponding density is infinite.)

The ***lower density*** and ***upper density*** of an arrangement $\mathcal{A}$ are given by the limits $d_-(\mathcal{A}) = \liminf_{\lambda\to\infty} d_{\text{inn}}(\mathcal{A}|\lambda B^d)$, $d_+(\mathcal{A}) = \limsup_{\lambda\to\infty} d_{\text{out}}(\mathcal{A}|\lambda B^d)$. If $d_-(\mathcal{A}) = d_+(\mathcal{A})$, then we call the common value the ***density*** of $\mathcal{A}$ and denote it by $d(\mathcal{A})$. It is easily seen that these quantities are independent of the choice of the origin.

The ***packing density*** $\delta(K)$ and ***covering density*** $\vartheta(K)$ of a convex body (or more generally of a measurable set) K are defined by

$$\delta(K) = \sup\{d_+(\mathcal{P}) \mid \mathcal{P} \text{ is a packing of } \mathbb{E}^d \text{ with congruent copies of } K\}$$

and

$$\vartheta(K) = \inf\{d_-(\mathcal{C}) \mid \mathcal{C} \text{ is a covering of } \mathbb{E}^d \text{ with congruent copies of } K\}.$$

The ***translational packing density*** $\delta_T(K)$, ***lattice packing density*** $\delta_L(K)$, ***translational covering density*** $\vartheta_T(K)$, and ***lattice covering density*** $\vartheta_L(K)$ are defined analogously, by taking the supremum and infimum over arrangements consisting of translates of K and over lattice arrangements of K, respectively. It is obvious that in the definitions of $\delta_L(K)$ and $\vartheta_L(K)$ we can take maximum and minimum instead of supremum and infimum. By a theorem of Groemer, the same holds for the translational and for the general packing and covering densities.

KNOWN VALUES OF PACKING AND COVERING DENSITIES

Apart from the obvious examples of space fillers, there are only a few specific bodies for which the packing or covering densities have been determined. The bodies for

TABLE 2.1.1 Bodies K for which $\delta(K)$ is known.

BODY	AUTHOR	SEE
Circle	Thue	[Fej72, p. 58]
Parallel body of a rectangle	L. Fejes Tóth	[EGH89]
Intersection of two congruent circles	L. Fejes Tóth	[EGH89]
Centrally symmetric n-gon (algorithm in $O(n)$ time)	Mount and Silverman	[FK93b]
Truncated rhombic dodecahedron	A. Bezdek	[Bez94]

which the packing density is known are given in Table 2.1.1. The circle is the only body that is not a space filler for which the covering density is known.

We have $\delta(B^2) = \pi/\sqrt{12}$. For the rest of the bodies in Table 2.1.1, the packing density can be given only by rather complicated formulas. We note that, with appropriate modification of the definition, the packing density of a set with infinite volume can also be defined. A. Bezdek and W. Kuperberg (see [FK93b]) showed that the packing density of an infinite circular cylinder is $\pi/\sqrt{12}$, that is, infinite circular cylinders cannot be packed more densely than their base. It is conjectured that the same statement holds for circular cylinders of any finite height.

A theorem of L. Fejes Tóth (see [Fej64, p. 163]) states that

$$\delta(K) \leq \frac{a(K)}{H(K)} \qquad \text{for } K \in \mathcal{K}(\mathbb{E}^2), \tag{2.1.1}$$

where $H(K)$ denotes the minimum area of a hexagon containing K. This bound is best possible for centrally symmetric disks, and it implies that

$$\delta(K) = \delta_T(K) = \delta_L(K) = \frac{a(K)}{H(K)} \qquad \text{for } K \in \mathcal{K}^*(\mathbb{E}^2).$$

The packing densities of the convex disks in Table 2.1.1 have been determined utilizing this relation.

It is conjectured that an inequality analogous to (2.1.1) holds for coverings, and this is supported by the following weaker result (see [Fej64, p. 167]):

Let $h(K)$ denote the maximum area of a hexagon contained in a convex disk K. Let $\mathcal{C}$ be a covering of the plane with congruent copies of K such that no two copies of K cross. Then

$$d_-(\mathcal{C}) \geq \frac{a(K)}{h(K)}.$$

The convex disks A and B ***cross*** if both $A \setminus B$ and $B \setminus A$ are disconnected. As translates of a convex disk do not cross, it follows that

$$\vartheta_T(K) \geq \frac{a(K)}{h(K)} \qquad \text{for } K \in \mathcal{K}(\mathbb{E}^2).$$

Again, this bound is best possible for centrally symmetric disks, and it implies that

$$\vartheta_T(K) = \vartheta_L(K) = \frac{a(K)}{h(K)} \qquad \text{for } K \in \mathcal{K}^*(\mathbb{E}^2). \tag{2.1.2}$$

Based on this, Mount and Silverman gave an algorithm that determines $\vartheta_T(K)$ for a centrally symmetric n-gon in $O(n)$ time. Also the classical result $\vartheta(B^2) = 2\pi/\sqrt{27}$ of Kershner (see [Fej72, p. 58]) follows from this relation.

One could expect that the restriction to arrangements of translates of a set means a considerable simplification. However, this apparent advantage has not been exploited so far in dimensions greater than 2. On the other hand, the lattice packing density of some special convex bodies in $\mathbb{E}^3$ has been determined; see Table 2.1.2.

TABLE 2.1.2 Bodies $K \in \mathbb{E}^3$ for which $\delta_L(K)$ is known.

BODY	$\delta_L(K)$	AUTHOR
Tetrahedron	$18/49$	Hoylman
$\{x \mid \|x\| \le 1, \|x_3\| \le \lambda\} \quad (\lambda \le 1)$	$\pi(3-\lambda^2)^{1/2}/6$	Chalk
$\{x \mid \|x_i\| \le 1,\ \|x_1+x_2+x_3\| \le \lambda\}$	$\begin{cases} \dfrac{9-\lambda^2}{9} & \text{for } 0 < \lambda \le \frac{1}{2} \\ \dfrac{9\lambda(9-\lambda^2)}{4(-\lambda^3-3\lambda^2+24\lambda-1)} & \text{for } \frac{1}{2} \le \lambda \le 1 \\ \dfrac{9(\lambda^3-9\lambda^2+27\lambda-3)}{8\lambda(\lambda^2-9\lambda+27)} & \text{for } 1 \le \lambda \le 3 \end{cases}$	Whitworth
$\{x \mid \sqrt{(x_1)^2+(x_2)^2}+\|x_3\| \le 1\}$	$\pi\sqrt{6}/9$	Whitworth

All results given in Table 2.1.2 can be traced in [EGH89]. We emphasize the following two special cases: Gauss's result that $\delta_L(B^3) = \pi/\sqrt{18}$ is the special case $\lambda = 1$ of Chalk's theorem concerning the frustrum of the ball, and Minkowski's result stating that the lattice packing density of the regular octahedron is $18/19$ is the case $\lambda = 1$ of Whitworth's theorem about the truncated cube.

The list in Table 2.1.2 can be augmented by additional bodies using the following observations.

It has been noticed by Chalk and Rogers that the relation $\delta_T(K) = \delta_L(K)$ $(K \in \mathcal{K}(\mathbb{E}^2))$ readily implies that for a cylinder C in $\mathbb{E}^3$ based on a convex disk K we have $\delta_L(C) = \delta_L(K)$. Thus, $\delta_L(C)$ is known if the lattice packing density of its base is known.

Next, we recall the observation of Minkowski (see [Rog64, p. 69]) that an arrangement $\mathcal{A}$ of translates of a convex body K is a packing if and only if the arrangement of translates of the body $\frac{1}{2}(K-K)$ by the same vectors is a packing. This implies that, for $K \in \mathcal{K}(\mathbb{E}^d)$,

$$\delta_T(K) = 2^d \delta_T(K-K)\frac{V(K)}{V(K-K)} \quad \text{and} \quad \delta_L(K) = 2^d \delta_L(K-K)\frac{V(K)}{V(K-K)} \tag{2.1.3}$$

If K is a regular tetrahedron, then $K-K$ is a cuboctahedron with volume $20V(K)$. Hence we get that the lattice packing density of the cuboctahedron is $45/49$.

Generally, K is not uniquely determined by $K-K$; e.g., we have $K-K = B^d$ for every $K \subset \mathbb{E}^d$ that is a body of constant width 1, and the determination of $\delta_L(K)$ for such a body is reduced to the determination of $\delta_L(B^d)$, which is established for $d \le 8$. We give the known values of $\delta_L(B^d)$, together with those of $\vartheta(B^d)$, in Table 2.1.3. All results given there can be traced in [CS93].

TABLE 2.1.3 Known values of $\delta_L(B^d)$ and $\vartheta_L(B^d)$.

d	$\delta_L(B^d)$	AUTHOR	$\vartheta_L(B^d)$	AUTHOR
2	$\frac{\pi}{2\sqrt{3}}$	Lagrange	$\frac{2\pi}{3\sqrt{3}}$	Kershner
3	$\frac{\pi}{\sqrt{18}}$	Gauss	$\frac{5\sqrt{5}\pi}{24}$	Bambah
4	$\frac{\pi^2}{16}$	Korkin and Zolotarev	$\frac{2\pi^2}{5\sqrt{5}}$	Delone and Ryškov
5	$\frac{\pi^2}{15\sqrt{2}}$	Korkin and Zolotarev	$\frac{245\sqrt{35}\pi^2}{3888\sqrt{3}}$	Baranovskiĭ and Ryškov
6	$\frac{\pi^3}{48\sqrt{3}}$	Blichfeldt		
7	$\frac{\pi^3}{105}$	Blichfeldt		
8	$\frac{\pi^4}{384}$	Blichfeldt		

EXISTENCE OF ECONOMICAL ARRANGEMENTS

Table 2.1.4 lists the known bounds establishing the existence of reasonably dense packings and thin coverings. When c appears in a bound without specification, it means a suitable constant characteristic of the specific bound. The proofs of most of these are nonconstructive. For constructive methods yielding slightly weaker bounds, as well as improvements for special convex bodies, see Chapter 50.

Bound 1 for the packing density of general convex bodies follows by combining Bound 6 with the relation (2.1.3) and the inequality $V(K-K) \leq \binom{2d}{d} V(K)$ of Rogers and Shephard (see [Rog64, Theorem 2.4]). For $d \geq 3$ all methods establishing the existence of dense packings rely on the theory of lattices, thus providing the same lower bounds for $\delta(K)$ and $\delta_T(K)$ as for $\delta_L(K)$.

Gritzmann (see [PA95]) proved a bound similar to Bound 4 for a larger class of convex bodies:

$$\vartheta_L(K) \leq cd(\ln d)^{1+\log_2 e}$$

holds for a suitable constant c and for every convex body K in $\mathbb{E}^d$ that has an affine image symmetric about at least $\log_2 \ln d + 4$ coordinate hyperplanes.

UPPER BOUNDS FOR $\delta(B^d)$ AND LOWER BOUNDS FOR $\vartheta(B^d)$

The packing and covering density of B^d is not known for $d \geq 3$. Asymptotically, the best upper bound known for $\delta(B^d)$ is

$$\delta(B^d) \leq 2^{-0.599d+o(d)} \qquad (\text{as } d \to \infty), \tag{2.1.4}$$

given by Kabatjanskiĭ and Levenštein (see [CS93]). For low dimensions, Rogers's simplex bound

$$\delta(B^d) \leq \sigma_d \tag{2.1.5}$$

gives a better estimate (see [Rog64, Theorem 7.1]). Here, σ_d is the ratio between

TABLE 2.1.4 Bounds establishing the existence of dense packings and thin coverings.

No.	BOUND	AUTHOR	SEE
	Bounds for general convex bodies in $\mathbb{E}^d$		
1	$\delta_L(K) \geq cd^{3/2}/4^d \quad (d \text{ large})$	Schmidt, Rogers, and Shephard	[Rog64]
2	$\vartheta_T(K) \leq d \ln d + d \ln \ln d + 5d$	Rogers	[Rog64, Theorem 3.2]
3	$\vartheta_L(K) \leq d^{\log_2 \ln d + c}$	Rogers	[Rog64]
4	$\vartheta_L(B^d) \leq cd(\ln d)^{\log_2 \sqrt{2\pi e}}$	Rogers	[Rog64]
	Bounds for centrally symmetric convex bodies in $\mathbb{E}^d$		
5	$\delta_L(K) \geq \zeta(d)/2^{d-1}$	Minkowski-Hlawka	[PA95, Theorem 7.7]
6	$\delta_L(K) \geq cd/2^d \quad (d \text{ large})$	Schmidt	[Rog64]
	Bounds for general convex bodies in $\mathbb{E}^2$		
7	$\delta(K) \geq \sqrt{3}/2 = 0.8660\ldots$	G. Kuperberg and W. Kuperberg	[PA95, Theorem 4.5]
8	$\vartheta(K) \leq 1.2281771\ldots$	Ismailescu	[Ism]
9	$\delta_L(K) \geq 2/3$	Fáry	[Fej72, p. 100]
10	$\vartheta_L(K) \leq 3/2$	Fáry	[Fej72, p. 100]
	Bounds for centrally symmetric convex bodies in $\mathbb{E}^2$		
11	$\delta_L(K) \geq 0.892656\ldots$	Tammela	[PA95]
12	$\vartheta_L(K) \leq 2\pi/\sqrt{27}$	L. Fejes Tóth	[Fej72, p. 103]

the total volume of the sectors of $d+1$ unit balls centered at the vertices of a regular simplex of edge 2 and the volume of the simplex.

A remark of Kepler can be interpreted in modern terminology as the conjecture that $\delta(B^3) = \pi/\sqrt{18} = 0.740480\ldots$. A packing of balls reaching this density is obtained by placing the centers at the vertices and face-centers of a cubic lattice. The best upper bound for $\delta(B^3)$ was given by Muder [Mud93]:

$$\delta(B^3) \leq 0.773055\ldots .$$

Coxeter, Few, and Rogers (see [Rog64, Theorem 8.1]) proved a dual counterpart to Rogers's simplex bound:

$$\vartheta(B^d) \geq \tau_d,$$

where τ_d is the ratio between the total volume of the intersections of $d+1$ unit balls with the regular simplex of edge $\sqrt{2(d+1)/d}$ if their centers lie at the vertices of the simplex, and the volume of the simplex. Asymptotically,

$$\tau_d \sim d/e^{3/2}.$$

In contrast to packings, where there is a sizable gap between bound (2.1.4) and the bound from the other direction (Bound 6 in Table 2.1.4), this bound compares quite favorably with the corresponding Bound 2 in Table 2.1.4.

According to a result of W. Schmidt (see [FK93b]), we have $\delta(K) < 1$ and $\vartheta(K) > 1$ for every smooth convex body; but the method of proof does not allow one to derive any explicit bound. There is a general upper bound for $\delta(K)$ that is nontrivial (smaller than 1) for a wide class of convex bodies [FK93a]. It is quite reasonable for "longish" bodies. For cylinders in $\mathbb{E}^d$, the bound is asymptotically equal to the Kabatjanskiĭ and Levenštein bound for B^d (as $d \to \infty$). We note that no nontrivial bound is known for $\vartheta(K)$ for any K other than a ball.

REGULARITY OF OPTIMAL ARRANGEMENTS

The packings and coverings attaining the packing and covering densities of a set are, of course, not uniquely determined, but it is a natural question whether there exist among the optimal arrangements some that satisfy certain regularity properties. Of particular interest are those bodies for which the densest packing and/or thinnest covering with congruent copies can be realized by a lattice arrangement. As mentioned above, $\delta(K) = \delta_L(K)$ for $K \in \mathcal{K}^*(\mathbb{E}^2)$. A plausible interpretation of this result is that the assumption of maximum density creates from a chaotic structure a regular one. Unfortunately, certain results indicate that such bodies are rather exceptional.

Let $\mathcal{L}_p$ and $\mathcal{L}_c$ be the classes of those convex disks $K \in \mathcal{K}(\mathbb{E}^2)$ for which $\delta(K) = \delta_L(K)$ and $\vartheta(K) = \vartheta_L(K)$, respectively. Then, in the topology induced by the Hausdorff metric on $\mathcal{K}(\mathbb{E}^2)$, the sets $\mathcal{L}_p$ and $\mathcal{L}_c$ are nowhere dense [FZ94, Fej95]. It is conjectured that an analogous statement holds also in higher dimensions.

Rogers [Rog64, p. 15] conjectures that for sufficiently large d we have $\delta(B^d) > \delta_L(B^d)$. The following result of A. Bezdek and W. Kuperberg (see [FK93b]) supports this conjecture: For $d \geq 3$ there are ellipsoids E in $\mathbb{E}^d$ for which $\delta(E) > \delta_L(E)$. An even more surprising result holds for coverings [FK95]: For $d \geq 3$ every strictly convex body K in $\mathbb{E}^d$ has an affine image K' such that $\vartheta(K') < \vartheta_L(K')$. In particular, there is an ellipsoid E in $\mathbb{E}^3$ for which

$$\vartheta(E) < 1.394 < \frac{3\sqrt{3}}{2}(3\,\mathrm{arcsec}\,3 - \pi) = \tau_3 \leq \vartheta_T(E) \leq \vartheta_L(E).$$

We note that no example of a convex body K is known for which $\delta_L(K) < \delta_T(K)$ or $\vartheta_L(K) > \vartheta_T(K)$.

Schmitt [Sch88] constructed a star-shaped prototile for a monohedral tiling in $\mathbb{E}^3$ such that no tiling with its replicas is periodic. It is not known whether a convex body with this property exists; however, with a slight modification of Schmitt's construction, Conway produced a convex prototile that admits only non-periodic tilings if no mirror-image is allowed (see Section 3.4). Another result of Schmitt's [Sch91] is that there are star-shaped sets in the plane whose densest packing cannot be realized in a periodic arrangement.

2.2 FINITE ARRANGEMENTS

PACKING IN AND COVERING OF A BODY WITH GIVEN SHAPE

What is the size of the smallest square tray that can hold n given glasses? Thue's result gives a bound that is asymptotically sharp as $n \to \infty$; however, for practical reasons, small values of n are of interest.

Generally, for given sets K and C and a positive integer n one can ask for the quantities

$$M_p(K, C, n) = \inf\{\lambda \mid n \text{ congruent copies of } C \text{ can be packed in } \lambda K\}$$

and

$$M_c(K, C, n) = \sup\{\lambda \mid n \text{ congruent copies of } C \text{ can cover } \lambda K\}.$$

Tables 2.2.1 and 2.2.2 contain the known results about the cases when C is a circle and K is a circle, square, or regular triangle. In addition, economical circle packings and circle coverings have been constructed for many special values of n. All of these results can be traced in [HM, Mel93, Mel94, Mel97, Pei94]. Concerning the thinnest covering of a circle with congruent circles, we mention the conjecture that $M_c(K, B^2, n) = 1 + 2\cos\dfrac{2\pi}{n-1}$ for $8 \leq n \leq 10$. The paper of Krotoszyński [Kro93] claims this as a theorem even for $n \leq 11$; however, his proof contains a gap. In fact, Melissen and Schuur have found an example showing that the result does not hold for $n = 11$.

Most of these results were obtained by ad hoc methods. Recently, however, Peikert described a heuristic algorithm for the determination of $M_p(K, B^2, n)$ and the corresponding optimal arrangements in the case where K is the unit square. His algorithm consists of the following steps:

Step 1. Find a good upper bound m for $M_p(K, B^2, n)$. This requires the construction of a reasonably good arrangement, which can be established, e.g., by the Monte Carlo method.

Step 2. Iterate an elimination process on a successively refined grid to restrict possible locations for the centers of a packing of unit circles in mK.

Step 3. Based on the result of Step 2, guess the nerve graph of the packing, then determine the optimal packing with the given graph.

Step 4. Verify that the arrangement obtained in Step 3 is indeed optimal.

Peikert does not prove that these steps always provide the optimal arrangement in finite time, but he implemented the method successfully for $n \leq 20$. The best arrangements are shown in Figure 2.2.1. Observe that quite often an optimal arrangement can contain a freely movable circle.

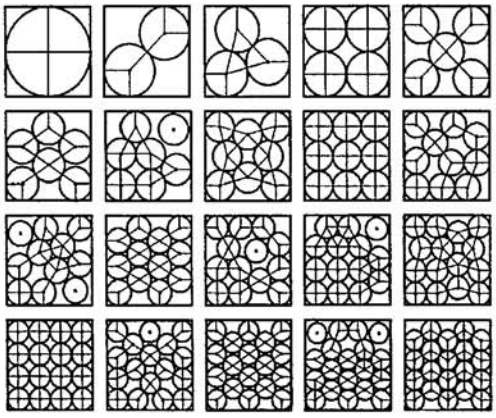


FIGURE 2.2.1
Densest packing of $n \leq 20$ equal circles in a square.

The sequence $M_p(K, B^2, n)$ seems to be strictly increasing when K is a square or when K is a circle and $n \geq 7$. In contrast to this, it is conjectured that in the case where K is a triangle, we have $M_p(K, B^2, n) = M_p(K, B^2, n-1)$ for all triangular numbers $n = k(k+1)/2$ $(k > 1)$.

TABLE 2.2.1 Packing of congruent circles in circles, squares, and equilateral triangles.

K	n	$M_p(K, B^2, n)$	AUTHOR
B^2	2	2	(elementary)
	3	2.154700538...	(elementary)
	4	2.414213562...	(elementary)
	5	2.701301617...	(elementary)
	6	3	(elementary)
	7	3	(elementary)
	8	3.304764871...	Pirl
	9	3.61312593...	Pirl
	10	3.813898249...	Pirl
	11	3.9238044...	Melissen
Unit square	2	3.414213562...	(elementary)
	3	3.931851653...	(elementary)
	4	4	(elementary)
	5	4.828427125...	(elementary)
	6	5.328201177...	Graham, Melissen
	7	5.732050807...	Schaer
	8	5.863703305...	Schaer and Meir
	9	6	Schaer
	10	6.747441523...	Peikert
	11	7.022509506...	Peikert
	12	7.144957554...	Peikert
	13	7.463047839...	Peikert
	14	7.732050808...	Wengerodt
	15	7.863703305...	Peikert
	16	8	Wengerodt
	17	8.532660354...	Peikert
	18	8.656402355...	Peikert
	19	8.907460939...	Peikert
	20	8.978083353...	Peikert
	25	10	Wengerodt
	36	12	Kirchner and Wengerodt
Regular triangle of side 1	2	5.464101615...	(elementary)
	3	5.464101615...	(elementary)
	4	6.92820323...	Melissen
	5	7.464101615...	Melissen
	6	7.464101615...	Ohler, Groemer
	7	8.92820323...	Melissen
	8	9.293810046...	Melissen
	9	9.464101615...	Melissen
	10	9.464101615...	Ohler, Groemer
	11	10.73008794...	Melissen
	12	10.92820323...	Melissen
	$k(k+1)/2$	$2(k+\sqrt{3}-1)$	Ohler, Groemer

The problem of finding the densest packing of n congruent circles in a circle has been considered also in the Minkowski plane. In terms of Euclidean geometry, this is the same as asking for the smallest number $\varrho(n, K)$ such that n mutually disjoint translates of the centrally symmetric convex disk K (the unit circle in the

TABLE 2.2.2 Covering circles, squares, and equilateral triangles with congruent circles.

K	n	$M_c(K, B^2, n)$	AUTHOR
B^2	2	2	(elementary)
	3	$2/\sqrt{3}$	(elementary)
	4	$\sqrt{2}$	(elementary)
	5	1.64100446...	K. Bezdek
	6	1.7988...	K. Bezdek
	7	2	(elementary)
Unit square	2	$4\sqrt{5}/5$	(elementary)
	3	1.984555...	Heppes and Melissen
	4	$2\sqrt{2}$	Heppes and Melissen
	5	3.065975...	Heppes and Melissen
	7	3.6457524...	Heppes and Melissen
Regular triangle of side 1	2	2	(elementary)
	3	$2\sqrt{3}$	Melissen
	4	$2+\sqrt{3}$	Melissen
	5	4	Melissen
	6	$\sqrt{27}$	Melissen

Minkowski metric) can be contained in $\varrho(n, K)K$. Doyle, Lagarias, and Randell [DLR92] solved the problem for all $K \in \mathcal{K}^*(\mathbb{E}^2)$ and $n \leq 7$. There is an n-gon inscribed in K having equal sides in the Minkowski metric (generated by K) and having a vertex at an arbitrary boundary point of K. Let $\alpha(n, K)$ be the maximum Minkowski side-length of such an n-gon. Then we have $\varrho(n, K) = 1 + 2/\alpha(n, K)$ for $2 \leq n \leq 6$ and $\varrho(7, K) = \varrho(6, K) = 3$.

The densest packing of n congruent balls in a cube is known for $n \leq 10$ (see [Sch94]). The problem of finding the densest packing of congruent balls in other regular polytopes has been investigated by K. Bezdek (see [CFG91]).

SAUSAGE CONJECTURES

Intensive research on another type of finite packing and covering problem has been generated by the sausage conjectures of L. Fejes Tóth and Wills (see [GW93]):

What is the convex body of minimum volume in $\mathbb{E}^d$ that can accommodate k nonoverlapping unit balls?

What is the convex body of maximum volume in $\mathbb{E}^d$ that can be covered by k unit balls?

According to the conjectures mentioned above, for $d \geq 5$ the extreme bodies are "sausages" and in the optimal arrangements the centers of the balls are equally spaced on a line segment (Figure 2.2.2).

After several partial results supporting these conjectures (see [GW93]) the breakthrough concerning the sausage conjecture for ball packings was achieved by Betke, Henk, and Wills [BHW94]: they proved that the conjecture holds for dimensions $d \geq 13387$. Later, Betke and Henk [BH] improved the bound on d to $d \geq 42$.

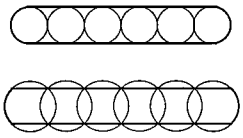


FIGURE 2.2.2
Sausage-like arrangements of circles.

THE COVERING PROBLEMS OF BORSUK AND HADWIGER-LEVI

In 1933, Borsuk formulated the conjecture that any bounded set in $\mathbb{E}^d$ can be partitioned into $d+1$ subsets of smaller diameter. Borsuk verified the conjecture for $d = 2$, and the three-dimensional case was settled independently by Eggleston, Grünbaum, and Heppes. The conjecture is known to be true also for many special cases: for smooth convex bodies (Hadwiger), for centrally symmetric sets (Riesling), as well as for sets having the symmetry group of the regular simplex (Rogers). Quite recently, however, Kahn and Kalai [KK93] showed that Borsuk's conjecture is false in the following very strong sense: Let $b(d)$ denote the smallest integer such that every bounded set in $\mathbb{E}^d$ can be partitioned into $b(d)$ subsets of smaller diameter. Then $b(d) \geq (1.2)^{\sqrt{d}}$ for every sufficiently large value of d.

In the fifties, Hadwiger and Levi, independently of each other, asked for the smallest integer $h(K)$ such that the convex body K can be covered by $h(K)$ smaller positively homothetic copies of K. Hadwiger conjectured that $h(K) \leq 2^d$ for all $K \in \mathcal{K}(\mathbb{E}^d)$ and that equality holds only for parallelotopes. Levi verified the conjecture for the plane, but it is open for $d \geq 3$. Lassak proved Hadwiger's conjecture for centrally symmetric convex bodies in $\mathbb{E}^3$, and K. Bezdek extended Lassak's result to convex polytopes with any affine symmetry.

Boltjanskiĭ observed that the Hadwiger-Levi covering problem for convex bodies is equivalent to an illumination problem. We say that a boundary point x of the convex body K is ***illuminated from the direction u*** if the ray issuing from x in the direction u intersects the interior of K. Let $i(K)$ be the minimum number of directions from which the boundary of K can be illuminated. Then $h(K) = i(K)$ for every convex body. For literature and further results concerning the Hadwiger-Levi problem, we refer to [Bez93].

2.3 MULTIPLE ARRANGEMENTS

GLOSSARY

k-fold packing: An arrangement $\mathcal{A}$ such that each point of the space belongs to the interior of at most k members of $\mathcal{A}$.

k-fold covering: An arrangement $\mathcal{A}$ such that each point of the space belongs to at least k members of $\mathcal{A}$.

Densities: In analogy to the packing and covering densities of a body K, we define the quantities $\delta^k(K)$, $\delta_T^k(K)$, $\delta_L^k(K)$, $\vartheta^k(K)$, $\vartheta_T^k(K)$, and $\vartheta_L^k(K)$ as the suprema of the densities of all k-fold packings and the infima of the densities of all k-fold coverings with congruent copies, translates, and lattice translates of K, respectively.

TABLE 2.3.1 Bounds for k-fold packing and covering densities.

BOUND	AUTHOR
$\delta_T^k(K) \geq ck \qquad K \in \mathcal{K}(\mathbb{E}^d)$	Erdős and Rogers
$\vartheta_L^k(K) \leq ((k+1)^{1/d} + 8d)^d \qquad K \in \mathcal{K}(\mathbb{E}^d)$	Cohn
$\delta_L^k(K) \geq k - ck^{2/5} \qquad K \in \mathcal{K}(\mathbb{E}^2)$	Bolle
$\vartheta_L^k(K) \leq k + ck^{2/5} \qquad K \in \mathcal{K}(\mathbb{E}^2)$	Bolle
$\delta^k(B^d) \geq (2k/(k+1))^{d/2}\delta(B^d)$	Few
$\delta_L^k(B^d) \geq (2k/(k+1))^{d/2}\delta_L(B^d)$	Few
$\delta^k(B^d) \leq (1+d^{-1})((d+1)^k - 1)(k/(k+1))^{d/2}$	Few
$\delta^2(B^d) \leq \frac{4}{3}(d+2)(\frac{2}{3})^{d/2}$	Few
$\vartheta^k(B^d) \geq ck \qquad c = c_d > 1$	G. Fejes Tóth
$\delta^k(B^2) \leq \frac{\pi}{6}\cot\frac{\pi}{6k}$	G. Fejes Tóth
$\vartheta^k(B^2) \geq \frac{\pi}{3}\csc\frac{\pi}{3k}$	G. Fejes Tóth

The information known about the asymptotic behavior of k-fold packing and covering densities is summarized in Table 2.3.1. There, in the various bounds, different constants appear, all of which we denote by c. All results given in the table can be traced in [EGH89] and [Fej83].

The known values of $\delta_L^k(B^d)$ and $\vartheta_L^k(B^d)$ (for $k \geq 2$) are given in Table 2.3.2 and can be traced in [EGH89, Fej83, FK93b, Tem94a, Tem94b].

Recently, general methods for the determination of the densest k-fold lattice packings and the thinnest k-fold lattice coverings with circles have been developed by Horváth, Temesvári, and Yakovlev and by Temesvári, respectively (see [FK93b]).

These methods reduce both problems to the determination of the optima of finitely many well-defined functions of one variable. The proofs readily provide algorithms for finding the optimal arrangements; however, the authors did not try to implement them. Only the values of $\delta_L^9(B^2)$ and $\vartheta_L^8(B^2)$ have been added in this way to the list of values of $\delta_L^k(B^2)$ and $\vartheta_L^k(B^2)$ that had been determined previously by ad hoc methods.

We note that we have $\delta_L^k(B^2) = k\delta_L(B^2)$ for $k \leq 4$ and $\vartheta_L^2(B^2) = 2\vartheta_L(B^2)$. These are the only cases where the extreme multiple arrangements of circles are not better than repeated simple arrangements. These relations have been extended to arbitrary centrally symmetric convex disks by Dumir and Hans-Gill and by G. Fejes Tóth (see [FK93b]). There is a simple reason for the relations $\delta_L^3(K) = 3\delta_L(K)$ and $\delta_L^4(K) = 4\delta_L(K)$ $(K \in \mathcal{K}^*(\mathbb{E}^2))$: Every 3-fold lattice packing of the plane with a centrally symmetric disk is the union of 3 simple lattice packings and every 4-fold packing is the union of two 2-fold packings.

This last observation brings us to the topic of decompositions of multiple arrangements. Our goal here is to find insight into the structure of multiple arrangements by decomposing them into possibly few simple ones. Pach showed (see [FK93b]) that any double packing with positively homothetic copies of a convex disk can be decomposed into 4 simple packings. Further, if $\mathcal{P}$ is a k-fold packing with convex disks such that for some integer L the inradius $r(K)$ and the area $a(K)$ of each member K of $\mathcal{P}$ satisfy the inequality $9\pi^2 kr^2(K)/a(K) \leq L$, then $\mathcal{P}$ can be decomposed into L simple packings.

Concerning the decomposition of multiple coverings Pach proved (see [FK93b]) that for any centrally symmetric polygon P and positive integer r there exists an

TABLE 2.3.2 Known values of $\delta_L^k(B^d)$ and $\vartheta_L^k(B^d)$.

RESULT	AUTHOR
$\delta_L^2(B^2) = \dfrac{\pi}{\sqrt{3}}$	Heppes
$\delta_L^3(B^2) = \dfrac{\sqrt{3}\pi}{2}$	Heppes
$\delta_L^4(B^2) = \dfrac{2\pi}{\sqrt{3}}$	Heppes
$\delta_L^5(B^2) = \dfrac{4\pi}{\sqrt{7}}$	Szirucsek, Blundon
$\delta_L^6(B^2) = \dfrac{35\pi}{8\sqrt{6}}$	Blundon
$\delta_L^7(B^2) = \dfrac{8\pi}{\sqrt{15}}$	Blundon, Krejcarek, Bolle
$\delta_L^8(B^2) = \dfrac{3969\pi}{4\sqrt{220 - 2\sqrt{193}}\sqrt{449 + 32\sqrt{193}}}$	Bolle, Yakovlev
$\delta_L^9(B^2) = \dfrac{25\pi}{2\sqrt{21}}$	Temesvári
$\delta_L^2(B^3) = \dfrac{8\pi}{9\sqrt{3}}$	Few and Kanagasabapathy
$\vartheta_L^2(B^2) = \dfrac{4\pi}{3\sqrt{3}}$	Blundon
$\vartheta_L^3(B^2) = \dfrac{\pi\sqrt{27138 + 2910\sqrt{97}}}{216}$	Blundon
$\vartheta_L^4(B^2) = \dfrac{25\pi}{18}$	Blundon
$\vartheta_L^5(B^2) = \dfrac{32\pi}{7\sqrt{7}}$	Subak, Temesvári
$\vartheta_L^6(B^2) = \dfrac{98\pi}{27\sqrt{3}}$	Subak, Temesvári
$\vartheta_L^7(B^2) = 7.672\ldots$	Haas, Temesvári
$\vartheta_L^8(B^2) = \dfrac{32\pi}{3\sqrt{15}}$	Temesvári
$\vartheta_L^2(B^3) = \dfrac{8\pi}{\sqrt{3}\sqrt{76\sqrt{6} - 159}}$	Few

integer $k = k(P, r)$ such that every k-fold covering with translates of P can be decomposed into r coverings. The attempt to extend this result by an approximation argument to all centrally symmetric disks fails, since, for fixed r, $k(P, r)$ approaches infinity as the number of sides of P tends to infinity. For circle coverings, however, Mani and Pach (see [FK93b]) were able to establish a decomposition theorem: Every 33-fold covering with congruent circles can be decomposed into two coverings. In 3-space, results analogous to the two theorems above do not hold.

2.4 PROBLEMS IN NON-EUCLIDEAN SPACES

Research on packing and covering in spherical and hyperbolic spaces has been concentrated on arrangements of balls. In contrast to spherical geometry, where the

finite, combinatorial nature of the problems, as well as applications, have inspired research, investigations in hyperbolic geometry have been hampered by the lack of a reasonable notion of density relative to the whole hyperbolic space.

SPHERICAL SPACE

Let $M(d,\varphi)$ be the maximum number of caps of spherical diameter φ forming a packing on the d-dimensional spherical space $\mathbb{S}^d$, that is, on the boundary of B^{d+1}, and let $m(d,\varphi)$ be the minimum number of caps of spherical diameter φ covering $\mathbb{S}^d$. An upper bound for $M(d,\varphi)$, which is sharp for certain values of d and φ and yields the best estimate known as $d \to \infty$, is the so-called ***linear programming bound*** (see [CS93, pp. 257-266]). It establishes a surprising connection between $M(d,\varphi)$ and the expansion of real polynomials in terms of certain Jacobi polynomials. The Jacobi polynomials, $P_i^{(\alpha,\beta)}(x)$, $i = 0, 1 \ldots, \alpha > -1$, $\beta > -1$, form a complete system of orthogonal polynomials on $[-1,1]$ with respect to the weight function $(1-x)^\alpha(1+x)^\beta$. Set $\alpha = \beta = (d-1)/2$ and let

$$f(t) = \sum_{i=0}^{k} f_i P_i^{(\alpha,\alpha)}(t)$$

be a real polynomial such that $f_0 > 0$, $f_i \geq 0$ $(i = 1, 2, \ldots, k)$, and $f(t) \leq 0$ for $-1 \leq t \leq \cos\varphi$. Then

$$M(d,\varphi) \leq f(1)/f_0.$$

With the use of appropriate polynomials Kabatjanskiĭ and Levenštein (see [CS93]) obtained the asymptotic bound:

$$\frac{1}{d}\ln M(d,\varphi) \leq \frac{1+\sin\varphi}{2\sin\varphi}\ln\frac{1+\sin\varphi}{2\sin\varphi} - \frac{1-\sin\varphi}{2\sin\varphi}\ln\frac{1-\sin\varphi}{2\sin\varphi} + o(1).$$

This implies the simpler bound

$$M(d,\varphi) \leq (1-\cos\varphi)^{-d/2}2^{-0.099d+o(d)} \qquad (\text{as } d \to \infty,\ \varphi \leq \varphi^* = 62.9974\ldots).$$

Bound (2.1.4) for $\delta(B^d)$ follows in the limiting case when $\varphi \to 0$.

The following is a list of some special values of d and φ for which the linear programming bound turns out to be exact (see [CS93]).

$M(2, \arccos 1/\sqrt{5}) = 12$	$M(4, \arccos 1/5) = 16$	$M(5, \arccos 1/4) = 27$
$M(6, \arccos 1/3) = 56$	$M(7, \pi/3) = 240$	$M(20, \arccos 1/9) = 112$
$M(20, \arccos 1/7) = 162$	$M(21, \arccos 1/11) = 100$	$M(21, \arccos 1/6) = 275$
$M(21, \arccos 1/4) = 891$	$M(22, \arccos 1/5) = 552$	$M(22, \arccos 1/3) = 4600$
	$M(23, \pi/3) = 196560$	

For small values of d and specific values of φ the linear programming bound is superseded by the "simplex bound" of Böröczky (see [FK93b]), which is the generalization of Rogers's bound (2.1.5) for ball packings in $\mathbb{S}^d$.

The value of $M(d,\varphi)$ has been determined for all d and $\varphi \geq \pi/2$ (see [CS93]). We have

$$M(d,\varphi) = i+1 \qquad \text{for } \frac{1}{2}\pi + \arcsin\frac{1}{i+1} < \varphi \leq \frac{1}{2}\pi + \arcsin\frac{1}{i}, \qquad i = 1, \ldots, d,$$

$$M(d,\varphi) = d+2 \qquad \text{for } \frac{1}{2}\pi < \varphi \le \frac{1}{2}\pi + \arcsin\frac{1}{d+1},$$

and

$$M(d,\frac{1}{2}\pi) = 2(d+1).$$

Except for an upper bound on $m(d,\varphi)$ establishing the existence of reasonably economic coverings of $\mathbb{S}^d$ by equal balls due to Rogers (see [Fej83]), no results on coverings in spherical spaces of high dimensions are known.

Extensive research has been done on circle packings and circle coverings on $\mathbb{S}^2$. Traditionally, here the inverse functions of $M(2,\varphi)$ and $m(2,\varphi)$ are considered. Let a_n be the maximum number such that n caps of spherical diameter a_n can form a packing and let A_n be the minimum number such that n caps of spherical diameter A_n can form a covering on $\mathbb{S}^2$. The known values of a_n and A_n are given in Table 2.4.1. All the results mentioned in the table can be traced in [Fej72]. In addition, conjecturally best circle packings and circle coverings for $n \le 130$, as well as good arrangements with icosahedral symmetry for $n \le 55000$, have been constructed [HSS]. The ad hoc methods of the earlier constructions have recently been replaced by different computer algorithms, but none of them has been shown to give the optimum.

TABLE 2.4.1 Densest packing and thinnest covering with congruent circles on a sphere.

n	a_n	AUTHOR	A_n	AUTHOR
2	180°	(elementary)	180°	(elementary)
3	120°	(elementary)	180°	(elementary)
4	109.471...°	L. Fejes Tóth	141.047...°	L. Fejes Tóth
5	90°	Schütte and van der Waerden	126.869...°	Schütte
6	90°	L. Fejes Tóth	109.471...°	L. Fejes Tóth
7	77.866...°	Schütte and van der Waerden	102.053...°	Schütte
8	74.869...°	Schütte and van der Waerden		
9	70.528...°	Schütte and van der Waerden		
10	66.316...°	Danzer, Hárs	84.615...°	G. Fejes Tóth
11	63.435...°	Böröczky, Danzer		
12	63.435...°	L. Fejes Tóth	74.754...°	L. Fejes Tóth
14			69.875...°	G. Fejes Tóth
24	43.667...°	Robinson		

Observe that $a_5 = a_6$ and $a_{11} = a_{12}$. Also, $A_2 = A_3$. It is conjectured that $a_n > a_{n+1}$ and $A_n > A_{n+1}$ in all other cases.

HYPERBOLIC SPACE

There is no sensible way to define the density of a general arrangement of sets in the d-dimensional hyperbolic space $\mathbb{H}^d$ (see [FK93b]). The main difficulty is that in hyperbolic geometry the volume and the surface area of a ball of radius r are of the same order of magnitude as $r \to \infty$. In the absence of a reasonable definition of density with respect to the whole space, two natural problems arise:

(i) Estimate the density of an arrangement relative to a bounded domain;

(ii) Find substitutes for the notions of densest packing and thinnest covering.

Concerning the first problem, we mention the following result of K. Bezdek (see [FK93b]). Consider a packing of finitely many, but at least two, circles of radius r in the hyperbolic plane $\mathbb{H}^2$. Then the density of the circles relative to the outer parallel domain of radius r of the convex hull of their centers is at most $\pi/\sqrt{12}$.

As a corollary it follows that if at least two congruent circles are packed in a circular domain in $\mathbb{H}^2$, then the density of the packing relative to the domain is at most $\pi/\sqrt{12}$. We note that the density of such a finite packing relative to the convex hull of the circles can be arbitrarily close to 1 as $r \to \infty$.

$\mathcal{P}$ is a ***solid packing*** if no finite subset of $\mathcal{P}$ can be rearranged so as to form, together with the rest of $\mathcal{P}$, a packing not congruent to $\mathcal{P}$. Analogously, $\mathcal{C}$ is a ***solid covering*** if no finite subset of $\mathcal{C}$ can be rearranged so as to form, together with the rest of $\mathcal{C}$, a covering not congruent to $\mathcal{C}$. Obviously, in $\mathbb{E}^d$ a solid packing with congruent copies of a body K has density $\delta(K)$, and a solid covering with congruent copies of K has density $\vartheta(K)$. This justifies the use of solidity as a natural substitute for "densest packing" and "thinnest covering" in hyperbolic space.

The tiling with Schläfli symbol $\{p, 3\}$ (see Chapters 3, 16, or 18 of this Handbook) has regular p-gonal faces such that at each vertex of the tiling three faces meet. There exists such a tiling for each $p \geq 2$: for $p \leq 5$ on the sphere, for $p \geq 7$ on the hyperbolic plane, while for $p = 6$ we have the well-known hexagonal tiling on the Euclidean plane. The incircles of such a tiling form a solid packing and the circumcircles form a solid covering. In addition, the incircles and the circumcircles of certain trihedral Archimedean tilings have been confirmed to be solid (see [FK93b]).

Another substitute for the notion of densest packing and thinnest covering is complete saturation and complete reduction. A packing $\mathcal{P}$ with congruent copies of a body K is ***completely saturated*** if no finite subset of $\mathcal{P}$ can be replaced by a greater number of congruent copies of K that, together with the rest of $\mathcal{P}$, form a packing. Analogously, a covering $\mathcal{C}$ with congruent copies of K is ***completely reduced*** if no finite subset of $\mathcal{C}$ can be replaced by a smaller number of congruent copies of K that, together with the rest of $\mathcal{C}$, form a covering. While there are convex bodies that do not admit a solid packing or solid covering, it is conjectured that each body in $\mathbb{E}^d$ or $\mathbb{H}^d$ admits a completely saturated packing and a completely reduced covering. This has recently been established for convex bodies in $\mathbb{E}^d$ [FKK].

2.5 NEIGHBORS

GLOSSARY

Neighbors: Two members of a packing whose closures intersect.

Newton number $N(K)$ of a convex body K: The maximum number of neighbors of K in all packings with congruent copies of K.

Hadwiger number $H(K)$ of a convex body K: The maximum number of neighbors of K in all packings with translates of K.

n-neighbor packing: A packing in which each member has exactly n neighbors.

n^+-neighbor packing: A packing in which each member has at least n neighbors.

Table 2.5.1 contains the results known about Newton numbers and Hadwiger numbers (see [CS93] and [FK93b]).

TABLE 2.5.1 Newton and Hadwiger numbers.

BODY K	RESULT	AUTHOR
B^3	$N(K) = 12$	Schütte and van der Waerden
B^8	$N(K) = 240$	Levenštein; Odlyzko and Sloane
B^{24}	$N(K) = 196560$	Levenštein; Odlyzko and Sloane
Regular triangle	$N(K) = 12$	Böröczky
Square	$N(K) = 8$	Böröczky
Regular pentagon	$N(K) = 6$	Linhart
Regular n-gon for $n \geq 6$	$N(K) = 6$	Böröczky
Isosceles triangle with base angle $\pi/6$	$N(K) = 21$	Wegner
Convex disk of diameter d and width w	$N(K) \leq (4 + 2\pi)\, d/w + w/d + 2$	L. Fejes Tóth
Parallelotope in $\mathbb{E}^d$	$H(K) = 3^d - 1$	Hadwiger
Convex body in $\mathbb{E}^d$	$H(K) \leq 3^d - 1$	Hadwiger
Compact set in $\mathbb{E}^d$ with $\operatorname{int}(K - K) \neq \emptyset$	$H(K) \geq d^2 + d$	Smith

It seems that the maximum number of neighbors of one body in a lattice packing with congruent copies of K is considerably smaller than $H(K)$. While $H(B^d)$ is of exponential order of magnitude, the highest known number of neighbors in a lattice packing with B^d occurs in the Barnes Wall lattice and is $c^{O(\log d)}$ [CS93]. Moreover, Gruber showed that, in the sense of Baire categories, most convex bodies in $\mathbb{E}^d$ have no more than $2d^2$ neighbors in their densest lattice packing. Recently, Alon [Alo95] constructed a finite ball packing in $\mathbb{E}^d$ in which each ball has $c^{O(\sqrt{d})}$ neighbors.

A problem related to the determination of the Hadwiger number concerns the maximum number $C(K)$ of mutually nonoverlapping translates of a set K that have a common point. No more than four nonoverlapping translates of a topological disk in the plane can share a point [BKK95], while for $d \geq 3$ there are starlike bodies in $\mathbb{E}^d$ for which $C(K)$ is arbitrarily large.

For a given convex body K, let $M(K)$ denote the maximum natural number with the property that an $M(K)$-neighbor packing with finitely many congruent copies of K exists. For $n \leq M(K)$, let $L(n, K)$ denote the minimum cardinality, and, for $n > M(K)$, let $\lambda(n, K)$ denote the minimum density, of an n-neighbor packing with congruent copies of K. The quantities $M_T(K)$, $M^+(K)$, $M_T^+(K)$, $L_T(n, K)$, $L^+(n, K)$, $L_T^+(n, K)$, $\lambda_T(n, K)$, $\lambda^+(n, K)$, and $\lambda_T^+(n, K)$ are defined analogously.

Österreicher and Linhart showed (see [FK93c]) that for a smooth convex disk K we have $L(2, K) \geq 3$, $L(3, K) \geq 6$, $L(4, K) \geq 8$, and $L(5, K) \geq 16$. All of these inequalities are sharp. We have $M_T^+(K) = 3$ for all convex disks, and there exists a

4-neighbor packing of density 0 with translates of any convex disk. There exists a 5-neighbor packing of density 0 with translates of a parallelogram, but Makai proved (see [FK93c]) that $\lambda_T^+(5,K) \geq 3/7$ and $\lambda_T^+(6,K) \geq 1/2$ for every $K \in \mathcal{K}(\mathbb{E}^2)$ that is not a parallelogram, and that $\lambda_T^+(5,K) \geq 9/14$ and $\lambda_T^+(6,K) \geq 3/4$ for every $K \in \mathcal{K}^*(\mathbb{E}^2)$ that is not a parallelogram. The case of equality characterizes triangles and affinely regular hexagons, respectively. According to a result of Chvátal (see [FK93b]), $\lambda_T^+(6,P) = 11/15$ for a parallelogram P.

A construction of Wegner (see [FK93b] shows that $M(B^3) \geq 6$ and $L(6,B^3) \leq 240$, while Kertész [Ker94] proved that $M(B^3) \leq 8$. It is an open problem whether an n-neighbor or n^+-neighbor packing of finitely many congruent balls exists for $n = 7$ and $n = 8$.

For 6^+-neighbor packings with (not necessarily equal) circles, the following nice theorem of Bárány, Füredi, and Pach (see [FK93c]) holds:

In a 6^+-neighbor packing with circles, either all circles are congruent or arbitrarily small circles occur.

2.6 SELECTED PROBLEMS ON LATTICE ARRANGEMENTS

In this section we discuss, from the vast literature on lattices, some special problems concerning arrangements of convex bodies in which the restriction to lattice arrangements is automatically imposed by the nature of the problem.

GLOSSARY

Point-trapping arrangement: An arrangement $\mathcal{A}$ such that every component of the complement of the union of the members of $\mathcal{A}$ is bounded.

Connected arrangement: An arrangement $\mathcal{A}$ such that the union of the members of $\mathcal{A}$ is connected.

j-impassable arrangement: An arrangement $\mathcal{A}$ such that every j-dimensional flat intersects the interior of a member of $\mathcal{A}$.

Obviously, a point-trapping arrangement of congruent copies of a body can be arbitrarily thin. On the other hand, Bárány, Böröczky, Makai, and Pach showed that the density of a point-trapping *lattice* arrangement of any convex body in $\mathbb{E}^d$ is greater than or equal to $1/2$. For $d \geq 3$, equality is attained only in the "checkerboard" arrangement of parallelotopes (see [FK93b]).

Bleicher (see [FK93b]) showed that the minimum density of a point-trapping lattice of unit balls in $\mathbb{E}^3$ is equal to

$$32\sqrt{(7142 + 1802\sqrt{17})^{-1}} = 0.265\ldots .$$

The extreme lattice is generated by three vectors of length $\frac{1}{2}\sqrt{7+\sqrt{17}}$, any two of which make an angle of $\arccos\frac{\sqrt{17}-1}{8} = 67.021\ldots^\circ$

For a convex body K, let $c(K)$ denote the minimum density of a connected lattice arrangement of congruent copies of K. According to a theorem of Groemer

(see [FK93b]),

$$\frac{1}{d!} \leq c(K) \leq \frac{\pi^{d/2}}{2^d \Gamma(1+d/2)} \qquad \text{for } K \in \mathcal{K}^d.$$

The lower bound is attained when K is a simplex or cross-polytope, and the upper bound is attained for a ball.

For a given convex body K in $\mathbb{E}^d$, let $\varrho_j(K)$ denote the infimum of the densities of all j-impassable lattice arrangements of copies of K. Obviously, $\varrho_0(K) = \vartheta_L(K)$. Let $\widehat{K} = (K-K)^*$ denote the polar body of the difference body of K. Between $\varrho_{d-1}(K)$ and $\delta_L(\widehat{K})$ Makai (see [FK93b]) found the following surprising connection:

$$\varrho_{d-1}(K)\delta_L(\widehat{K}) = 2^d V(K)V(\widehat{K}).$$

Little is known about $\varrho_j(K)$ for $0 < j < d-1$. The value of $\varrho_1(B^3)$ has been determined recently [BW94]. We have

$$\varrho_1(B^3) = 9\pi/32 = 0.8835\ldots .$$

An extreme lattice is generated by the vectors $\frac{4}{3}(1,1,0)$, $\frac{4}{3}(0,1,1)$, and $\frac{4}{3}(1,0,1)$.

2.7 PACKING AND COVERING WITH SEQUENCES OF CONVEX BODIES

In this section we consider the following problem: Given a convex set K and a sequence $\{C_i\}$ of convex bodies in $\mathbb{E}^d$, is it possible to find rigid motions σ_i such that $\{\sigma_i C_i\}$ covers K, or forms a packing in K? If there are such motions σ_i, then we say that the sequence $\{C_i\}$ permits an ***isometric covering*** of K, or an ***isometric packing*** in K, respectively. If there are not only rigid motions but even translations τ_i so that $\{\tau_i C_i\}$ is a covering of K, or a packing in K, then we say that $\{C_i\}$ permits a ***translative covering*** of K, or a ***translative packing*** in K, respectively.

First we consider translative packings and coverings of cubes by sequences of boxes. By a ***box*** we mean an orthogonal parallelotope whose sides are parallel to the coordinate axes. We let $I^d(s)$ denote a cube of side s in $\mathbb{E}^d$.

Groemer (see [Gro85]) proved that a sequence $\{C_i\}$ of boxes whose edge lengths are at most 1 permits a translative covering of $I^d(s)$ if

$$\sum V(C_i) \geq (s+1)^d - 1,$$

and that it permits a translative packing in $I^d(s)$ if

$$\sum V(C_i) \leq (s-1)^d - \frac{s-1}{s-2}((s-1)^{d-2} - 1).$$

Slightly stronger conditions (see [Las97]) guarantee even the existence of on-line algorithms for the determination of the translations τ_i. This means that the determination of τ_i is based only on C_i and the previously fixed sets $\tau_i C_i$.

We recall (see [Las97]) that to any convex body K in $\mathbb{E}^d$ there exist two boxes, say Q_1 and Q_2, with $V(Q_1) \geq 2d^{-d}V(K)$ and $V(Q_2) \leq d!V(K)$, such that $Q_1 \subset$

$K \subset Q_2$. It follows immediately that if $\{C_i\}$ is a sequence of convex bodies in $\mathbb{E}^d$ whose diameters are at most 1 and

$$\sum V(C_i) \geq \frac{1}{2} d^d((s+1)^d - 1),$$

then $\{C_i\}$ permits an isometric covering of $I^d(s)$; and that if

$$\sum V(C_i) \leq \frac{1}{d!}\left((s-1)^d - \frac{s-1}{s-2}((s-1)^{d-2} - 1)\right),$$

then it permits an isometric packing in $I^d(s)$.

The sequence $\{C_i\}$ of convex bodies is ***bounded*** if the set of the diameters of the bodies is bounded. As further consequences of the results above we mention the following. If $\{C_i\}$ is a bounded sequence of convex bodies such that $\sum V(C_i) = \infty$, then it permits an isometric covering of $\mathbb{E}^d$ with density $\frac{1}{2}d^d$ and an isometric packing in $\mathbb{E}^d$ with density $\frac{1}{d!}$. Moreover, if all the sets C_i are boxes, then $\{C_i\}$ permits a translative covering of $\mathbb{E}^d$ and a translative packing in $\mathbb{E}^d$ with density 1.

In $\mathbb{E}^2$, any bounded sequence $\{C_i\}$ of convex disks with $\sum a(C_i) = \infty$ permits even a translative packing and covering with density $\frac{1}{2}$ and 2, respectively. It is an open problem whether for $d > 2$ any bounded sequence $\{C_i\}$ of convex bodies in $\mathbb{E}^d$ with $\sum V(C_i) = \infty$ permits a translative covering. If the sequence $\{C_i\}$ is unbounded, then the condition $\sum V(C_i) = \infty$ no longer suffices for $\{C_i\}$ to permit even an isometric covering of the space. For example, if C_i is the rectangle of side lengths i and $\frac{1}{i^2}$, then $\sum a(C_i) = \infty$ but $\{C_i\}$ does not permit an isometric covering of $\mathbb{E}^2$. There is a simple reason for this, which brings us to one of the most interesting topics of this subject, namely Tarski's plank problem.

A ***plank*** is a region between two parallel hyperplanes. Tarski conjectured that if a convex body of minimum width w is covered by a collection of planks in $\mathbb{E}^d$, then the sum of the widths of the planks is at least w. Tarski's conjecture was first proved by Bang. Bang's theorem immediately implies that the sequence of rectangles above does not permit an isometric covering of $\mathbb{E}^2$, not even of $(\frac{\pi^2}{12} + \epsilon)B^2$.

There is a nice account of the history of Tarski's plank problem and its generalizations in [Gro85]. In his paper, Bang asked whether his theorem can be generalized so that the width of each plank is measured relative to the width of the convex body being covered, in the direction normal to the plank. Bang's problem has been solved for centrally symmetric bodies by Ball [Bal91]. This case has a particularly appealing formulation in terms of normed spaces:

If the unit ball in a Banach space is covered by a countable collection of planks, then the total width of the planks is at least 2.

2.8 SOURCES AND RELATED MATERIAL

SURVEYS

The monographs [Fej72] and [Rog64] are devoted solely to packing and covering; also the books [CS93], [CFG91], [EGH89], [Fej64], [GL87], and [PA95] contain re-

sults relevant to this chapter. Additional material and bibliography can be found in the following surveys: [Bar69], [Fej83], [FK93c], [FK93b], [Fej84], [Few67], [Flo87], [GW93], [Gro85], [Gru79], [MP93], and [SA75].

RELATED CHAPTERS

Chapter 3: Tilings
Chapter 7: Lattice points and lattice polytopes
Chapter 10: Geometric discrepancy theory and uniform distribution
Chapter 16: Symmetry of polytopes and polyhedra
Chapter 18: Polyhedral maps
Chapter 50: Sphere packing and coding theory
Chapter 51: Crystals and quasicrystals

REFERENCES

[Alo95] N. Alon. Packings with large minimum kissing numbers. 1995. Preprint.

[Bal91] K. Ball. The plank problem for symmetric bodies. *Invent. Math.*, 104:535–543, 1991.

[Bar69] E.P. Baranovskiĭ. Packings, coverings, partitionings and certain other distributions in spaces of constant curvature (Russian). *Itogi Nauki—Ser. Mat. (Algebra, Topologiya, Geometriya)*, 14:189–225, 1969. Translated in *Progr. Math.*, 9:209–253, 1971.

[Bez93] K. Bezdek. Hadwiger-Levi's covering problem revisited. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, pages 199–233. Springer-Verlag, New York, 1993.

[Bez94] A. Bezdek. A remark on the packing density in the 3-space. In K. Böröczky and G. Fejes Tóth, editors, *Intuitive Geometry*, volume 63 of *Colloq. Math. Soc. János Bolyai*, pages 17–22. North-Holland, Amsterdam/New York, 1994.

[BH] U. Betke and M. Henk. Finite packings of spheres. *Discrete Comput. Geom.*, to appear.

[BHW94] U. Betke, M. Henk, and J.M. Wills. Finite and infinite packings. *J. Reine Angew. Math.*, 453:165–191, 1994.

[BKK95] A. Bezdek, K. Kuperberg, and W. Kuperberg. Mutually contiguous and concurrent translates of a plane disk. *Duke Math. J.*, 78:19–31, 1995.

[BW94] R.P. Bambah and A.C. Woods. On a problem of G. Fejes Tóth. *Proc. Indian Acad. Sci. Math. Sci.*, 104:137–156, 1994.

[CFG91] H.T. Croft, K.J. Falconer, and R.K. Guy. *Unsolved Problems in Geometry.* Springer-Verlag, New York, 1991.

[CS93] J.H. Conway and N.J.A. Sloane. *Sphere Packings, Lattices and Groups*, 2nd edition. Springer-Verlag, New York, 1993.

[DLR92] P.G. Doyle, J.C. Lagarias, and D. Randall. Self-packing of centrally symmmetric convex discs in R^2. *Discrete Comput. Geom.*, 8:171–189, 1992.

[EGH89] P. Erdős, P.M. Gruber, and J. Hammer. *Lattice Points*, number 39 of *Pitman Monographs*. Longman Scientific/John Wiley, New York, 1989.

[Fej64] L. Fejes Tóth. *Regular Figures.* Pergamon, Oxford, 1964.

[Fej72] L. Fejes Tóth. *Lagerungen in der Ebene auf der Kugel und im Raum*, 2nd edition. Springer-Verlag, Berlin/New York, 1972.

[Fej83] G. Fejes Tóth. New results in the theory of packing and covering. In P.M. Gruber and J.M. Wills, editors, *Convexity and its Applications*, pages 318–359. Birkhäuser, Basel/Boston/Stuttgart, 1983.

[Fej84] L. Fejes Tóth. Density bounds for packing and covering with convex discs. *Exposition. Math.*, 2:131–153, 1984.

[Fej95] G. Fejes Tóth. Densest packings of typical convex sets are not lattice-like. *Discrete Comput. Geom.*, 14:1–8, 1995.

[Few67] L. Few. Multiple packing of spheres: a survey. In *Proc. Colloquium Convexity (Copenhagen 1965)*, pages 88–93. Københavns Univ. Mat. Inst., 1967.

[FK93a] G. Fejes Tóth and W. Kuperberg. Blichfeldt's density bound revisited. *Math. Ann.*, 295:721–727, 1993.

[FK93b] G. Fejes Tóth and W. Kuperberg. Packing and covering with convex sets. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 799–860. North-Holland, Amsterdam, 1993.

[FK93c] G. Fejes Tóth and W. Kuperberg. Recent results in the theory of packing and covering. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, pages 251–279. Springer-Verlag, New York, 1993.

[FK95] G. Fejes Tóth and W. Kuperberg. Thin non-lattice covering with an affine image of a strictly convex body. *Mathematika*, 42:239–250, 1995.

[FKK] G. Fejes Tóth, G. Kuperberg, and W. Kuperberg. Highly saturated packings and reduced coverings. *Monatsh. Math.*, to appear.

[Flo87] A. Florian. Packing and covering with convex discs. In K. Böröczky and G. Fejes Tóth, editors, *Intuitive Geometry (Siófok, 1985)*, volume 48 of *Colloq. Math. Soc. János Bolyai*, pages 191–207. North-Holland, Amsterdam/New York, 1987.

[FZ94] G. Fejes Tóth and T. Zamfirescu. For most convex discs thinnest covering is not lattice-like. In K. Böröczky and G. Fejes Tóth, editors, *Intuitive Geometry*, volume 63 of *Colloq. Math. Soc. János Bolyai*, pages 105–108. North-Holland, Amsterdam/New York, 1994.

[GL87] P.M. Gruber and C.G. Lekkerkerker. *Geometry of Numbers*. Elsevier, North-Holland, Amsterdam, 1987.

[Gro85] H. Groemer. Coverings and packings by sequences of convex sets. In J.E. Goodman, E. Lutwak, J. Malkevitch, and R. Pollack, editors, *Discrete Geometry and Convexity*, volume 440 of *Annals of the New York Academy of Sciences*, pages 262–278. 1985.

[Gru79] P.M. Gruber. Geometry of numbers. In J. Tölke and J.M. Wills, editors, *Contributions to Geometry*, Proc. Geom. Symp. (Siegen, 1978), pages 186–225. Birkhäuser, Basel/Boston/Stuttgart, 1979.

[GW93] P. Gritzmann and J.M. Wills. Finite packing and covering. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 861–897. North-Holland, Amsterdam, 1993.

[HM] A. Heppes and J.B.M. Melissen. Covering a rectangle with equal circles. *Period. Math. Hungar.*, 34:63–79, 1997.

[HSS] R.J. Hardin, N.J.A. Sloane, and W.D. Smith. *Spherical Codes*. Book in preparation.

[Ism] D. Ismailescu. Covering the plane with copies of a convex disc. *Discrete Comput. Geom.*, to appear.

[Ker94] G. Kertész. Nine points on the hemisphere. In K. Böröczky and G. Fejes Tóth, editors, *Intuitive Geometry*, volume 63 of *Colloq. Math. Soc. János Bolyai*, pages 189–196. North-Holland, Amsterdam/New York, 1994.

[KK93] J. Kahn and G. Kalai. A counterexample to Borsuk's conjecture. *Bull. Amer. Math. Soc.*, 29:60–62, 1993.

[Kro93] S. Krotoszyński. Covering a disc with smaller discs. *Studia Sci. Math. Hungar.*, 28:271–283, 1993.

[Las97] M. Lassak. A survey of algorithms for on-line packing and covering by sequences of convex bodies. In I. Bárány and K. Böröczky, editors, *Intuitive Geometry*, volume 6 of *Bolyai Soc. Math. Studies*, pages 129–157. János Bolyai Math. Soc., Budapest, 1997.

[Mel93] J.B.M. Melissen. Densest packings of congruent circles in an equilateral triangle. *Amer. Math. Monthly*, 100:816–825, 1993.

[Mel94] J.B.M. Melissen. Densest packings of eleven congruent circles in a circle. *Geom. Dedicata*, 50:15–25, 1994.

[Mel97] J.B.M. Melissen. Loosest circle coverings of an equilateral triangle. *Math. Mag.*, 70:119–125, 1997.

[MP93] W. Moser and J. Pach. Research problems in discrete geometry. Report 93-32., DIMACS, Rutgers, 1993.

[Mud93] D.J. Muder. A new bound on the local density of sphere packings. *Discrete Comput. Geom.*, 10:351–375, 1993.

[PA95] J. Pach and P.K. Agarwal. *Combinatorial Geometry.* John Wiley, New York, 1995.

[Pei94] R. Peikert. Dichteste Packung von gleichen Kreisen in einem Quadrat. *Elem. Math.*, 49:16–26, 1994.

[Rog64] C.A. Rogers. *Packing and Covering.* Cambridge University Press, Cambridge, 1964.

[SA75] T.L. Saaty and J.M. Alexander. Optimization and the geometry of numbers: packing and covering. *SIAM Rev.*, 17:475–519, 1975.

[Sch88] P. Schmitt. An aperiodic prototile in space. 1988. Preprint.

[Sch91] P. Schmitt. Disks with special properties of densest packings. *Discrete Comput. Geom.*, 6:181–190, 1991.

[Sch94] J. Schaer. The densest packing of ten congruent spheres in a cube. In K. Böröczky and G. Fejes Tóth, editors, *Intuitive Geometry*, volume 63 of *Colloq. Math. Soc. János Bolyai*, pages 403–424. North-Holland, Amsterdam/New York, 1994.

[Tem94a] Á. Temesvári. Die dichteste gitterförmige 9-fache Kreispackung. *Rad. Hrvatske Akad. Znan. Umj. Mat.*, 11:95–110, 1994.

[Tem94b] Á. Temesvári. Die dünnste 8-fache gitterförmige Kreisüberdeckung der Ebene. *Studia Sci. Math. Hungar.*, 29:323–340, 1994.

3 TILINGS

Doris Schattschneider and Marjorie Senechal

INTRODUCTION

Tilings of surfaces and packings of space have been of interest to artisans and manufacturers throughout history; they are a means of artistic expression and lend economy and strength to modular constructions. Today scientists and mathematicians study tilings because they pose interesting mathematical questions and provide mathematical models for such diverse structures as the molecular anatomy of crystals, cell packings of viruses, n-dimensional algebraic codes, and "nearest neighbor" regions for a set of discrete points. The basic questions are: What bodies can tile space? In what ways do they tile? However, in this generality such questions are intractable. The tiles and tilings that are studied must be subject to constraints.

Even with constraints the subject is unmanageably large. In this chapter we restrict ourselves, for the most part, to tilings of unbounded spaces. In the next section we present some general results that are fundamental to the subject as a whole. Section 3.2 addresses tilings with congruent tiles. In Section 3.3 we discuss the classical subject of periodic tilings, which continues to be enriched with new results. Next, we briefly describe the newer theory of nonperiodic and aperiodic tilings, both of which are discussed in more detail in Chapter 51. We conclude with a very brief description of some kinds of tilings not considered here.

3.1 GENERAL CONSIDERATIONS

In this section we define terms that will be used throughout the chapter and state some basic results. Taken together, these results state that although there is no algorithm for deciding which bodies are tiles, there are criteria for deciding the question in certain cases. We can obtain some quantitative information about the tiling in particularly well-behaved cases.

Unless otherwise stated, we assume that S is an n-dimensional space, either Euclidean ($\mathbb{E}^n$) or hyperbolic. We also assume that the tiles are bounded and the tilings are locally finite (see the Glossary below). Throughout this chapter, n is the dimension of the space in which we are working.

GLOSSARY

Body: A bounded region (of S) that is the closure of its (nonempty) interior.

Tiling (of S): A decomposition of S into a countable number of n-dimensional bodies whose interiors are pairwise disjoint. In this context, the bodies are also called n-cells and are the tiles of the tiling (see below). Synonyms: tessellation, parquetry (when $n = 2$), honeycomb (for $n \geq 2$).

Tile: A body that is an n-cell of one or more tilings of S. To say that a body ***tiles*** a region $R \subseteq S$ means that R can be covered exactly by copies of the body without gaps or overlaps.

Locally-finite tiling: Every n-ball of finite radius in S meets only finitely many tiles of the tiling.

Prototile set (for a tiling $\mathcal{T}$ of S): A minimal subset of tiles in $\mathcal{T}$ such that each tile in the tiling $\mathcal{T}$ is the congruent image of one of those in the prototile set. The tiles in the set are called prototiles and the prototile set is said to admit $\mathcal{T}$.

k-face (of a tiling): An intersection of at least $n-k+1$ tiles of the tiling that is not contained in a j-face for $j < k$. (The 0-faces are the ***vertices*** and 1-faces the ***edges***; the $(n-1)$-faces are simply called the ***faces*** of the tiling.)

Patch (in a tiling): A set of tiles whose union is homeomorphic to an n-ball. See Figure 3.1.1. A spherical patch $P(r, s)$ is the set of tiles whose intersection with the ball of radius r centered at s is nonempty, together with any additional tiles needed to complete the patch (that is, to make it homeomorphic to an n-ball).

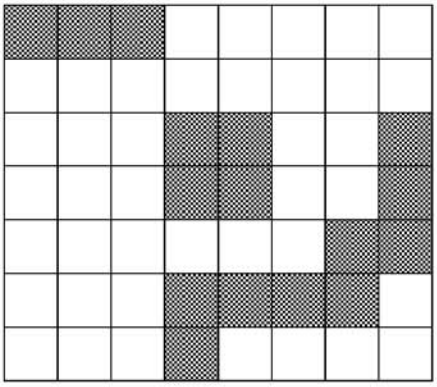


FIGURE 3.1.1
Three patches in a tiling of the plane by squares.

Normal tiling: A tiling in which (i) each prototile is homeomorphic to an n-ball, and (ii) the prototiles are uniformly bounded (there exist $r > 0$ and $R > 0$ such that each prototile contains a ball of radius r and is contained in a ball of radius R). It is technically convenient to include a third condition: (iii) the intersection of every pair of tiles is a connected set. (A normal tiling is necessarily locally finite.)

Face-to-face tiling (by polytopes): A tiling in which the faces of the tiling are also the $(n-1)$-dimensional faces of the polytopes. (A face-to-face tiling by convex polytopes is also k-face-to-k-face for $0 \leq k \leq n-1$.) In dimension 2, this is an ***edge-to-edge*** tiling by polygons, and in dimension 3, a face-to-face tiling by polyhedra.

Dual tiling: Two tilings $\mathcal{T}$ and $\mathcal{T}^*$ are dual if there is an incidence-reversing bijection between the k-faces of $\mathcal{T}$ and the $(n-k)$-faces of $\mathcal{T}^*$ (see Figure 3.1.2).

Voronoi (Dirichlet) tiling: A tiling whose tiles are the Voronoi cells of a discrete set Λ of points in S. The Voronoi cell of a point $p \in \Lambda$ is the set of all points

in S that are at least as close to p as to any other point in Λ (see Chapter 20).

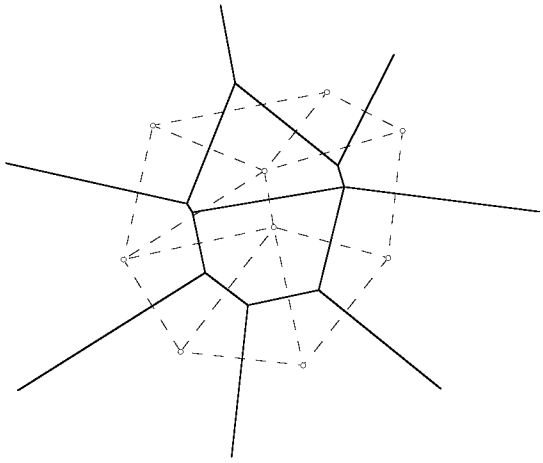


FIGURE 3.1.2
A Voronoi tiling (solid lines) and its Delaunay dual (dashed lines).

Delaunay (or Delone) tiling: A face-to-face tiling by convex circumscribable polytopes (i.e., the vertices of each polytope lie on a sphere).

Isometry: A distance-preserving self-map of S.

Symmetry group (of a tiling): The set of isometries of S that map the tiling to itself.

MAIN RESULTS

1. **The Undecidability Theorem.** There is no algorithm for deciding whether or not an arbitrary body or set of bodies admits a tiling of S.

2. **The Extension Theorem (for $\mathbb{E}^n$).** Let A be any finite set of bodies, each homeomorphic to a closed n-ball. If A tiles regions that contain arbitrarily large n-balls, then A admits a tiling of $\mathbb{E}^n$. (These regions need not be nested, nor need any of the tilings of the regions be extendable!)

3. **The Normality Lemma (for $\mathbb{E}^n$).** In a normal tiling, the ratio of the number of tiles that meet the boundary of a spherical patch to the number of tiles in the patch tends to zero as the radius of the patch tends to infinity. In fact, a stronger statement can be made: For $s \in S$ let $t(r, s)$ be the number of tiles in the spherical patch $P(r, s)$. Then, in a normal tiling, for every $x > 0$,

$$\lim_{r\to\infty} \frac{t(r + x, s) - t(r, s)}{t(r, s)} = 0.$$

4. **Euler's Theorem for tilings of $\mathbb{E}^2$.** Let $\mathcal{T}$ be a normal tiling of $\mathbb{E}^2$, and let $t(r, s)$, $e(r, s)$, and $v(r, s)$ be the numbers of tiles, edges, and vertices, respectively, in the circular patch $P(r, s)$. Then if one of the limits

$e(\mathcal{T}) = \lim_{r\to\infty} e(r,s)/t(r,s)$ or $v(\mathcal{T}) = \lim_{r\to\infty} v(r,s)/t(r,s)$ exists, so does the other, and $v(\mathcal{T}) - e(\mathcal{T}) + 1 = 0$. Like Euler's Theorem for Planar Maps, on which the proof of this theorem is based, this result can be extended in various ways.

5. **Voronoi Dual.** Every Voronoi tiling has a Delaunay dual, and conversely (see Figure 3.1.2).

3.2 TILINGS BY ONE TILE

To say that a body tiles $\mathbb{E}^n$ usually means that there is a tiling all of whose tiles are copies of this body. The artist M.C. Escher has demonstrated how intricate such tiles can be even when $n = 2$. But in higher dimensions the simplest tiles—for example, cubes—can produce surprises, as the recent counterexample to Keller's conjecture attests (see below).

GLOSSARY

Monohedral tiling: A tiling with a single prototile.

r-morphic tile: A prototile that admits exactly r distinct monohedral tilings. Figure 3.2.1 shows a 5-morphic tile and all its tilings, and Figure 3.2.3 shows a 1-morphic tile and its tilings.

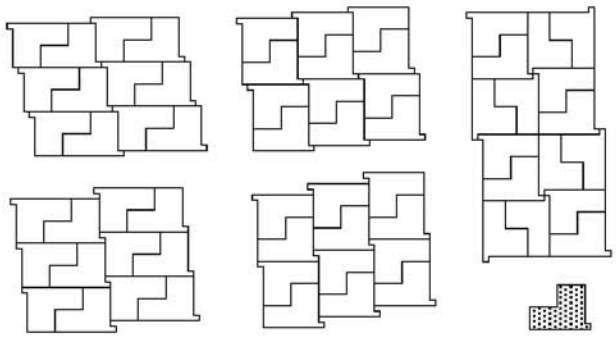


FIGURE 3.2.1
A pentamorphic tile

k-rep tile: A body for which k copies can be assembled into a larger, similar body. (Or, equivalently, a body that can be partitioned into k congruent bodies, each similar to the original.) More formally, a k-rep tile is a closed set A_1 in S with nonempty interior such that there are sets $A_2, \ldots, A_k$ congruent to A_1 that satisfy

$$\operatorname{Int} A_i \cap \operatorname{Int} A_j = \emptyset$$

for all $i \neq j \text{and} A_1 \cup .... \cup A_k = g(A_1)$, where g is a similarity mapping. (Figure 3.2.2 shows a 3-dimensional chair rep tile and the second-level chair. An n-dimensional chair rep tile can be formed in a similar manner.)

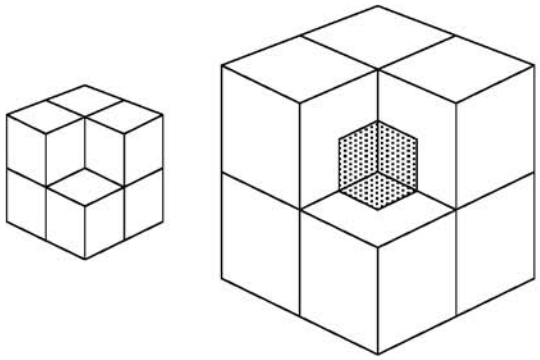


FIGURE 3.2.2
A 3-dimensional chair rep tile and a second-level chair in which seven copies surround the first.

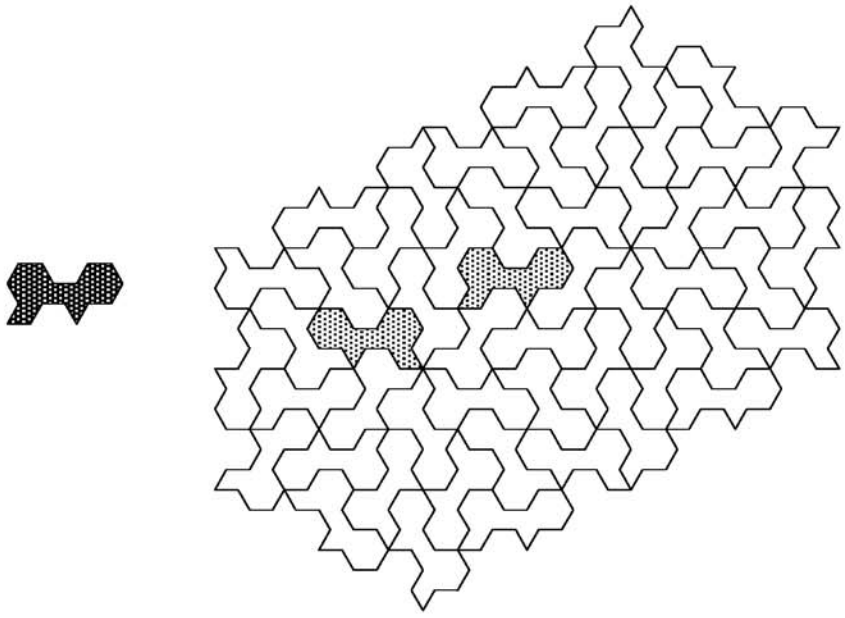
Transitive action: A group G is said to act transitively on a set $\{A_1, A_2, \ldots\}$ if the set is an orbit for G. (That is, for every pair A_i, A_j of elements of the set, there is a $g_{ij} \in G$ such that $g_{ij}A_i = A_j$.)

Regular system of points: A discrete set of points on which an infinite group of isometries acts transitively.

Isohedral (tiling): A tiling whose symmetry group acts transitively on its tiles.

Anisohedral tile: A prototile that admits monohedral tilings but no isohedral tilings. In Figure 3.2.3, the prototile admits a unique nonisohedral tiling; the shaded tiles are each surrounded differently.

FIGURE 3.2.3
An anisohedral tile (due to R. Penrose) and its unique tiling in which tiles are surrounded in two different ways.



Corona (of a tile P in a tiling $\mathcal{T}$): Define $C^0(P) = P$. Then $C^k(P)$, the k^{th} corona of P, is the set of all tiles $Q \in T$ for which there exists a path of tiles $P = P_0, P_1, \ldots, P_m = Q$ with $m \leq k$ in which $P_i \cap P_{i+1} \neq \emptyset$, $i = 0, 1, \ldots, m-1$.

Lattice: The group of integral linear combinations of n linearly independent vectors in S. A point orbit of a lattice, often called a ***point lattice***, is a particular case of a regular system of points.

Translation tiling: A monohedral tiling of S in which every tile is a translate of a fixed prototile. See Figure 3.2.4.

Lattice tiling: A monohedral tiling on whose tiles a lattice acts transitively. Figure 3.2.4 is not a lattice tiling since it is invariant by multiples of just one vector.

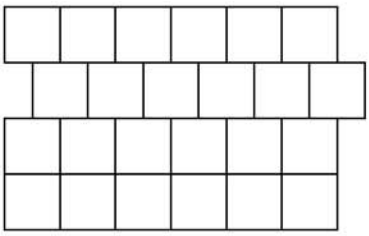


FIGURE 3.2.4
A translation nonlattice tiling.

n-parallelotope: A convex n-polytope that tiles $\mathbb{E}^n$ by translation.

Belt (of an n-parallelotope): A maximal subset of parallel $(n-2)$-faces of a parallelotope in $\mathbb{E}^n$. The number of $(n-2)$-faces in a belt is its length.

Center of symmetry (for a set A in $\mathbb{E}^n$): A point $a \in A$ such that A is invariant under the mapping $x \to 2a - x$; the mapping is called ***central inversion*** and an object that has a center of symmetry is said to be ***centrosymmetric***.

Stereohedron: A convex polytope that is the prototile of an isohedral tiling. A Voronoi cell of a regular system of points is a stereohedron.

Linear expansive map: A linear transformation all of whose eigenvalues have modulus greater than one.

MAIN RESULTS

1. **The Local Theorem**. Let $\mathcal{T}$ be a monohedral tiling of S with prototile P, and let $S_i(P)$ be the subgroup of the symmetry group of P that leaves invariant $C^i(P)$, the i^{th} corona of P. $\mathcal{T}$ is isohedral if and only if there exists an integer $k > 0$ for which the following two conditions hold: (a) $S_{k-1}(P) = S_k(P)$ and (b) For every pair of tiles P, P' in $\mathcal{T}$, there exists an isometry γ such that $\gamma(P) = P'$ and $\gamma(C^k(P)) = C^k(P')$. In particular, if P is asymmetric, then $\mathcal{T}$ is isohedral if and only if condition (b) holds for $k = 1$. See [DS97].

2. A convex polytope is a parallelotope if and only if it is centrosymmetric, its faces are centrosymmetric, and its belts have lengths four or six. First proved by Venkov, this theorem was rediscovered independently by McMullen.

3. The number $|F|$ of faces of a convex parallelotope in $\mathbb{E}^n$ satisfies Minkowski's inequality, $2n \leq |F| \leq 2(2^n - 1)$. Both upper and lower bounds are realized in every dimension.

4. The number of faces of an n-dimensional stereohedron in $\mathbb{E}^n$ is bounded. In fact, if a is the number of translation classes of the stereohedron in an isohedral tiling, then the number of faces is at most the Delaunay bound $2^n(1+a)-2$.

5. Using a classification system that takes into account the symmetry groups of the tilings and their tiles, the combinatorial structure of the tiling, and the ways in which the tiles are related to adjacent tiles, Grünbaum and Shephard proved that there are 81 classes of isohedral tilings of $\mathbb{E}^2$, 93 classes if the tiles are ***marked*** (that is, they have decorative markings to express symmetry in addition to the tile shape). There is an infinite number of classes of isohedral tilings of $\mathbb{E}^n$, $n > 2$.

6. Anisohedral tiles exist in $\mathbb{E}^n$ for every $n \geq 2$. (The first example, given for $n = 3$ by Reinhardt, was the solution to part of Hilbert's 18th problem.)

7. Every n-parallelotope admits a lattice tiling. However, for $n \geq 3$, nonconvex tiles have been found that tile by translation but do not admit lattice tilings [SS94].

8. In a lattice tiling of $\mathbb{E}^n$ by unit cubes there must be a pair of cubes that share a whole face. However, a famous conjecture of Keller, which stated that for every n, any tiling of $\mathbb{E}^n$ by congruent cubes must contain at least one pair of cubes that share a whole face, is false: for $n \geq 10$, there exist translation tilings by unit cubes in which no two cubes share a whole face [LS92].

9. Every linear expansive map that transforms the lattice $\mathbb{Z}^n$ of integer vectors into itself defines a family of k-rep tiles; these tiles, which usually have fractal boundaries, admit lattice tilings [Ban91].

OPEN PROBLEMS

1. Which convex n-polytopes in $\mathbb{E}^n$ are prototiles for monohedral tilings of $\mathbb{E}^n$? This is unsolved for all $n \geq 2$ (see [GS87] for the case $n = 2$; the list of convex pentagons that tile has not been proved complete). For higher dimensions, little is known; it is not even known which tetrahedra tile $\mathbb{E}^3$.

2. **Heesch's Problem.** Is there an integer k_n, depending only on the dimension n of the space S, such that if a body A can be completely surrounded k_n times by tiles congruent to A, then A is a prototile for a monohedral tiling of S? (A is completely surrounded if A, together with congruent copies that have nonempty intersection with A, tile a region R that is homeomorphic to an n-ball and A is in the interior of R.) When $S = \mathbb{E}^2$, $k_2 > 3$ (the body shown in Figure 3.2.5 can be completely surrounded three times but not four). This problem is unsolved for all n.

3. **Keller's conjecture** is true for $n \leq 6$ and false for $n \geq 10$ (see above). The cases $n = 7, 8$, and 9 are still open.

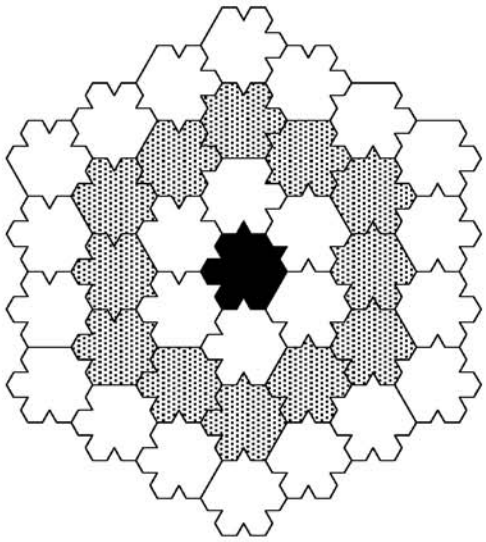


FIGURE 3.2.5
Robert Ammann's 3-corona tile.

4. Do r-morphic tiles exist for every positive integer r? Fontaine and Martin have shown the answer is yes in $\mathbb{E}^2$ for $r \leq 10$.

5. Find a good upper bound for the number of faces of an n-dimensional stereohedron. Delaunay's bound, stated above, is evidently much too high (for example, it gives 390 as the bound in $\mathbb{E}^3$, while the maximal known number of faces of a three-dimensional stereohedron (found by P. Engel) is 38).

6. For monohedral (face-to-face) tilings by convex polytopes there is an integer k_n, depending only on the dimension n of S, that is an upper bound for the constant k in the Local Theorem. Find the value of this k_n. For the Euclidean plane $\mathbb{E}^2$ it is known that $k_2 = 1$ (convexity of the tiles is not necessary), but for the hyperbolic plane, $k_2 \geq 2$. For $\mathbb{E}^3$, it is known that $2 \leq k_3 \leq 5$.

3.3 PERIODIC TILINGS

Periodic tilings have been studied intensely, in part because their applications range from ornamental design to crystallography, and in part because many techniques (algebraic, geometric, and combinatorial) are available for studying them.

GLOSSARY

Periodic tiling of $\mathbb{E}^n$***:*** A tiling, not necessarily monohedral, whose symmetry group contains an n-dimensional lattice. This definition can be adapted to include "subperiodic" tilings (those whose symmetry groups contain $1 \leq k < n$ linearly independent vectors) and tilings of other spaces (for example, cylinders). Tilings in Figures 3.2.1, 3.2.3, 3.3.1, and 3.3.4 are periodic.

Fundamental domain (generating region) for a periodic tiling: A minimal subset of S whose orbit under the symmetry group of the tiling is the whole

tiling. A fundamental domain may be a tile (Figure 3.2.1), a subset of a single tile (Figure 3.3.4), or a subset of patches of tiles (two shaded tiles in Figure 3.2.3).

Orbifold (of a tiling of S): The manifold obtained by identifying points of S that are in the same orbit under the action of the symmetry group of the tiling.

Free tiling: A tiling whose symmetry group acts freely and transitively on the tiles.

k-isohedral (tiling): A tiling whose tiles belong to k transitivity classes under the action of its symmetry group. Isohedral means 1-isohedral (Figures 3.3.1 and 3.3.4). The tiling in Figure 3.2.3 is 2-isohedral.

Equitransitive (tiling by polytopes): A tiling in which each combinatorial class of tiles forms a single transitivity class under the action of the symmetry group of the tiling.

k-isogonal (tiling): A tiling whose vertices belong to k transitivity classes under the action of its symmetry group. Isogonal means 1-isogonal.

k-uniform (tiling of a 2-dimensional surface): A k-isogonal tiling by regular polygons.

Uniform (tiling for $n > 2$): An isogonal tiling with congruent edges and uniform faces.

Flag of a tiling (of S): An ordered $(n+1)$-tuple $(X_0, X_1, ..., X_n)$, with X_n a tile and X_k a k-face for $0 \leq k \leq n-1$, in which $X_{i-1} \subset X_i$ for $i = 1, \ldots, n$.

Regular tiling (of S): A tiling $\mathcal{T}$ whose symmetry group is transitive on the flags of $\mathcal{T}$. (For $n > 2$, these are also called regular honeycombs.) See Figure 3.3.4.

k-colored tiling: A tiling in which each tile has a single color, and k different colors are used. Unlike the case of map colorings, in a colored tiling adjacent tiles may have the same color.

Perfectly k-colored tiling: A k-colored tiling for which each element of the symmetry group G of the uncolored tiling effects a permutation of the colors. The ordered pair (G, Π), where Π is the corresponding permutation group, is called a k-color symmetry group.

CLASSIFICATION OF PERIODIC TILINGS

The mathematical study of tilings has (like most mathematical investigations) been accompanied by the development and use of a variety of notations for classification of different "types" of tilings and tiles. Far from being merely names by which to distinguish types, these notations tell us the investigators' point of view and the questions they ask. Notation may tell us the global symmetries of the tiling, or how each tile is surrounded, or the topology of its orbifold. Notation makes possible the computer implementation of investigations of combinatorial questions about tilings.

Periodic tilings are classified by symmetry groups and, sometimes, by their skeletons (of vertices, edges, ..., $(n-1)$-faces). The groups are known as ***crystallographic groups***; up to isomorphism, there are 17 in $\mathbb{E}^2$ and 219 in $\mathbb{E}^3$. For $\mathbb{E}^2$ and $\mathbb{E}^3$, the most common notation for the groups has been that of the International Union of Crystallography (IUCr). This is cross-referenced to earlier notations in [Sch78]. Two recently developed notations are Delaney-Dress symbols [Dre87] and Conway's orbifold notation [Con92].

GLOSSARY

International symbol (for periodic tilings of $\mathbb{E}^2$ and $\mathbb{E}^3$): Encodes lattice type and particular symmetries of the tiling. In Figure 3.3.1, the lattice unit diagram encodes the symmetries of the tiling and the IUCr symbol p31m indicates that the highest order rotation symmetry in the tiling is 3-fold, that there is no mirror normal to the edge of the lattice unit, and that there is a mirror at 60° to the edge of the lattice unit. These symbols are augmented to denote symmetry groups of perfectly 2-colored tilings.

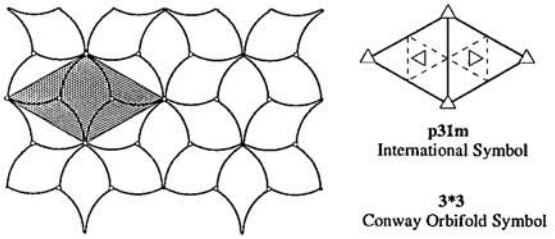


FIGURE 3.3.1
An isohedral tiling with standard IUCr lattice unit shaded; a half-leaf is a fundamental domain. The classification symbols are for the symmetry group of the tiling.

Delaney-Dress symbol (for tilings of Euclidean, hyperbolic, or spherical space of any dimension): Associates an edge-colored and vertex-labeled graph derived from a ***chamber system*** (a formal barycentric subdivision) of the tiling. In Figure 3.3.2, the nodes of the graph represent distinct triangles A, B, C, D in the chamber system, and colored edges (dashed, thick, or thin) indicate their adjacency relations. Numbers on the nodes of the graph show the degree of the tile that contains that triangle and the degree of the vertex of the tiling that is also a vertex of that triangle.

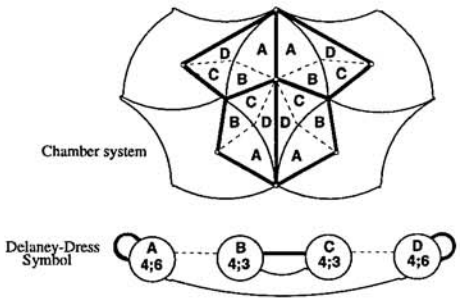
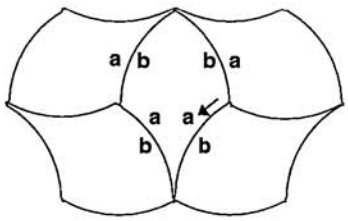


FIGURE 3.3.2
A chamber system of the tiling in Figure 3.3.1 determines the graph that is its Delaney-Dress symbol.

Orbifold notation (for symmetry groups of tilings of 2-dimensional surfaces of constant curvature): Encodes properties of the orbifold induced by the symmetry group of a periodic tiling of the Euclidean plane or hyperbolic plane, or a finite tiling of the surface of a sphere; introduced by Conway. In Figure 3.3.1, the first 3 in the orbifold symbol 3*3 for the symmetry group of the tiling indicates there is a 3-fold rotation center (gyration point) that becomes a cone point in the orbifold, while *3 indicates that the boundary of the orbifold

is a mirror with a corner where three mirrors intersect.



Grünbaum-Shephard Incidence Symbol $[\,a^+a^-b^+b^-;\ b^-a^-\,]$

FIGURE 3.3.3
Labeling and orienting the edges of the isohedral tiling in Figure 3.3.1 determines its Grünbaum-Shephard incidence symbol.

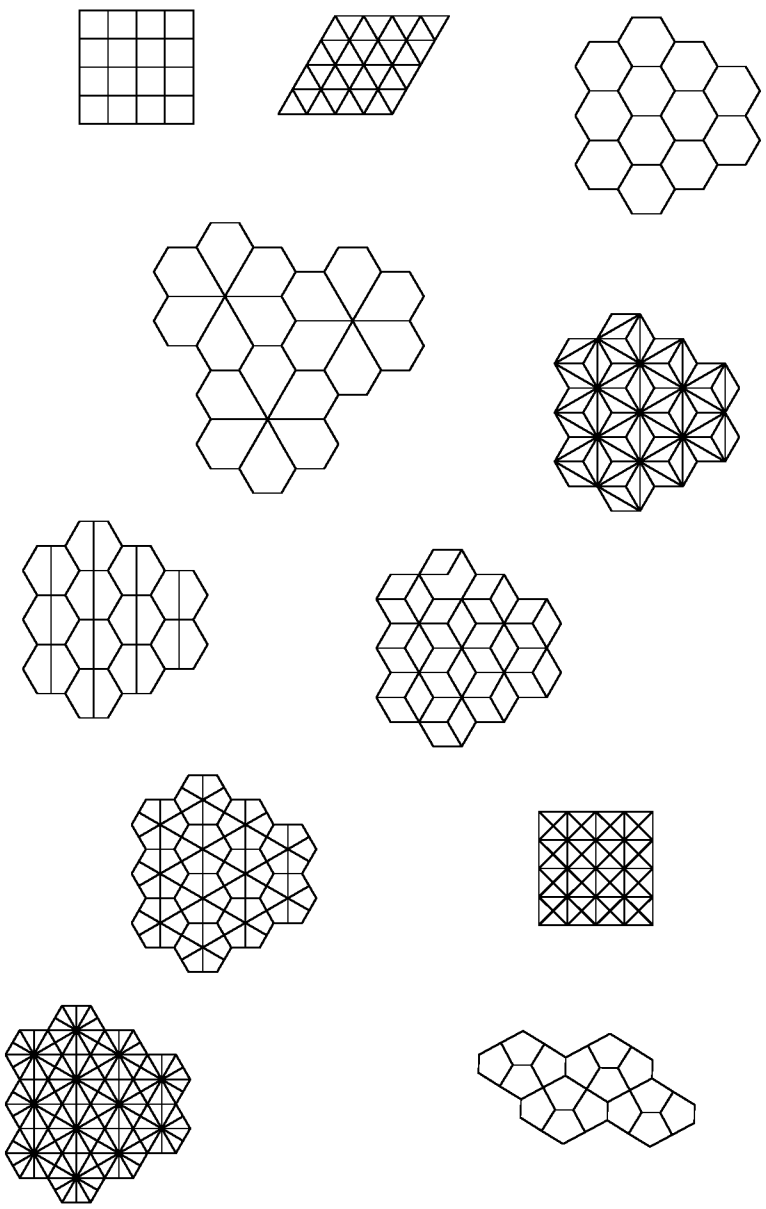
See Table 3.3.1 for the IUCr and orbifold notations for $\mathbb{E}^2$.

TABLE 3.3.1 IUCr and orbifold notations for the 17 symmetry groups of periodic tilings of $\mathbb{E}^2$.

IUCr	ORBIFOLD	IUCr	ORBIFOLD
p1	o or o1	p3	333
pg	×× or 1××	p31m	3*3
cm	*× or 1*×	p3m1	*333
pm	** or 1**	p4	442
p2	2222	p4g	4*2
pgg	22×	p4m	*442
pmg	22*	p6	632
cmm	2*22	p6m	*632
pmm	*2222		

Isohedral tilings of $\mathbb{E}^2$ fall into 11 combinatorial classes, typified by the Laves nets (Figure 3.3.4). The Laves net for the tiling in Figure 3.3.1 is [3.6.3.6]; this gives the vertex degree sequence for each tile. In an isohedral tiling, every tile is surrounded in the same way. Grünbaum and Shephard provide an incidence symbol for each isohedral type by labeling and orienting the edges of each tile [GS79]. Figure 3.3.3 gives the incidence symbol for the tiling in Figure 3.3.1. The tile symbol $a^+a^-b^+b^-$ records the cycle of edges of a tile and their orientations with respect to the (arrowed) first edge (+ indicates the same, − indicates opposite orientation). The adjacency symbol b^-a^- records for each different letter edge of a single tile, beginning with the first, the edge it abuts in the adjacent tile and their relative orientations (now − indicates same, + opposite). These symbols can be augmented to adjacency symbols to denote k-color symmetry groups. Earlier, Heesch devised signatures for the 28 types of tiles that could be fundamental domains of isohedral tilings [HK63].

FIGURE 3.3.4
The 11 *Laves nets. The three regular tilings of* $\mathbb{E}^2$ *are at the top of the illustration.*



MAIN RESULTS

1. Grünbaum and Shephard have shown that if a finite prototile set of polytopes admits a face-to-face tiling of $\mathbb{E}^n$ that has translational symmetry, then the prototile set also admits a periodic tiling.

2. The number of symmetry groups of periodic tilings in $\mathbb{E}^n$ is finite (this is a famous theorem of Bieberbach: see also Chapter 51); the number of symmetry groups of corresponding tilings in hyperbolic n-space is infinite.

3. Every k-isohedral tiling of the Euclidean plane, hyperbolic plane, or sphere can be obtained from a $(k-1)$-isohedral tiling by a process of ***splitting*** (splitting an asymmetric prototile) and ***gluing*** (amalgamating two or more equivalent asymmetric tiles adjacent in the tiling into one new tile); Huson has shown that there are 1270 classes of 2-isohedral tilings and 48,231 classes of 3-isohedral tilings of $\mathbb{E}^2$.

4. Classifying isogonal tilings in a manner analogous to isohedral ones, Grünbaum and Shephard have shown that there are 91 classes of isogonal tilings of $\mathbb{E}^2$ (93 classes if the tiles are marked). Similarly, there are 26 classes of normal tilings of $\mathbb{E}^2$ for which the symmetry group acts transitively on the edges (30 if the tiles are marked); these tilings are called ***isotoxal***.

5. Dress, Molnár, and Huson [DHM93] have shown that there are 88 combinatorial classes of periodic tilings of $\mathbb{E}^3$ for which the symmetry group acts transitively on the faces of the tiling.

6. For every k, the number of k-uniform tilings of $\mathbb{E}^2$ is finite. There are 11 uniform tilings of $\mathbb{E}^2$ (also called ***Archimedean***, or ***semiregular***), of which 3 are regular. The Laves nets in Figure 3.3.4 are duals of these 11 uniform tilings. There are 28 uniform tilings of $\mathbb{E}^3$ and 20 2-uniform tilings of $\mathbb{E}^2$.

7. Danzer, Grünbaum, and Shephard have shown that in any equitransitive tiling of $\mathbb{E}^2$ by convex polygons, the maximum number of edges of any tile is 66 [DGS87].

8. There is a finite number of regular tilings of $\mathbb{E}^n$ (three for $n = 2$, one for $n = 3$, three for $n = 4$, and one for each $n > 4$). There is an infinite number of normal regular tilings of the hyperbolic plane, four regular tilings of hyperbolic 3-space, five regular tilings of hyperbolic 4-space, and no regular tilings of hyperbolic space for dimensions $n > 4$.

9. Balke and Huson have shown that if two orbifold symbols for a tiling of the Euclidean or hyperbolic plane look exactly the same except for the numerical values of their digits, which may differ by a permutation of the natural numbers (such as *632 and *532), then the number of k-isohedral tilings for each of these orbifold types is the same [BH96].

10. There is a one-one correspondence between perfect k-colorings of a free tiling and the subgroups of index k of its symmetry group.

OPEN PROBLEMS

1. Does every convex pentagon that tiles $\mathbb{E}^2$ admit a k-isohedral tiling, and if so, is there an upper bound on k? (All pentagons known to tile the plane admit k-isohedral tilings, with $k \leq 3$.)

2. Classification of uniform tilings of the hyperbolic plane is complete only for the cases of vertex valences of 3 and 4; beyond that, little is known.

3. Enumerate the uniform tilings of $\mathbb{E}^n$ for $n > 3$.

4. Delaney-Dress symbols and orbifold notations have made progress possible on the classification of k-isohedral tilings in all three 2-dimensional spaces of constant curvature; extend this work to higher-dimensional spaces.

3.4 NONPERIODIC AND APERIODIC TILINGS

Nonperiodic tilings are found everywhere in nature, from cracked glazes to biological tissues to real crystals. In a remarkable number of cases, such tilings exhibit strong regularities. For example, many such tilings have simplicial Delaunay duals. Others repeat on increasingly larger scales. An even larger class of tilings are those now called repetitive, in which every bounded configuration appearing anywhere in the tiling is repeated infinitely many times throughout it. Aperiodic tilings—those whose prototile sets admit only nonperiodic tilings—are particularly interesting. They were first introduced to prove the Undecidability Theorem (Section 3.1). Later, after Penrose found pairs of aperiodic prototiles (see Figure 3.4.1), they became popular in recreational mathematical circles. Their deep mathematical

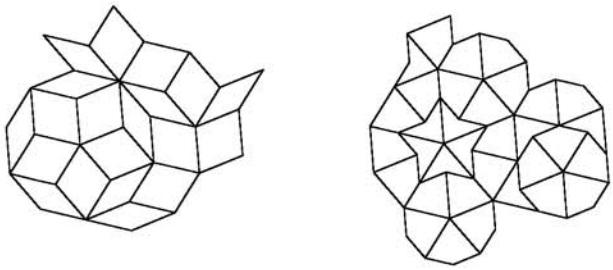


FIGURE 3.4.1
Portions of Penrose tilings of the plane (a) *by rhombs;* (b) *by kites and darts. Matching rules are not shown (see Chapter* 51*).*

properties were first studied by Penrose, Conway, de Bruijn, and others. After the discovery of "quasicrystals" in 1984, aperiodic tilings became the focus of intense research. The basic ideas of this rapidly-developing subject are only introduced here; they are discussed in more detail in Chapter 51.

GLOSSARY

Nonperiodic tiling: A tiling with no translation symmetry.

Hierarchical tiling: A tiling whose tiles can be composed into larger tiles, called ***level-one*** tiles, whose level-one tiles can be composed into level-two tiles, and so on *ad infinitum.* Usually one assumes that the prototile sets at each level are

the same up to a similarity or a rigid motion. In some cases it is necessary to partition the original tiles before composition.

Self-similar tiling: A hierarchical tiling for which the larger tiles are copies of the prototiles (all enlarged by a constant expansion factor λ). k-rep tiles are the special case when there is just one prototile (Figure 3.2.2).

Uniquely hierarchical tiling: A tiling whose j-level tiles can be composed into $(j+1)$-level tiles in only one way ($j = 0, 1, \ldots$).

Composition rule (for a hierarchical tiling): The equations $T_i' = m_{i1}T_1 \cup \ldots \cup m_{ik}T_k$, $i = 1, ..., k$, that describe the numbers m_{ij} of each prototile T_j in the next higher level prototile T_i'. These equations define a linear map whose matrix has i, j entry m_{ij}.

Relatively dense configuration: A configuration C of tiles in a tiling for which there exists a radius r_C such that every ball of radius r_C in the tiling contains a copy of C.

Repetitive: A tiling in which every bounded configuration of tiles is relatively dense in the tiling.

Local isomorphism class: A family of tilings such that every bounded configuration of tiles that appears in any of them appears in all of the others. (For example, the uncountably many Penrose tilings with the same prototile set form a single local isomorphism class.)

Projected tiling: A tiling obtained by the canonical projection method (see Chapter 51).

Aperiodic prototile set: A prototile set that admits only nonperiodic tilings (see Figure 3.4.1).

Aperiodic tiling: A tiling with an aperiodic prototile set.

Matching rules: A list of rules for fitting together the prototiles of a given prototile set.

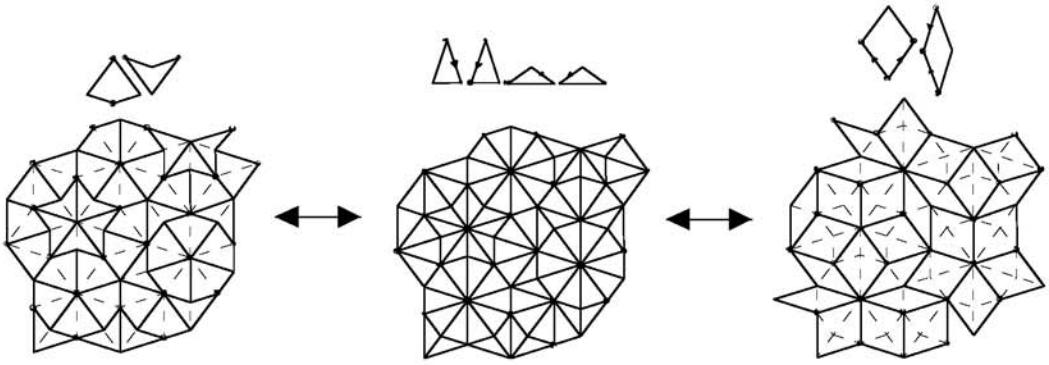
Mutually locally derivable tilings: Two tilings are mutually locally derivable if the tiles in either tiling can, through a process of decomposition into smaller tiles, or regrouping with adjacent tiles, or a combination of both processes, form the tiles of the other (see Figure 3.4.2).

Complex Perron number: An algebraic integer which is strictly larger in modulus than its Galois conjugates (except for its complex conjugate).

MAIN RESULTS

1. Self-similar and projected tilings are repetitive.
2. Uniquely hierarchical tilings are nonperiodic.
3. For each complex Perron number λ there is a self-similar tiling with expansion λ [Ken95].
4. "Irrational" projected tilings are nonperiodic (see Chapter 51).

FIGURE 3.4.2
The Penrose tilings by kites and darts and by rhombs are mutually locally derivable.



5. The prototile sets of certain irrational projected tilings can be equipped with matching rules so that all tilings admitted by the prototile set belong to a single local isomorphism class (see Chapter 51).

6. Mutual local derivability is an equivalence relation on the set of all tilings. The existence or nonexistence of hierarchical structure and matching rules is a class property.

7. Certain convex biprisms admit only nonperiodic monohedral tilings of $\mathbb{E}^3$ if no mirror-image copies of the tiles are allowed (see Figure 3.4.3). These tiles can be altered to produce nonconvex aperiodic prototiles for $\mathbb{E}^3$ [Dan95].

OPEN PROBLEMS

1. Can the prototile set of every uniquely hierarchical tiling be equipped with matching rules that force the hierarchical structure? (Matching rules exist for some well-known hierarchical tilings—the Penrose tilings and the chair tiling in $\mathbb{E}^2$, for instance.)

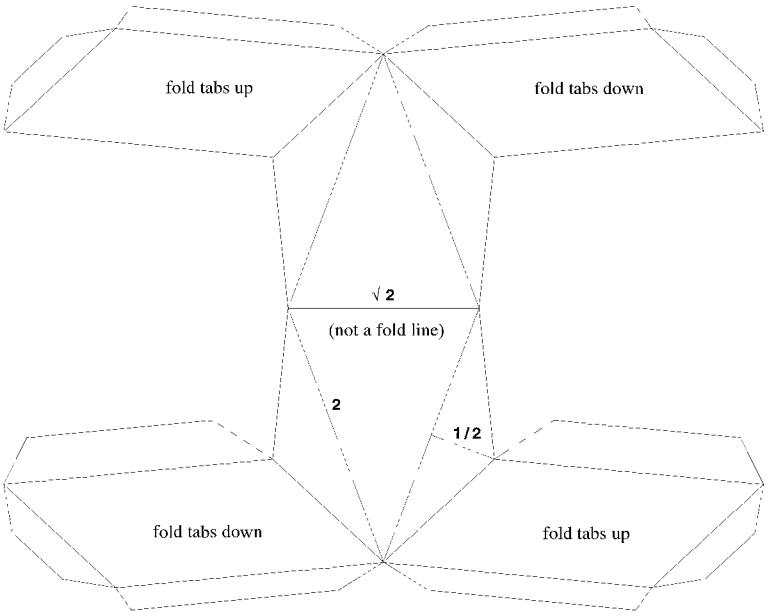
2. Does there exist a prototile in $\mathbb{E}^2$ that is aperiodic? Does there exist a convex prototile for $\mathbb{E}^3$ that is aperiodic without restriction?

3.5 OTHER TILINGS

There is a vast literature on tilings (or dissections) of bounded regions (such as rectangles and boxes, polygons, and polytopes) by tiles to satisfy particular conditions. This and much of the recreational literature focuses on tilings by tiles of a particular type, such as tilings by rectangles, tilings by clusters of n-cubes (polyominoes—see

FIGURE 3.4.3

Conway's biprism consists of two prisms fused at a common rhombus face. Small angle of rhombus is $\text{acos}(3/4) \approx 41.4°$*; diagonal of prism* ≈ 2.87*. When assembled, the vertices of the rhombus that is a common face of the two prisms are the poles of two 2-fold rotation axes.*



Chapter 12—and polycubes) or n-simplices (polyiamonds in $\mathbb{E}^2$), or tilings by recognizable animate figures. In the search for new ways to produce tiles and tilings, both mathematicians (such as P.A. MacMahon) and amateurs (such as M.C. Escher) have contributed to the subject. Recently the search for new shapes that tile S has produced knotted tiles, toroidal tiles, and twisted tiles. Kuperberg and Adams have shown that for any given knot K, there is a monohedral tiling of $\mathbb{E}^3$ (or of hyperbolic 3-space, or of spherical 3-space) whose prototile is a solid torus that is knotted as K. Also, Adams has shown that, given any polyhedral submanifold M with one boundary component in $\mathbb{E}^n$, a monohedral tiling of $\mathbb{E}^n$ can be constructed whose prototile has the same topological type as M.

Other directions of research seek to broaden the definition of prototile set: in new contexts, the tiles in a tiling may be homothetic (rather than congruent) images of tiles in a prototile set, or be topological images of tiles in a prototile set. For example, a tiling of $\mathbb{E}^n$ by polytopes in which every tile is combinatorially isomorphic to a fixed convex n-polytope (the combinatorial prototile) is said to be ***monotypic***. It has been shown that in $\mathbb{E}^2$, there exist monotypic face-to-face tilings by convex n-gons for all $n \geq 3$; in $\mathbb{E}^3$, every convex 3-polytope is the combinatorial prototile of a monotypic tiling. Many (but not all) classes of convex 3-polytopes admit monotypic face-to-face tilings.

3.6 SOURCES AND RELATED MATERIALS

SURVEYS

All results not given an explicit reference above may be traced in the following surveys.

[GS87]: The definitive, comprehensive treatise on tilings of $\mathbb{E}^2$, state-of-the-art as of the mid-1980's. All subsequent work (in any dimension) has taken this as its starting point for terminology, notation, and basic results. The Main Results of our Section 3.1 can be found here.

[Joh96]: A comprehensive and detailed account of uniform polytopes and honeycombs in Euclidean and non-Euclidean spaces of n dimensions.

[MP96]: The proceedings of the NATO Advanced Study Institute on the Mathematics of Aperiodic Order, held in Waterloo, Canada in August 1995. This book is state-of-the art for the theory of aperiodic tilings.

[Sch93]: A contemporary survey of tiling theory, especially useful for its accounts of monotypic and other kinds of tilings more general than those discussed in this chapter.

[Sen95]: Chapters 5–8 form an introduction to the emerging theory of aperiodic tilings.

[SS94]: This book is especially useful for its account of tilings in $\mathbb{E}^n$ by clusters of cubes.

RELATED CHAPTERS

Chapter 12: Polyominoes
Chapter 20: Voronoi diagrams and Delaunay triangulations
Chapter 51: Crystals and quasicrystals

REFERENCES

[Ada95] C. Adams. Tilings of space by knotted tiles. *Math. Intelligencer*, 17:41–51, 1995.

[BH96] L. Balke and D.H. Huson. Two-dimensional groups, orbifolds and tilings. *Geom. Dedicata*, 60:89–106, 1996.

[Ban91] C. Bandt. Self-similar sets 5. Integer matrices and fractal tilings of R^n. *Proc. Amer. Math. Soc.*, 112:549–562, 1991.

[BW94] H.-G. Bigalke and H. Wippermann. *Reguläre Parkettierungen*. B.I. Wissenschaftsverlag. Mannheim, 1994.

[Con92] J.H. Conway. The orbifold notation for surface groups. In M. Liebeck and J. Saxl, editors, *Groups, Combinatorics and Geometry*, Cambridge University Press, 1992.

[Cox54] H.S.M. Coxeter. Regular honeycombs in hyperbolic space. In *Proc. International Congress of Mathematicians*, volume III, Nordhoff, Groningen and North-Holland, Amsterdam, 1954.

[Dan95] L. Danzer. A family of 3D-spacefillers not permitting any periodic or quasiperiodic tilings. In *Proc. Aperiodic '94*. World Scientific, Singapore, 1995.

[DGS87] L. Danzer, B. Grünbaum, and G.C. Shephard. Equitransitive tilings, or how to discover new mathematics. *Math. Mag.*, 60:67–89, 1987.

[DS97] N. Dolbilin and D. Schattschneider. The local theorem for tilings. *Proc. Fields Institute*, to appear.

[Dre87] A.W.M. Dress. Presentations of discrete groups, acting on simply connected manifolds. *Adv. Math.*, 63:196–212, 1987.

[DHM93] A.W.M. Dress, D.H. Huson, and E. Molnár. The classification of face-transitive 3-D tilings. *Acta Cryst. Sect. A*, 49:806–817, 1993.

[FM84] A. Fontaine and G. Martin. Polymorphic polyominoes. *Math. Mag.*, 57:275–283, 1984.

[Grü94] B. Grünbaum. Uniform tilings of 3-space. *Geombinatorics*, 4:49–56, 1994.

[GS79] B. Grünbaum and G.C. Shephard. Incidence symbols and their applications. *Proc. Sympos. Pure Math.*, 34:199–244, Amer. Math. Soc., 1979.

[GS87] B. Grünbaum and G.C. Shephard. *Tilings and Patterns.* W. H. Freeman, New York, 1987.

[Hah83] T. Hahn, editor. *International Tables for Crystallography*, volume A. *Space Group Symmetry.* Dordrecht, Boston, published for the IUCr by Reidel Publishing Co., 1983.

[HK63] H. Heesch and O. Kienzle. *Flächenschluss. System der Formen lückenlos aneinanderschliessender Flachteile.* Springer, Berlin, 1963.

[Hus93] D.H. Huson. The generation and classification of tile-k-transitive tilings of the Euclidean plane, the sphere, and the hyperbolic plane. *Geom. Dedicata*, 47:269–296, 1993.

[Joh96] N. Johnson. *Uniform Polytopes.* Cambridge University Press, 1996.

[Ken95] R. Kenyon. The construction of self-similar tilings. Preprint, 1995.

[LS92] J. Lagarias and P. Shor. Keller's cube-tiling conjecture is false in high dimensions. *Bull. Amer. Math. Soc.*, 27:279–283, 1992.

[Mac21] P.A. MacMahon. *New Mathematical Pastimes.* Cambridge University Press, 1921.

[MP96] R. Moody and G. Patera. *The Mathematics of Aperiodic Order.* Kluwer (NATO series), 1996, to appear.

[Sch78] D. Schattschneider. The plane symmetry groups: their recognition and notation. *Amer. Math. Monthly*, 85:439–450, 1978.

[Sch88] P. Schmitt. An aperiodic prototile in space. Preprint, 1988.

[Sch90] D. Schattschneider. *Visions of Symmetry. Notebooks, Periodic Drawings, and Related Work of M.C. Escher.* W. H. Freeman, New York, 1990.

[Sch93] E. Schulte. Tilings. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, volume B, North Holland, Amsterdam, 1993, 899–932.

[Sen81] M. Senechal. Which tetrahedra fill space? *Math. Mag.*, 54:227–43, 1981.

[Sen88] M. Senechal. Color symmetry. *Comput. Math. Appl.*, 16:545–553, 1988.

[Sen91] M. Senechal. *Crystalline Symmetries. An Informal Mathematical Introduction.* Adam Hilger, Bristol, 1991.

[Sen95] M. Senechal. *Quasicrystals and Geometry.* Cambridge University Press, 1995.

[SS94] S. Stein and S. Szabó. *Algebra and Tiling.*, volume 25 of *The Carus Math. Monogr.* Math. Assoc. of Amer., Providence, 1994.

[Wie82] T.W. Wieting. *The Mathematical Theory of Chromatic Plane Ornaments.* Marcel Dekker, New York, 1982.

4 HELLY-TYPE THEOREMS AND GEOMETRIC TRANSVERSALS

Rephael Wenger

INTRODUCTION

A geometric transversal is an affine subspace of $\mathbb{R}^d$, such as a point, line, plane, or hyperplane, that intersects every member of a family of convex sets. Eduard Helly's celebrated theorem gives conditions for the members of a family of convex sets to have a point in common, i.e., a point transversal. In Section 4.1 we highlight some of the more notable theorems related to Helly's theorem and point transversals. Section 4.2 is devoted to geometric transversal theory.

4.1 HELLY-TYPE THEOREMS

In 1913, Eduard Helly proved the following theorem:

THEOREM 4.1.1 *Helly's Theorem*

Let $\mathcal{A}$ be a finite family of at least $d+1$ convex sets in $\mathbb{R}^d$. If every $d+1$ members of $\mathcal{A}$ have a point in common, then there is a point common to all members of $\mathcal{A}$.

The theorem also holds for infinite families of compact convex sets.

Helly's theorem spawned numerous generalizations and variants. These theorems usually have the form: If every m members of a family of objects have property $\mathcal{P}$ then the entire family has property $\mathcal{Q}$. When $\mathcal{P}$ equals $\mathcal{Q}$, theorems of this form are sometimes referred to as ***Helly-type*** theorems. In Helly's theorem the objects are convex sets in $\mathbb{R}^d$, properties $\mathcal{P}$ and $\mathcal{Q}$ are the properties of having a point in common, and m equals $d+1$. Most generalizations of Helly's theorem take four forms: replacing convex sets by other objects in $\mathbb{R}^d$, strengthening properties $\mathcal{P}$ and $\mathcal{Q}$, replacing $m = d+1$ by some other number or condition, and replacing $\mathbb{R}^d$ by the d-dimensional sphere, $\mathbb{S}^d$.

The first five parts of this section discuss various generalizations of Helly's theorem. The sixth and seventh part discuss some theorems and algorithms related to Helly's theorem. The last part contains some open problems. The theorems will all be stated for finite families of convex sets. As with Helly's theorem, many of them extend to infinite families of compact convex sets by standard topological arguments.

GLOSSARY

Convex: A set $a \subseteq \mathbb{R}^d$ is convex if $x, y \in a$ implies that line segment $xy \subseteq a$.

Convex hull: The convex hull of a set of points $X \subseteq \mathbb{R}^d$ is the smallest (inclusionwise) convex set containing X.

Homology cell: Metric space a is a homology cell if it is nonempty and homologically trivial (acyclic) in all dimensions.

Translate: Set $a \subseteq \mathbb{R}^d$ is a translate of set $b \subseteq \mathbb{R}^d$ if $a = \{v + x \mid x \in b\}$ for some vector $v \in \mathbb{R}^d$.

Homothet: Set $a \subseteq \mathbb{R}^d$ is a (positive) homothet of set $b \subseteq \mathbb{R}^d$ if $a = \{v + tx \mid x \in b\}$ for some vector $v \in \mathbb{R}^d$ and scalar $t > 0$.

Flat: An affine subspace of dimension k.

Support: Hyperplane h supports convex set a if a intersects h and is contained in one of the closed halfspaces bounded by h. k-flat f supports convex set a if a intersects f and f is contained in some supporting hyperplane of a.

Diameter: The diameter of a point set a is the supremum of the distances between pairs of points in a.

Width: The width of a closed convex set a is the smallest distance between parallel supporting hyperplanes of a.

Piercing number: The piercing number of a family $\mathcal{A}$ of convex sets in $\mathbb{R}^d$ is the minimum number of points needed to intersect every member of $\mathcal{A}$.

NOTATION

$\mathrm{conv}(X)$: The convex hull of point set X.

$f_i(\mathcal{A})$: The number of subfamilies $\mathcal{A}'$ of size $i+1$ of a family $\mathcal{A}$ of point sets such that $\bigcap_{a \in \mathcal{A}'} a \neq \emptyset$.

$\mathcal{C}_j^d$: The family of all sets of $\mathbb{R}^d$ that are the unions of j or fewer convex sets.

$\mathcal{K}_j^d$: The family of all sets of $\mathbb{R}^d$ that are the unions of j or fewer pairwise disjoint closed convex sets.

4.1.1 GENERALIZATIONS TO NONCONVEX SETS

In 1930, Helly himself gave the following topological generalization of his theorem:

THEOREM 4.1.2

Let $\mathcal{A}$ be a finite family of closed homology cells in $\mathbb{R}^d$. If the intersection of every $d+1$ or fewer members of $\mathcal{A}$ is a homology cell, then the intersection of all the members of $\mathcal{A}$ is a homology cell.

Since the intersection of convex sets is a convex set and nonempty convex sets are homology cells, Theorem 4.1.2 implies Helly's theorem.

Helly's theorem can also be generalized to objects that are the unions of convex sets. Let $\mathcal{C}_j^d$ be the family of all sets of $\mathbb{R}^d$ that are the unions of j or fewer convex sets. The intersection of members of $\mathcal{C}_j^d$ is not necessarily in $\mathcal{C}_j^d$. Alon and Kalai [AK95] and independently Matoušek [Mat] proved:

THEOREM 4.1.3

For every $j, d \geq 1$ there exists an integer $c(j,d) < \infty$ such that: If $\mathcal{A}$ is a finite subfamily of $\mathcal{C}_j^d$ of size at least $c(j,d)$, such that the intersection of every subfamily of $\mathcal{A}$ is also in $\mathcal{C}_j^d$ and such that every $c(j,d)$ members of $\mathcal{A}$ have a point in common, then there is a point common to all the members of $\mathcal{A}$.

A tight version of Theorem 4.1.3 is known for objects that are the unions of pairwise disjoint closed convex sets. Let $\mathcal{K}_j^d$ be the family of all sets of $\mathbb{R}^d$ that are the unions of j or fewer pairwise disjoint closed convex sets.

THEOREM 4.1.4

Let $\mathcal{A}$ be a finite subfamily of $\mathcal{K}_j^d$ of size at least $j(d+1)$ such that the intersection of every j members of $\mathcal{A}$ is also in $\mathcal{K}_j^d$. If every $j(d+1)$ members of $\mathcal{A}$ have a point in common, then there is a point common to all the members of $\mathcal{A}$.

The value $j(d+1)$ cannot be reduced. A recent elegant proof of this theorem appears in [Ame96].

4.1.2 INTERSECTIONS IN MORE THAN A POINT

The following generalizations of Helly's theorem apply to families of convex sets but strengthen both the hypothesis and the conclusion of the theorem, usually by assuming that the sets intersect in more than a single point.

THEOREM 4.1.5

Let $\mathcal{A}$ be a finite family of convex sets in $\mathbb{R}^d$. If every $d-k+1$ or fewer members of $\mathcal{A}$ contain a k-flat in common, then there is a k-flat contained in all the members of $\mathcal{A}$.

THEOREM 4.1.6

Let $\mathcal{A}$ be a finite family of convex sets in $\mathbb{R}^d$. Let $\psi(0,d) = d+1$ and $\psi(k,d) = \max(d+1, 2(d-k+1))$ for $1 \leq k \leq d$. If the intersection of every $\psi(k,d)$ or fewer members of $\mathcal{A}$ has dimension at least k, then the intersection of all the members of $\mathcal{A}$ is a set of dimension at least k.

The values of $\psi(k,d)$ are tight and cannot be reduced.

THEOREM 4.1.7

Let $\mathcal{A}$ be a finite family of at least $d+1$ convex sets in $\mathbb{R}^d$ and let b be some convex set in $\mathbb{R}^d$. If every $d+1$ members of $\mathcal{A}$ contain [intersect;are contained in] some translate of b, then some translate of b is contained in [intersects;contains] all the members of $\mathcal{A}$.

THEOREM 4.1.8

Let $\mathcal{A}$ be a finite family of at least $d+1$ closed convex sets in $\mathbb{R}^d$. If the intersection of every $d+1$ members of $\mathcal{A}$ has width at least w, then the intersection of all the members of $\mathcal{A}$ has width at least w.

THEOREM 4.1.9

Let $\mathcal{A}$ be a finite family of at least $2d$ convex sets in $\mathbb{R}^d$. If the intersection of every $2d$ members of $\mathcal{A}$ has diameter at least 1, then the intersection of all the members of $\mathcal{A}$ has diameter at least $d^{-2d}/2$.

THEOREM 4.1.10

Let $\mathcal{A}$ be a finite family of at least $2d$ convex sets in $\mathbb{R}^d$. If the intersection of every $2d$ members of $\mathcal{A}$ has volume at least 1, then the intersection of all the members of $\mathcal{A}$ has volume at least d^{-2d^2}.

The value $2d$ in Theorems 4.1.9 and 4.1.10 is tight and cannot be reduced. The values $d^{-2d}/2$ and d^{-2d^2} are not tight and can be increased. Bárány, Katchalski, and Pach conjecture that the correct values are approximately $c_1 d^{-1/2}$and $d^{-c_2 d}$ for some c_1 and c_2.

4.1.3 REDUCING $d+1$

Reducing the number of intersecting convex sets in the hypothesis of Helly's theorem gives:

THEOREM 4.1.11

Let $\mathcal{A}$ be a finite family of convex sets in $\mathbb{R}^d$. For any $m \leq d+1$, if every m or fewer members of $\mathcal{A}$ have a point in common, then every $(d-m+1)$-flat in $\mathbb{R}^d$ has some translate that intersects every member of $\mathcal{A}$ and every $(d-m)$-flat in $\mathbb{R}^d$ is contained in a $(d-m+1)$-flat that intersects every member of $\mathcal{A}$.

It is also true that if every $(d-m+1)$-flat in $\mathbb{R}^d$ has some translate that intersects every member of $\mathcal{A}$ or every $(d-m)$-flat in $\mathbb{R}^d$ is contained in a $(d-m+1)$-flat that intersects every member of $\mathcal{A}$, then every m members of $\mathcal{A}$ have a point in common.

For a family $\mathcal{A}$ of n convex sets, let $f_i(\mathcal{A})$ be the number of subfamilies $\mathcal{A}'$ of $\mathcal{A}$ of size $i+1$ such that the $i+1$ members of $\mathcal{A}'$ have a point in common. ($f_i(\mathcal{A})$ is the number of faces of dimension i in the *nerve* of $\mathcal{A}$.) Helly's theorem states that if $f_d(\mathcal{A})$ equals $\binom{n}{d+1}$, then there is a point common to all the members of $\mathcal{A}$. What if $f_d(\mathcal{A})$ is some value less than $\binom{n}{d+1}$?

THEOREM 4.1.12

Let $\mathcal{A}$ be a finite family of $n \geq d+1$ convex sets in $\mathbb{R}^d$. For any r where $0 \leq r \leq n-d-1$, if $f_d(\mathcal{A}) > \binom{n}{d+1} - \binom{n-r}{d+1}$, then some $d+r+1$ members of $\mathcal{A}$ have a point in common.

THEOREM 4.1.13

Let $\mathcal{A}$ be a finite family of $n \geq d+1$ convex sets in $\mathbb{R}^d$. For any ρ where $0 \leq \rho \leq 1$, if $f_d(\mathcal{A}) > (1-(1-\rho)^{d+1})\binom{n}{d+1}$, then some $\lfloor \rho n \rfloor + 1$ members of $\mathcal{A}$ have a point in common.

The values given in Theorems 4.1.12 and 4.1.13 are tight and cannot be reduced. Tight versions of these theorems are also known when $f_d(\mathcal{A})$ is replaced by $f_i(\mathcal{A})$ for any $i > d$. Theorem 4.1.13 is sometimes called a fractional Helly theorem.

The hypothesis that every $d+1$ members of $\mathcal{A}$ have a point in common can also be replaced by the hypothesis that out of every p members of $\mathcal{A}$ some q have

a point in common, where $p \geq q \geq d+1$. For certain values of p and q, Hadwiger and Debrunner proved the following result on their so-called (p,q)-problem:

THEOREM 4.1.14

Let $\mathcal{A}$ be a finite family of at least p convex sets in $\mathbb{R}^d$. If out of every p members of $\mathcal{A}$ some q have a point in common, where $p \geq q \geq d+1$ and $p(d-1) < (q-1)d$, then some set of $p-q+1$ points intersects every member of $\mathcal{A}$.

The value of $p-q+1$ is tight and cannot be reduced.

For general values of p and q Alon and Kleitman [AK92] proved:

THEOREM 4.1.15

For every $p \geq q \geq d+1$, there exists a positive integer $c(p,q,d) < \infty$ such that: If $\mathcal{A}$ is a finite family of at least p convex sets in $\mathbb{R}^d$ and out of every p members of $\mathcal{A}$ some q have a point in common, then some set of $c(p,q,d)$ points intersects every member of $\mathcal{A}$.

For the special case of homothets, the intersection of every two members of $\mathcal{A}$ suffices.

THEOREM 4.1.16

For every d there exists a positive integer $c(d) < \infty$ such that: If $\mathcal{A}$ is a finite family of homothets of a convex set in $\mathbb{R}^d$ and every two members of $\mathcal{A}$ intersect, then some set of $c(d)$ points intersects every member of $\mathcal{A}$.

Tight bounds are known for circular disks in $\mathbb{R}^2$.

THEOREM 4.1.17

Let $\mathcal{A}$ be a finite family of circular disks in $\mathbb{R}^2$. If every two members of $\mathcal{A}$ intersect, then some set of four points intersects every member of $\mathcal{A}$.

THEOREM 4.1.18

Let $\mathcal{A}$ be a finite family of circular unit disks in $\mathbb{R}^2$. If every two members of $\mathcal{A}$ intersect, then some set of three points intersects every member of $\mathcal{A}$.

Danzer proved Theorem 4.1.17, settling a question by Gallai on the minimum number of points needed to intersect all the members of any family of pairwise intersecting circular disks in $\mathbb{R}^2$. Such problems are often called Gallai-type problems.

Theorem 4.1.13 generalizes to objects that are unions of convex sets. Let $\mathcal{C}_j^d$ be as above.

THEOREM 4.1.19

For every α, $0 \leq \alpha \leq 1$, and every $j, d > 0$, there exists a constant $c(j,\alpha,d) > 0$ such that: If $\mathcal{A}$ is a finite subfamily of $\mathcal{C}_j^d$ of size $n \geq d+1$ and $f_d(\mathcal{A}) > \alpha\binom{n}{d+1}$, then some $c(j,\alpha,d)n$ members of $\mathcal{A}$ have a point in common.

Similarly, Theorem 4.1.15 generalizes to subfamilies of $\mathcal{C}_j^d$ [AK95]:

THEOREM 4.1.20

For every $p \geq q \geq d+1$ and every $j > 0$, there exists a positive integer $c(j,p,q,d) < \infty$ such that: If $\mathcal{A}$ is a finite subfamily of $\mathcal{C}_j^d$ of size at least p and out of every p

members of $\mathcal{A}$ some q have a point in common, then some set of $c(j,p,q,d)$ points intersects every member of $\mathcal{A}$.

4.1.4 SPHERICAL HELLY-TYPE THEOREMS

Various generalizations of convexity to a convexity structure on the d-sphere, $\mathbb{S}^d$, give rise to various Helly-type theorems.

GLOSSARY

Robinson-convex: A set $a \subseteq \mathbb{S}^d$ is Robinson-convex if for every $x, y \in a$ where x and y are not antipodal points, the small arc of the great circle joining x and y is contained in a.

Strongly convex: A set $a \subseteq \mathbb{S}^d$ is strongly convex if a is Robinson-convex and does not contain any antipodal points.

Convex cone: A set $a \subseteq \mathbb{R}^d$ is a convex cone centered at the origin if $x, y \in a$ implies $t_x x + t_y y \in a$ for any scalars $t_x, t_y \geq 0$.

NOTATION

$-a$: The set of points antipodal to the points in $a \subseteq \mathbb{S}^d$.

$\dim(a)$: The dimension of a manifold a with boundary. (By convention, the dimension of the empty set is -1.)

RESULTS

THEOREM 4.1.21

Let $\mathcal{A}$ be a finite family of at least $d+2$ strongly convex sets in $\mathbb{S}^d$. If every $d+2$ members of $\mathcal{A}$ have a point in common, then there is a point common to all the members of $\mathcal{A}$.

THEOREM 4.1.22

Let $\mathcal{A}$ be a finite family of at least $2d+2$ Robinson-convex sets in $\mathbb{S}^d$. If every $2d+2$ members of $\mathcal{A}$ have a point in common, then there is a point common to all the members of $\mathcal{A}$.

Theorems 4.1.21 and 4.1.22 generalize to:

THEOREM 4.1.23

Let $\mathcal{A}$ be a finite family of Robinson-convex sets in $\mathbb{S}^d$. Let m equal $\min_{a \in \mathcal{A}}[\dim(a) + \dim(a \cap -a)]$. If every $m+3$ or fewer members of $\mathcal{A}$ have a point in common, then there is a point common to all the members of $\mathcal{A}$.

The values $d+2$, $2d+2$, and $m+3$ in Theorems 4.1.21, 4.1.22, and 4.1.23 can be reduced by one under certain suitable circumstances. A subset of $\mathbb{S}^d$ is Robinson-convex if and only if it is the intersection of $\mathbb{S}^d$ with some convex cone centered at the origin. Thus Theorems 4.1.22 and 4.1.23 can be formulated in terms of convex cones.

Weakening the hypothesis of Theorem 4.1.22 by replacing $2d+2$ by $d+1$ gives the following theorem:

THEOREM 4.1.24

Let $\mathcal{A}$ be a finite family of at least $d+n+1$ Robinson-convex sets in $\mathbb{S}^d$, $n>0$. If every $d+1$ members of $\mathcal{A}$ have a point in common, then some $d+\lfloor n/2 \rfloor + 1$ members of $\mathcal{A}$ have a point in common.

A spherical variant of the topological Helly theorem (Theorem 4.1.2) generalizes Theorem 4.1.21.

THEOREM 4.1.25

Let $\mathcal{A}$ be a finite family of closed homology cells in $\mathbb{S}^d$. If the intersection of every $d+2$ or fewer members of $\mathcal{A}$ is a homology cell, then the intersection of all the members of $\mathcal{A}$ is a homology cell.

4.1.5 OTHER GENERALIZATIONS

Helly's theorem generalizes to multiple families of convex sets:

THEOREM 4.1.26

Let $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{d+1}$ be nonempty finite families of convex sets in $\mathbb{R}^d$. If $\bigcap_{i=1}^{d+1} a_i \neq \emptyset$ for each choice of $a_i \in \mathcal{A}_i$, then $\bigcap_{a \in \mathcal{A}_i} a \neq \emptyset$ for some $\mathcal{A}_i$.

Setting $\mathcal{A}_1 = \mathcal{A}_2 = \cdots = \mathcal{A}_{d+1}$ gives Helly's original theorem.

Dol'nikov gave a variation of Theorem 4.1.11 for multiple families of convex sets:

THEOREM 4.1.27

Let $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{d-m+2}$ be $d-m+2$ finite families of convex sets in $\mathbb{R}^d$, $2 \leq m \leq d+1$. If every m or fewer members of each family $\mathcal{A}_i$ have a point in common, then there is some $(d-m+1)$-flat in $\mathbb{R}^d$ that intersects every member of $\mathcal{A} = \bigcup_{i=1}^{d-m+2} \mathcal{A}_i$.

Theorem 4.1.27 is a special case of a much more general theorem by Dol'nikov that gives conditions for an algebraic surface of dimension $d-m+1$ to intersect every member of $\mathcal{A} = \bigcup_{i=1}^{d-m+2} \mathcal{A}_i$.

4.1.6 RELATED THEOREMS

Helly's theorem implies and/or is implied by some notable theorems.

THEOREM 4.1.28 *Carathéodory's Theorem*

Each point of $\text{conv}(X)$, $X \subseteq \mathbb{R}^d$, *is a convex combination of $d+1$ or fewer points of X.*

THEOREM 4.1.29 *Radon's Theorem*

Each set of $d+2$ or more points in $\mathbb{R}^d$ can be partitioned into two disjoint sets whose convex hulls have a point in common.

THEOREM 4.1.30 *Kirchberger's Theorem*

For point sets $X, Y \subseteq \mathbb{R}^d$, $\text{conv}(X) \cap \text{conv}(Y) \neq \emptyset$ *if and only if* $\text{conv}(X') \cap \text{conv}(Y') \neq \emptyset$ *for some* $X' \subseteq X$ *and* $Y' \subseteq Y$ *where* $|X| + |Y| \leq d + 2$.

A theorem similar to Carathéodory's theorem gives conditions for a point to lie in the interior of the convex hull of a set of points.

THEOREM 4.1.31 *Steinitz's Theorem*

Each point in the interior of $\text{conv}(X)$, $X \subseteq \mathbb{R}^d$, *is in the interior of* $\text{conv}(X')$ *for some* $X' \subseteq X$ *and* $|X'| \leq 2d$.

Theorem 4.1.26 is a generalization of Helly's theorem to multiple families of convex sets. Carathéodory's theorem has a similar, related generalization:

THEOREM 4.1.32

Let $X_1, X_2, \ldots, X_{d+1}$ *be subsets of* $\mathbb{R}^d$. *If* $x \in \text{conv}(X_i)$ *for each* X_i, *then there exist points* $x_i \in X_i$ *such that* $x \in \text{conv}(\{x_1, \ldots, x_{d+1}\})$.

Finally, Radon's theorem has the following generalization:

THEOREM 4.1.33 *Tverberg's Theorem*

Each set of $(r-1)(d+1)+1$ *or more points in* $\mathbb{R}^d$ *can be partitioned into* r *subsets whose convex hulls have a point in common.*

The theorem is tight and the number $(r-1)(d+1)+1$ cannot be reduced. For more details, see Chapter 11.

4.1.7 RELATED ALGORITHMS

Helly's theorem provokes the following algorithmic problem: Given a family $\mathcal{A}$ of n convex sets, find a point common to all the sets or, if there is no such point, find $d+1$ members of $\mathcal{A}$ that have no point in common. When $\mathcal{A}$ is a family of n halfspaces, this problem is simply a specialized version of linear programming. Sharir and Welzl have generalized linear programming to a more abstract framework that they call *generalized linear programming.* The problem of finding a point common to n convex sets can be formulated and solved as a generalized linear programming problem. In addition, other Helly-type theorems have related algorithmic questions that can be formulated and solved as generalized linear programming problems [Ame94]. For more on linear programming and generalized linear programming, see Chapters 38 and 39.

4.1.8 OPEN PROBLEMS

PROBLEM 4.1.34

Prove or disprove that there exists some constant c *such that: If the intersection of every* $2d$ *members of a family* $\mathcal{A}$ *of at least* $2d$ *convex sets in* $\mathbb{R}^d$ *has diameter at least 1, then the intersection of all the members of* $\mathcal{A}$ *has diameter at least* $cd^{-1/2}$.

PROBLEM 4.1.35

Prove or disprove that there exists some constant c such that: If the intersection of every $2d$ members of a family $\mathcal{A}$ of at least $2d$ convex sets in $\mathbb{R}^d$ has volume at least 1, then the intersection of all the members of $\mathcal{A}$ has volume at least d^{-cd}.

PROBLEM 4.1.36

Let $\mathcal{A}$ be a finite family of translates of a convex set in $\mathbb{R}^2$. Prove or disprove that if every two members of $\mathcal{A}$ intersect, then some set of three points intersects every member of $\mathcal{A}$.

4.2 GEOMETRIC TRANSVERSALS

Much research on geometric transversals focuses on necessary and sufficient conditions for the existence of line, plane, or hyperplane transversals to a family $\mathcal{A}$ of convex sets. This research includes conditions on the existence of transversals to special families of convex sets, such as translates or homothets. Most of the results apply either to line transversals in $\mathbb{R}^2$ or to hyperplane transversals in $\mathbb{R}^d$. The "order" in which a transversal intersects $\mathcal{A}$ plays an important role in stating and proving such theorems. Given a family $\mathcal{A}$ of convex sets, in how many different orders can $\mathcal{A}$ be intersected by transversals?

The set of transversals to a family $\mathcal{A}$ of convex sets forms a topological space with the usual topology associated with affine subspaces in $\mathbb{R}^d$, i.e., the topology inherited from the Grassmannian. What is the combinatorial structure and complexity of this space? What are efficient algorithms for constructing this space? Under what conditions does a set of k-flats form the space of transversals to some family of convex sets?

GLOSSARY

Transversal: An affine subspace $f \subseteq \mathbb{R}^d$ of dimension k is a k-transversal to a family $\mathcal{A}$ of convex sets if f intersects every member of $\mathcal{A}$.

Line transversal: A 1-tranversal to a family of convex sets in $\mathbb{R}^d$.

Hyperplane transversal: A $(d-1)$-tranversal to a family of convex sets in $\mathbb{R}^d$.

Separated: A family $\mathcal{A}$ of convex sets is k-separated if no $k+2$ members of $\mathcal{A}$ have a k-transversal.

Ordering: A k-ordering of a family $\mathcal{A} = \{a_1, \ldots, a_n\}$ of convex sets is a family of orientations of $(k+1)$-tuples of $\mathcal{A}$ defined by a mapping $\chi : \mathcal{A}^{k+1} \to \{-1, 0, 1\}$ corresponding to the orientations of some family of points $X = \{x_1, \ldots, x_n\}$ in $\mathbb{R}^k$. The orientation of $(a_{i_0}, a_{i_1}, \ldots, a_{i_k})$ is the orientation of the corresponding points $(x_{i_0}, x_{i_1}, \ldots, x_{i_k})$, i.e.,

$$\left(\operatorname{sgn} \det \begin{pmatrix} 1 & x_{i_0}^1 & \cdots & x_{i_0}^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{i_k}^1 & \cdots & x_{i_k}^k \end{pmatrix} \right).$$

Nontrivial ordering: A k-ordering is nontrivial if at least one of its orientations is nonzero.

Acyclic oriented matroid: A rank r acyclic oriented matroid on a set $\mathcal{A}$ is a family of orientations of r-tuples of $\mathcal{A}$ defined by a mapping $\chi : \mathcal{A}^r \to \{-1, 0, 1\}$ satisfying certain "chirotope" axioms and a condition of "acyclicity"; for more details, see Chapter 6.

Realizable acyclic oriented matroid: An acyclic oriented matroid of rank r is *realizable* if it can be represented as the family of orientations of a set of points in $\mathbb{R}^{r-1}$.

Geometric permutation: A geometric permutation of a $(k-1)$-separated family $\mathcal{A}$ of convex sets in $\mathbb{R}^d$ is the pair of k-orderings induced by some k-transversal of $\mathcal{A}$.

Ackermann function: The extremely rapidly growing function defined recursively by $A(n) = A_n(n)$, where $A_1(n) = 2n$ and $A_k(n) = A_{k-1}^{(n)}(1)$, $k \geq 2$.

Davenport-Schinzel sequence: An (n, s) Davenport-Schinzel sequence is a sequence of integers, $(u_1, \ldots, u_m)$, where $1 \leq u_i \leq n$ and $u_i \neq u_{i+1}$, which does not contain any alternating subsequence $(u_{i_1}, u_{i_2}, \ldots, u_{i_{s+2}})$ of length $s+2$ such that $u_{i_1} = u_{i_3} = u_{i_5} = \cdots$ and $u_{i_2} = u_{i_4} = u_{i_6} = \cdots$ and $u_{i_1} \neq u_{i_2}$; for more details, see Section 40.4 of this Handbook.

Constant description complexity: A convex set has constant description complexity if it is defined by a constant number of algebraic equalities and inequalities of constant maximum degree.

Strictly convex: A compact convex set a is strictly convex if its boundary contains no line segments.

Stubby: Convex set a is ρ-stubby, $\rho \geq 1$, if it is contained in a ball of radius ρ and contains a ball of radius one.

NOTATION

$\mathcal{T}_k^d(\mathcal{A})$: The space of k-transversals to a family $\mathcal{A}$ of convex sets in $\mathbb{R}^d$.

$g_k^d(n)$: The maximum number of geometric permutations induced by k-transversals of $(k-1)$-separated families of n compact convex sets in $\mathbb{R}^d$.

$\alpha(n)$: The inverse of the Ackermann function.

$\lambda_s(n)$: The maximum length of an (n,s) Davenport-Schinzel sequence.

4.2.1 HADWIGER'S TRANSVERSAL THEOREM

In 1935, Vincensini asked if there is a Helly-type theorem for line transversals to a family $\mathcal{A}$ of convex sets in $\mathbb{R}^2$. In other words, is there a number m such that if every m members of $\mathcal{A}$ are simultaneously intersected by a line then there exists a single line intersecting all the members of $\mathcal{A}$? As Figure 4.2.1 illustrates when m equals four, there is no such m, even for line transversals to families of pairwise disjoint line segments.

However, in 1957 Hadwiger added a condition about the order in which every m members of $\mathcal{A}$ are intersected by a line to give the following theorem:

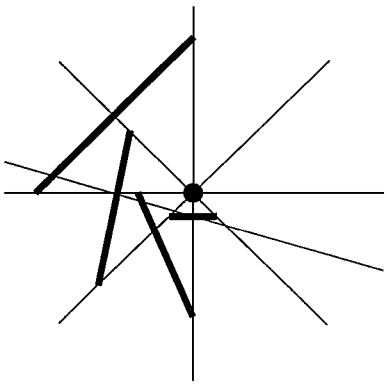


FIGURE 4.2.1
A counterexample to a Helly-type theorem for line transversals to families of convex sets in $\mathbb{R}^2$: Five convex sets, four line segments and a point, where every four sets have a line transversal but all five do not.

THEOREM 4.2.1 *Hadwiger's Transversal Theorem*

Let $\mathcal{A}$ be a finite family of pairwise disjoint convex sets in $\mathbb{R}^2$. If there exists a linear ordering of $\mathcal{A}$ such that every three members of $\mathcal{A}$ are intersected by a directed line in the given order, then $\mathcal{A}$ has a line transversal.

As with Helly's theorem, Hadwiger's transversal theorem and most of the similar theorems in this section also apply to infinite families of compact convex sets.

Hadwiger's transversal theorem generalizes to hyperplane transversals in $\mathbb{R}^d$ as follows:

THEOREM 4.2.2

Let $\mathcal{A}$ be a finite family of connected sets in $\mathbb{R}^d$. If, for some k, $0 \leq k < d$, there exists a nontrivial k-ordering of $\mathcal{A}$ such that every $k+2$ members of $\mathcal{A}$ are intersected by an oriented k-flat consistently with that k-ordering, then $\mathcal{A}$ has a hyperplane transversal.

An oriented k-flat f meets $\mathcal{A}' \subseteq \mathcal{A}$ consistently with a given k-ordering of $\mathcal{A}$ if one can choose a point y_i from the intersection of each set $a_i \in \mathcal{A}'$ and f such that the orientation of every $(k+1)$-tuple, $(y_{i_0}, y_{i_1}, \ldots, y_{i_k})$, of points in f matches the orientation of the corresponding $(k+1)$-tuple, $(a_{i_0}, a_{i_1}, \ldots, a_{i_k})$, of the k-ordering. Note that Theorem 4.2.2 eliminates the assumption of pairwise disjointness in Theorem 4.2.1.

Hadwiger's transversal theorem can be generalized even further in the language of oriented matroid theory [AW96]:

THEOREM 4.2.3

Let $\mathcal{A}$ be a finite family of connected sets in $\mathbb{R}^d$. If, for some k, $0 \leq k < d$, there exists an acyclic oriented matroid of rank $k+1$ on $\mathcal{A}$ such that every $k+2$ members of $\mathcal{A}$ are intersected by an oriented k-flat consistently with that oriented matroid, then $\mathcal{A}$ has a hyperplane transversal.

An oriented k-flat f meets $\mathcal{A}' \subseteq \mathcal{A}$ consistently with a given acyclic oriented matroid on $\mathcal{A}$ if one can choose a point y_i from the intersection of each set $a_i \in \mathcal{A}'$ and f such that the orientation of every $(k+1)$-tuple, $(y_{i_0}, y_{i_1}, \ldots, y_{i_k})$, of points in f matches the orientation of the corresponding $(k+1)$-tuple, $(a_{i_0}, a_{i_1}, \ldots, a_{i_k})$, of the oriented matroid. Theorem 4.2.2 is a restriction of Theorem 4.2.3 to realizable oriented matroids.

Hadwiger's theorem does not generalize to line transversals in $\mathbb{R}^3$. For each $m \geq 2$, there is a finite family $\mathcal{A}$ of pairwise disjoint convex sets in $\mathbb{R}^3$ and a linear ordering of $\mathcal{A}$ such that any $m-2$ members of $\mathcal{A}$ are met by a directed line in the given order, but $\mathcal{A}$ has no line transversal.

4.2.2 GALLAI-TYPE PROBLEMS

Under certain conditions a family $\mathcal{A}$ may not have a k-transversal but there may be some small set of k-flats whose union intersects every member of $\mathcal{A}$.

Alon and Kalai gave a variant of Theorem 4.1.15 for hyperplane transversals [AK95]:

THEOREM 4.2.4

For every $p \geq q \geq d+1$ there exists a positive integer $c(p,q,d) < \infty$ such that: If $\mathcal{A}$ is a finite family of at least p convex sets in $\mathbb{R}^d$ and out of every p members of $\mathcal{A}$ some q have a hyperplane transversal, then there are $c(p,q,d)$ hyperplanes whose union intersects every member of $\mathcal{A}$.

In $\mathbb{R}^2$ almost exact minimal values of $c(p,p,2)$ are known.

THEOREM 4.2.5

Let $\mathcal{A}$ be a finite family of convex sets in $\mathbb{R}^2$. If every four members of $\mathcal{A}$ have a line transversal, then there are two lines whose union intersects every member of $\mathcal{A}$.

THEOREM 4.2.6

Let $\mathcal{A}$ be a finite family of convex sets in $\mathbb{R}^2$. If every three members of $\mathcal{A}$ have a line transversal, then there are four lines whose union intersects every member of $\mathcal{A}$.

It is conjectured, but not proven, that the number four in the conclusion of Theorem 4.2.6 can be reduced to three. It cannot be reduced to two.

Theorem 4.2.4 generalizes to subfamilies of $\mathcal{C}_j^d$, i.e., families whose members are the unions of convex sets [AK95]:

THEOREM 4.2.7

For every $p \geq q \geq d+1$ and every j there exists a positive integer $c(j,p,q,d) < \infty$ such that: If $\mathcal{A}$ is a finite subfamily of $\mathcal{C}_j^d$ of size at least p and out of every p members of $\mathcal{A}$ some q have a hyperplane transversal, then there are $c(j,p,q,d)$ hyperplanes whose union intersects every member of $\mathcal{A}$.

4.2.3 TRANSLATES

Many special theorems apply to transversals of families of translates. Most noteworthy is the following Helly-type theorem conjectured by Grünbaum in 1958 and proved by Tverberg in 1989:

THEOREM 4.2.8

Let $\mathcal{A}$ be a family of at least five pairwise disjoint translates of a compact convex set in $\mathbb{R}^2$. If every five members of $\mathcal{A}$ have a line transversal, then $\mathcal{A}$ has a line transversal.

The number five cannot be reduced.

Under the weaker condition that every three members of $\mathcal{A}$ have a line transversal, the following theorem holds:

THEOREM 4.2.9

Let $\mathcal{A}$ be a family of pairwise disjoint translates of a compact convex set in $\mathbb{R}^2$. If every three members of $\mathcal{A}$ have a line transversal, then some subfamily $\mathcal{A}' \subseteq \mathcal{A}$ containing all but 108 members of $\mathcal{A}$ has a line transversal.

The number 108 is not known to be tight and can possibly be reduced. Katchalski and Lewis conjectured that the true value in the theorem should be two.

Versions of Theorems 4.2.8 and 4.2.9 exist for families of pairwise disjoint ρ-stubby convex sets where the constants are replaced by functions of ρ.

The condition that the members of $\mathcal{A}$ are pairwise disjoint can also be weakened [Rob].

THEOREM 4.2.10

For every $j > 0$ there exists a number $c(j)$ such that: If $\mathcal{A}$ is a family of at least $c(j)$ translates of a compact convex set in $\mathbb{R}^2$ such that the intersection of any j members of $\mathcal{A}$ is empty and such that every $c(j)$ members of $\mathcal{A}$ have a line transversal, then $\mathcal{A}$ has a line transversal.

Special Helly-type theorems are known for hyperplane transversals of families of translates of convex polytopes:

THEOREM 4.2.11

Let $\mathcal{A}$ be a family of translates of a convex polytope in $\mathbb{R}^d$ with n vertices. If every $\binom{n}{2}(d+1)$ or fewer members of $\mathcal{A}$ have a hyperplane transversal, then $\mathcal{A}$ has a hyperplane transversal.

THEOREM 4.2.12

Let $\mathcal{A}$ be a family of translates of a centrally symmetric convex polytope in $\mathbb{R}^d$ with n vertices. If every $\lfloor \frac{n}{2} \rfloor (d+1)$ or fewer members of $\mathcal{A}$ have a hyperplane transversal, then $\mathcal{A}$ has a hyperplane transversal.

The number $\lfloor \frac{n}{2} \rfloor (d+1)$ is tight and cannot be reduced.

4.2.4 GALLAI-TYPE PROBLEMS ON TRANSLATES

Eckhoff established Gallai-type results for line transversals of translates in $\mathbb{R}^2$:

THEOREM 4.2.13

Let $\mathcal{A}$ be a finite family of translates of a convex set in $\mathbb{R}^2$. If every three members of $\mathcal{A}$ have a line transversal, then there are two parallel lines whose union intersects every member of $\mathcal{A}$.

In higher dimensions, Eckhoff showed:

THEOREM 4.2.14

For every $k \geq 0$ there exists a number $c(k)$ such that: If $\mathcal{A}$ is a finite family of translates of a convex set in $\mathbb{R}^d$ and every $k+2$ members of $\mathcal{A}$ have a k-transversal, then there are $c(k)$ parallel k-flats whose union intersects every member of $\mathcal{A}$.

4.2.5 SPACE OF TRANSVERSALS

Given a family $\mathcal{A}$ of convex sets in $\mathbb{R}^d$, let $\mathcal{T}_k^d(\mathcal{A})$ be the space of all k-transversals of $\mathcal{A}$. If the members of $\mathcal{A}$ are closed, then the boundary of $\mathcal{T}_k^d(\mathcal{A})$ consists of k-flats that support one or more members of $\mathcal{A}$. This boundary can be partitioned into subspaces of k-flats that support the same subfamily of $\mathcal{A}$. Each of these subspaces can be further partitioned into connected components. The combinatorial complexity of $\mathcal{T}_k^d(\mathcal{A})$ is the number of such connected components.

Even in $\mathbb{R}^2$, the boundaries of two convex sets can intersect in an arbitrarily large number of points and have an arbitrarily large number of common supporting lines. Thus the space of line transversals to two convex sets in $\mathbb{R}^2$ can have arbitrarily large combinatorial complexity. However, if $\mathcal{A}$ consists of pairwise disjoint convex sets in $\mathbb{R}^2$ or, more generally, suitably separated convex sets in $\mathbb{R}^d$, then the complexity is bounded:

THEOREM 4.2.15

Let $\mathcal{A}$ be a $(d-2)$-separated family of n compact and strictly convex sets in $\mathbb{R}^d$. The combinatorial complexity of $\mathcal{T}_{d-1}^d(\mathcal{A})$ is $O(n^{d-1})$.

For certain types of convex sets, $\mathcal{T}_{d-1}^d(\mathcal{A})$ can be bounded without any assumptions about pairwise disjointness or separability.

THEOREM 4.2.16

Let $\mathcal{A}$ be a family of n $(d-1)$-balls in $\mathbb{R}^d$. The combinatorial complexity of $\mathcal{T}_{d-1}^d(\mathcal{A})$ is $O(n^{\lceil d/2 \rceil})$.

It is not known whether the bounds in Theorems 4.2.15 and 4.2.16 are asymptotically tight.

THEOREM 4.2.17

Let $\mathcal{A}$ be a family of convex polytopes in $\mathbb{R}^d$ with a total of n_f faces. The combinatorial complexity of $\mathcal{T}_{d-1}^d(\mathcal{A})$ is $O(n_f^{d-1}\alpha(n))$.

$\alpha(n)$ is the very slowly growing inverse of the Ackermann function.

THEOREM 4.2.18

Let $\mathcal{A}$ be a family of n line segments in $\mathbb{R}^d$. The combinatorial complexity of $\mathcal{T}_{d-1}^d(\mathcal{A})$ is $O(n^{d-1})$.

The asymptotic bounds in Theorems 4.2.17 and 4.2.18 are tight.

If $\mathcal{A}$ is a family of polytopes, the boundary of $\mathcal{T}_k^d(\mathcal{A})$ can also be partitioned into subspaces of k-flats that support the same polytope faces. Each of these subspaces can be further partitioned into connected components. The asymptotic

bounds in Theorems 4.2.17 and 4.2.18 also apply to the number of such connected components.

In $\mathbb{R}^2$ the pairwise disjointness condition of Theorem 4.2.15 can be relaxed to permit pairs of convex sets to have at most s common supporting lines. Let $\lambda_s(n)$ be the maximum length of an (n, s) Davenport-Schinzel sequence. Then $\lambda_s(n) = n\alpha(n)^{O(\alpha(n)^{s-3})}$.

THEOREM 4.2.19

Let $\mathcal{A}$ be a family of n compact connected sets in $\mathbb{R}^2$ such that any two members of $\mathcal{A}$ have at most s common supporting lines. The combinatorial complexity of $\mathcal{T}_1^2(\mathcal{A})$ is $O(\lambda_s(n))$.

In $\mathbb{R}^3$ bounds on $\mathcal{T}_2^3(\mathcal{A})$ can be given for families of sets that have algebraically simple descriptions [ASS95].

THEOREM 4.2.20

Let $\mathcal{A}$ be a family of n compact convex sets with constant description complexity in $\mathbb{R}^3$. The combinatorial complexity of $\mathcal{T}_2^3(\mathcal{A})$ is $O(n^{2+\epsilon})$ for any $\epsilon > 0$.

Finally, near tight asymptotic bounds are known for the complexity of the space of line transversals to convex polytopes in $\mathbb{R}^3$.

THEOREM 4.2.21

Let $\mathcal{A}$ be a family of convex polytopes in $\mathbb{R}^3$ with a total of n_f faces. The combinatorial complexity of $\mathcal{T}_1^3(\mathcal{A})$ is $O(n_f^{3+\epsilon})$ for any $\epsilon > 0$.

There are examples of families $\mathcal{A}$ of convex polytopes where the complexity of $\mathcal{T}_1^3(\mathcal{A})$ is $\Omega(n_f^3)$.

4.2.6 GEOMETRIC PERMUTATIONS

A directed line intersects a family $\mathcal{A}$ of pairwise disjoint convex sets in a well-defined order. Thus an undirected line transversal to $\mathcal{A}$ induces a pair of linear orderings or "permutations" on $\mathcal{A}$ consisting of the two orders in which oriented versions of the line intersect $\mathcal{A}$. Similarly an oriented k-transversal f intersects a $(k-1)$-separated family $\mathcal{A} = \{a_1, \ldots, a_n\}$ of convex sets in a well-defined k-ordering. The orientation of $(a_{i_0}, a_{i_1}, \ldots, a_{i_k})$ is the orientation in f of any corresponding set of points $(x_{i_0}, x_{i_1}, \ldots, x_{i_k})$, where $x_{i_j} \in a_{i_j} \cap f$. An unoriented k-transversal to a $(k-1)$-separated family $\mathcal{A}$ of convex sets induces a pair of k-orderings on $\mathcal{A}$, consisting of the two k-orderings in which oriented versions of the k-transversal intersect $\mathcal{A}$. Each such pair of k-orderings is called a *geometric permutation* of $\mathcal{A}$.

If $\mathcal{A}$ is $(k-1)$-separated, then two k-transversals that induce different geometric permutations on $\mathcal{A}$ must lie in different connected components of $\mathcal{T}_k^d(\mathcal{A})$. The converse also holds for hyperplane transversals.

THEOREM 4.2.22

Let $\mathcal{A}$ be a $(d-2)$-separated family of compact convex sets in $\mathbb{R}^d$. Two hyperplane transversals induce the same geometric permutation on $\mathcal{A}$ if and only if they lie in the same connected component of $\mathcal{T}_{d-1}^d(\mathcal{A})$.

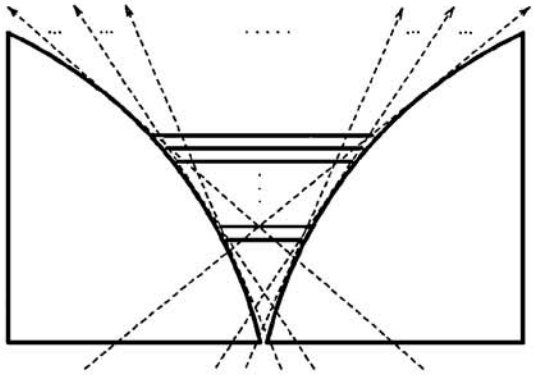


FIGURE 4.2.2
An example of n convex sets, two quarter circles and $n-2$ line segments, that have $2n-2$ geometric permutations. (From [GPW93], *with permission.)*

Consider geometric permutations induced by k-transversals of $(k-1)$-separated families of compact convex sets in $\mathbb{R}^d$. Let $g_k^d(n)$ be the maximum number of such geometric permutations over all such families $\mathcal{A}$ of size n. The following is known about $g_k^d(n)$:

THEOREM 4.2.23

1. $g_1^2(n) = 2n-2$. (See Figure 4.2.2.)
2. $g_1^d(n) = \Omega(n^{d-1})$.
3. $g_{d-1}^d(n) = O(n^{d-1})$.
4. $g_k^d(n) = O(k)^{d^2}\left(\binom{2^{k+1}-2}{k}\binom{n}{k+1}\right)^{k(d-k)}$ *(or* $g_k^d(n) = O(n^{k(k+1)(d-k)})$ *for fixed* k *and* d*)* [GPW96].

For families of pairwise disjoint translates, special bounds hold. Note that such families also have a special Helly-type transversal theorem (Theorem 4.2.8).

THEOREM 4.2.24

A family of pairwise disjoint translates of a compact convex set in $\mathbb{R}^2$ *has at most three geometric permutations.*

A family of pairwise disjoint ρ-stubby compact convex sets in $\mathbb{R}^2$ has at most c_ρ geometric permutations, where the constant c_ρ depends upon ρ.

4.2.7 TRANSVERSAL ALGORITHMS

Let $\mathcal{A}$ be a family of convex polytopes in $\mathbb{R}^d$ with a total of n_0 vertices. Table 4.2.1 gives known bounds on the worst case time to construct a representation of the space $\mathcal{T}_k^d(\mathcal{A})$ for various values of k and d.

The model of computation used in the lower bound for the time to construct $\mathcal{T}_1^2(\mathcal{A})$ is an algebraic decision tree. In the worst case, $\mathcal{T}_2^3(\mathcal{A})$ may have $\Omega(n_0^2\alpha(n_0))$ complexity, which gives the lower bound on constructing $\mathcal{T}_2^3(\mathcal{A})$. Similarly, $\mathcal{T}_1^3(\mathcal{A})$ may have $\Omega(n_0^3)$ complexity, giving an $\Omega(n_0^3)$ lower bound on the time to construct $\mathcal{T}_1^3(\mathcal{A})$.

Algorithms have also been proposed for constructing a representation of $\mathcal{T}_k^d(\mathcal{A})$ for families of convex sets with algebraically simple descriptions.

TABLE 4.2.1 Algorithms to construct $\mathcal{T}_k^d(\mathcal{A})$.

TRANSVERSAL SPACE	TIME COMPLEXITY
$\mathcal{T}_1^2(\mathcal{A})$	$\Theta(n_0 \log n_0)$
$\mathcal{T}_2^3(\mathcal{A})$	$\Theta(n_0^2 \alpha(n_0))$
$\mathcal{T}_{d-1}^d(\mathcal{A})$	$O(n_0^d), \quad d > 3$
$\mathcal{T}_1^3(\mathcal{A})$	$O(n_0^{3+\epsilon})$ for any $\epsilon > 0$

THEOREM 4.2.25

Let $\mathcal{A}$ be a family of n compact convex sets with constant description complexity in $\mathbb{R}^2$ such that any two members of $\mathcal{A}$ have at most s common supporting lines. A representation of $\mathcal{T}_1^2(n)$ can be constructed in $O(\lambda_s(n) \log n)$ time. In particular, if $\mathcal{A}$ is a family of convex translates or convex homothets with constant description complexity in $\mathbb{R}^2$, then $\mathcal{T}_1^2(n)$ can be constructed in $O(n \log n)$ time.

THEOREM 4.2.26

Let $\mathcal{A}$ be a family of n compact convex sets with constant description complexity in $\mathbb{R}^3$. A representation of $\mathcal{T}_2^3(\mathcal{A})$ can be constructed in $O(n^{2+\epsilon})$ time for any $\epsilon > 0$.

THEOREM 4.2.27

Let $\mathcal{A}$ be a $(d-2)$-separated family of n compact and strictly convex sets with constant description complexity in $\mathbb{R}^d$. A representation of $\mathcal{T}_{d-1}^d(\mathcal{A})$ can be constructed in $O(n^{d-1} \log^2 n)$ time.

THEOREM 4.2.28

Let $\mathcal{A}$ be a family of n d-balls in $\mathbb{R}^d$. A representation of $\mathcal{T}_{d-1}^d(\mathcal{A})$ can be constructed in $O(n^{\lfloor d/2 \rfloor + 1})$ time.

As noted in Section 4.1, algorithmic problems related to Helly-type theorems can be formulated and solved using generalized linear programming. In particular, Theorem 4.2.8 has the following algorithmic analogue:

THEOREM 4.2.29

Let $\mathcal{A}$ be a family of n pairwise disjoint translates of a compact convex set in $\mathbb{R}^2$. A line transversal for $\mathcal{A}$, if one exists, can be found in $O(n)$ time.

4.2.8 CONVEXITY ON THE AFFINE GRASSMANNIAN

Goodman and Pollack in [GP95] extend the notion of point set convexity to convexity of a set of k-flats in $\mathbb{R}^d$, giving several alternate and equivalent formulations of this convexity structure. In one such formulation, a set $\mathcal{F}$ of k-flats is *convex* if $\mathcal{F}$ is the transversal space of some family of convex point sets. They explore the conditions for $\mathcal{F}$ to be such a transversal space.

GLOSSARY

Convex (set of k-flats): A set $\mathcal{F}$ of k-flats is convex if $\mathcal{F}$ is the space of k-transversals to some (possibly infinite) family of convex sets in $\mathbb{R}^d$.

Surround: A set $\mathcal{F}$ of k-flats surrounds a k-flat f if there is some j-flat g containing f such that every $(j-1)$-flat containing f and lying in g strictly separates two members of $\mathcal{F}$ also lying in g.

Convex hull (of a set of k-flats): The convex hull of a set $\mathcal{F}$ of k-flats in $\mathbb{R}^d$ is the set of all k-flats surrounded by $\mathcal{F}$ in $\mathbb{R}^d$.

THEOREM 4.2.30

A set $\mathcal{F}$ of k-flats in $\mathbb{R}^d$ is the space of k-transversals to some (possibly infinite) family of convex point sets in $\mathbb{R}^d$ if and only if every k-flat surrounded by $\mathcal{F}$ is in $\mathcal{F}$.

There is no Helly-type theorem for convex sets of k-flats in $\mathbb{R}^d$ since such a theorem would be equivalent to a Helly-type theorem for k-transversals in $\mathbb{R}^d$. Such convex sets may have many connected components and may even have arbitrarily complex homology. Under suitable conditions in $\mathbb{R}^3$, however, each such connected component is itself convex [GPW95].

THEOREM 4.2.31

Let $\mathcal{F}$ be the space of all line transversals to a finite family of pairwise disjoint compact convex sets in $\mathbb{R}^3$. Each connected component of $\mathcal{F}$ can itself be represented as the space of line transversals to some finite family of pairwise disjoint compact convex sets in $\mathbb{R}^3$.

The theorem does not hold for line transversals to infinite families of noncompact convex sets.

4.2.9 OPEN PROBLEMS

PROBLEM 4.2.32

Let $\mathcal{A}$ be a finite family of convex sets in $\mathbb{R}^2$. Prove or disprove that if every three members of $\mathcal{A}$ have a line transversal, then there are three lines whose union intersects every member of $\mathcal{A}$.

PROBLEM 4.2.33

Let $\mathcal{A}$ be a family of pairwise disjoint translates of a compact convex set in $\mathbb{R}^2$. Prove or disprove that if every three members of $\mathcal{A}$ have a line transversal, then some subfamily $\mathcal{A}' \subseteq \mathcal{A}$ containing all but two members of $\mathcal{A}$ has a line transversal.

PROBLEM 4.2.34

Prove or disprove that there exists some constant m such that: If every m members of a family $\mathcal{A}$ of at least m pairwise disjoint unit balls in $\mathbb{R}^3$ have a line transversal, then $\mathcal{A}$ has a line transversal. Do the same for families of pairwise disjoint translates.

PROBLEM 4.2.35

Prove or disprove that there exist some m and r such that: If every m members of a finite family $\mathcal{A}$ of at least m convex sets in $\mathbb{R}^3$ have a line transversal, then there are r lines whose union intersects every member of $\mathcal{A}$. Prove a similar result under the conditions that out of every p members of $\mathcal{A}$ some q have a line transversal, for suitably large p and q. Generalize to k-transversals in $\mathbb{R}^d$.

PROBLEM 4.2.36

Let $\mathcal{F}$ be the space of all k-transversals to a finite $(k-1)$-separated family of compact convex sets in $\mathbb{R}^d$. Prove or disprove that each connected component of $\mathcal{F}$ can itself be represented as the space of k-transversals to some family of convex sets in $\mathbb{R}^d$.

4.3 SOURCES AND RELATED MATERIAL

SURVEYS

The following surveys contain references for any results not given an explicit citation.

[DGK63]: The classical survey of Helly's theorem and related results.

[Eck93]: A recent survey of Helly's theorem and related results, updating the material in [DGK63].

[GPW93]: A survey of geometric transversal theory.

RELATED CHAPTERS

Chapter 2: Packing and covering
Chapter 3: Tilings
Chapter 6: Oriented matroids
Chapter 11: Topological methods
Chapter 15: Face numbers of polytopes and complexes
Chapter 21: Arrangements
Chapter 38: Linear programming in low dimensions
Chapter 39: Mathematical programming
Chapter 40: Algorithmic motion planning

REFERENCES

[AK92] N. Alon and D. Kleitman. Piercing convex sets and the Hadwiger-Debrunner (p, q)-problem. *Adv. Math.*, 96:103–112, 1992.

[AK95] N. Alon and G. Kalai. Bounding the piercing number. *Discrete Comput. Geom.*, 13:245–256, 1995.

[Ame94] N. Amenta. Helly-type theorems and generalized linear programming. *Discrete Comput. Geom.*, 12:241–261, 1994.

[Ame96] N. Amenta. A short proof of an interesting Helly-type theorem. *Discrete Comput. Geom.*, 15:423–427, 1996.

[ASS95] P. Agarwal, O. Schwarzkopf, and M. Sharir. Overlay of lower envelopes and its applications. *Discrete Comput. Geom.*, 15:1–13, 1995.

[AW96] L. Anderson and R. Wenger. Oriented matroids and hyperplane transversals. *Adv. Math.*, 119:117–125, 1996.

[DGK63] L. Danzer, B. Grünbaum, and V. Klee. Helly's theorem and its relatives. In V. Klee, editor, *Convexity*, volume 7 of Proc. Sympos. Pure Math., pages 100–181. Amer. Math. Soc., Providence, 1963.

[Eck93] J. Eckhoff. Helly, Radon, and Carathéodory type theorems. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 389–448. North-Holland, Amsterdam, 1993.

[GP95] J.E. Goodman and R. Pollack. Foundations of a theory of convexity on affine Grassmann manifolds. *Mathematika*, 42:305–328, 1995.

[GPW93] J.E. Goodman, R. Pollack, and R. Wenger. Geometric transversal theory. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms Combin.*, pages 163–198. Springer-Verlag, Berlin, 1993.

[GPW95] J.E. Goodman, R. Pollack, and R. Wenger. On the connected components of the space of line transversals to a family of convex sets. In I. Bárány and J. Pach, editors, *The László Fejes Tóth Festschrift, Discrete Comput. Geom.*, 13:469–476, 1995.

[GPW96] J.E. Goodman, R. Pollack, and R. Wenger. Bounding the number of geometric permutations induced by k-transversals. *J. Combin. Theory Ser. A*, 75:187–197, 1996.

[Mat] J. Matoušek. A Helly-type theorem for unions of convex sets. *Discrete Comput. Geom.*, to appear.

[Rob] J.-M. Robert. Geometric orderings of intersecting translates and their applications. *Comput. Geom. Theory Appl.*, 7:59–72, 1997.

5 PSEUDOLINE ARRANGEMENTS

Jacob E. Goodman

INTRODUCTION

Pseudoline arrangements generalize in a natural way arrangements of straight lines, discarding the straightness aspect, but preserving their basic topological and combinatorial properties. Elementary and intuitive in nature, at the same time, by the Folkman-Lawrence topological representation theorem (see Chapter 6), they provide a concrete geometric model for oriented matroids of rank 3.

After their explicit description by Levi in the 1920's, and the subsequent development of the theory by Ringel in the 1950's, the major impetus was given in the 1970's by Grünbaum's monograph *Arrangements and Spreads*, in which a number of results were collected and a great many problems and conjectures posed about arrangements of both lines and pseudolines. The connection with oriented matroids discovered several years later led to further work. The theory is by now very well developed, with many combinatorial and topological results and relations to other areas, as well as an increasing number of applications in computational geometry.

Section 5.1 is devoted to the basic properties of pseudoline arrangements, and Section 5.2 to related structures, such as arrangements of straight lines, configurations (and generalized configurations) of points, and allowable sequences of permutations. (We do not discuss the connection with oriented matroids, however; that is included in Chapter 6.) In Section 5.3 we discuss the stretchability problem and in Section 5.4 summarize some of the (many) combinatorial results known about line and pseudoline arrangements. Section 5.5 deals with results of a topological nature, Section 5.6 with issues of combinatorial and computational complexity, and Section 5.7 with several applications, including sweeping arrangements and visibility graphs.

Unless otherwise noted, we work in the real projective plane $\mathbf{P}^2$.

5.1 BASIC PROPERTIES

GLOSSARY

Arrangement of lines: A labeled set of lines not all passing through the same point (the latter is called a ***pencil***).

Pseudoline: A simple closed curve whose removal does not disconnect $\mathbf{P}^2$.

Arrangement of pseudolines: A labeled set of pseudolines not a pencil, every pair meeting no more than once (hence exactly once and crossing).

Isomorphic arrangements: Two arrangements such that the mapping induced

by their labelings is an isomorphism of the cell complexes into which they partition $\mathbf{P}^2$. (Isomorphism classes of pseudoline arrangements correspond to reorientation classes of oriented matroids of rank 3; see Chapter 6.)

Stretchable: A pseudoline arrangement isomorphic to an arrangement of straight lines. Figure 5.1.1 illustrates what was once believed to be an arrangement of straight lines, but which was later proven not to be stretchable. We will see in Section 5.6 that most pseudoline arrangements, in fact, are not stretchable.

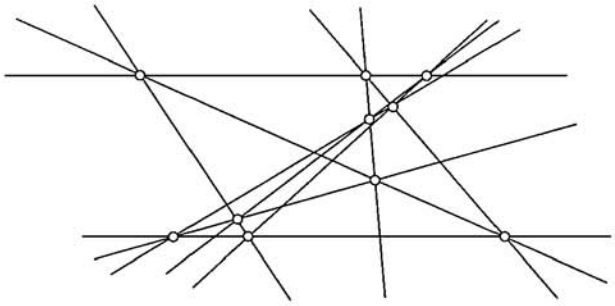


FIGURE 5.1.1
An arrangement of 10 *pseudolines, each containing* 3 *triple points; the arrangement is nonstretchable.*

Vertex: The intersection of two or more pseudolines in an arrangement.

Ordinary vertex: A vertex at which only two pseudolines meet.

Simple arrangement: An arrangement (of lines or pseudolines) in which each vertex is ordinary.

Wiring diagram: An (affine) arrangement of pseudolines consisting of piecewise linear "wires," each horizontal except for a short segment where it crosses another wire; see Figure 5.1.2, which shows a wiring diagram labeled $1, \ldots, n$ in upward order on the left and in downward order on the right.

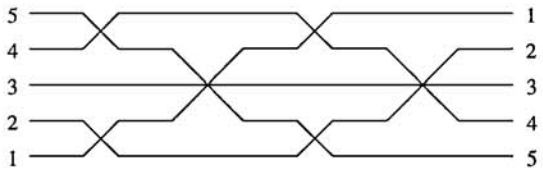


FIGURE 5.1.2
A wiring diagram.

p-convex hull: If $\mathcal{A}$ is an arrangement of pseudolines and p is a point not contained in any member of $\mathcal{A}$, $L \in \mathcal{A}$ is in the p-convex hull of $\mathcal{B} \subset \mathcal{A}$ if every path from p to a point of L meets some member of $\mathcal{B}$.

A fundamental tool in working with arrangements of pseudolines, which takes the place of the fact that two points determine a line, is the following.

THEOREM 5.1.1 *Levi Enlargement Lemma* [Lev26]

If $\mathcal{A} = \{L_1, \ldots, L_n\}$ is an arrangement of pseudolines and $p, q \in \mathbf{P}^2$ are two distinct points not on the same member of $\mathcal{A}$, there is a pseudoline L passing through p and q such that $\mathcal{A} \cup \{L\}$ is an arrangement.

Theorem 5.1.1 has been shown by Goodman and Pollack (see [BLS+93]) not

to extend to arrangements of pseudohyperplanes. It has, however, been extended by Snoeyink and Hershberger to the case of "2-intersecting curves" (where three points are given) [SH91], and shown by them not to extend to k-intersecting curves and $k+1$ points for $k > 2$.

The Levi Enlargement Lemma is used to prove extensions to pseudoline arrangements of a number of convexity results on arrangements of straight lines, duals of statements perhaps better known in the setting of configurations of points: Helly's theorem, Radon's theorem, Carathéodory's theorem, Kirchberger's theorem, the Hahn-Banach theorem, the Krein-Milman theorem, and Tverberg's generalization of Radon's theorem (cf. Chapter 4). We state two of these.

THEOREM 5.1.2 *Helly's theorem for pseudoline arrangements* [GP82]

If $\mathcal{A}_1, \ldots, \mathcal{A}_n$ are subsets of an arrangement $\mathcal{A}$ of pseudolines, and p is a point not on any pseudoline of $\mathcal{A}$ such that, for any i, j, k, $\mathcal{A}$ contains a pseudoline in the p-convex hull of each of $\mathcal{A}_i, \mathcal{A}_j, \mathcal{A}_k$, then there is an extension $\mathcal{A}'$ of $\mathcal{A}$ containing a pseudoline lying in the p-convex hull of each of $\mathcal{A}_1, \ldots, \mathcal{A}_n$.

THEOREM 5.1.3 *Tverberg's theorem for pseudoline arrangements* [Rou88]

If $\mathcal{A} = \{L_1, \ldots, L_n\}$ is a pseudoline arrangement with $n \geq 3m - 2$, and p is a point not on any member of $\mathcal{A}$, then $\mathcal{A}$ can be partitioned into subarrangements $\mathcal{A}_1, \ldots, \mathcal{A}_m$ and extended to an arrangement $\mathcal{A}'$ containing a pseudoline lying in the p-convex hull of $\mathcal{A}_i$ for every $i = 1, \ldots, m$.

Some of these convexity theorems, but not all, extend to higher dimensional arrangements; see [BLS+93], as well as Section 11.3 of this Handbook.

It is not difficult to see that the pseudolines in an arrangement may be drawn as polygonal lines, with bends only at vertices [Grü72]. Related to this is the following representation, which will be discussed further in Section 5.3.

THEOREM 5.1.4 [Goo80]

Every arrangement of pseudolines is isomorphic to a wiring diagram.

Theorem 5.1.4 is used in proving the following duality theorem, which extends to the setting of pseudolines the fundamental duality theorem between lines and points in the projective plane.

THEOREM 5.1.5 [Goo80]

If $\mathcal{A}$ is a pseudoline arrangement and $\mathcal{S}$ a point set in $\mathbf{P}^2$, and if I is the set of all true statements of the form "$p \in \mathcal{S}$ is incident to $L \in \mathcal{A}$," then there is a pseudoline arrangement $\hat{\mathcal{S}}$ and a point set $\hat{\mathcal{A}}$ such that the set of all incidences holding between members of $\hat{\mathcal{A}}$ and members of $\hat{\mathcal{S}}$ is precisely the dual $\hat{I}$ of I.

5.2 RELATED STRUCTURES

GLOSSARY

Circular sequence of permutations: A doubly infinite sequence of permuta-

tions of $1, \ldots, n$ associated with an arrangement $\mathcal{A}$ of lines $L_1, \ldots, L_n$ by sweeping a directed line across $\mathcal{A}$; see Figure 5.2.3 and the corresponding sequence below.

Local equivalence: Two circular sequences of permutations are locally equivalent if, for each index i, the order in which it switches with the remaining indices is either the same or opposite in the two sequences; see Figure 5.2.4 and Theorem 5.2.2 below.

Configuration of points: A (labeled) family $\mathcal{S} = \{p_1, \ldots, p_n\}$ of points, not all collinear, in $\mathbf{P}^2$.

Order type of a configuration $\mathcal{S}$: The mapping that assigns to each ordered triple i, j, k in $\{1, \ldots, n\}$ the orientation of the triple (p_i, p_j, p_k).

Combinatorial equivalence: Configurations $\mathcal{S}$ and $\mathcal{S}'$ are combinatorially equivalent if the set of permutations of $1, \ldots, n$ obtained by projecting $\mathcal{S}$ onto every line in general position agrees with the corresponding set for $\mathcal{S}'$.

Generalized configuration: A finite set of points in $\mathbf{P}^2$, together with a pseudoline joining each pair, the pseudolines forming an arrangement. (Several connecting pseudolines may coincide.) This is sometimes called a ***pseudoconfiguration***. An example is shown in Figure 5.2.1.

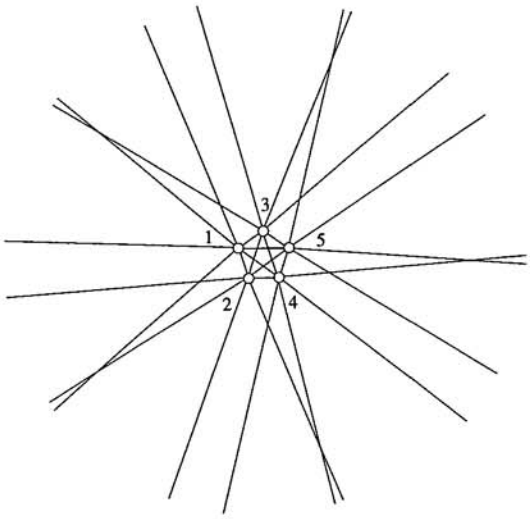


FIGURE 5.2.1
A generalized configuration of 5 *points.*

Allowable sequence of permutations: A doubly infinite sequence of permutations of $1, \ldots, n$ satisfying the three conditions of Theorem 5.2.1. It follows from those conditions that the sequence is periodic of length $\leq n(n-1)$, and that its period has length $n(n-1)$ if and only if the sequence is ***simple***, i.e., each move consists of the switch of a single pair of indices.

ARRANGEMENTS OF STRAIGHT LINES

Much of the work on pseudoline arrangements has been motivated by problems

involving straight-line arrangements. In some cases the question has been whether known results in the case of lines really depended on the *straightness* of the lines; for many (but not all) combinatorial results the answer has turned out to be negative. In other cases, generalization to pseudolines (or, equivalently, reformulations in terms of allowable sequences of permutations—see below) has permitted the solution of a more general problem where none was known previously in the straight case. Finally, pseudolines have turned out to be more useful than lines for certain algorithmic applications; this will be discussed in Section 5.7.

For arrangements of straight lines, there is a rich history of combinatorial results, some of which will be summarized in Section 5.4. Much of this is discussed in [Grü72].

Line arrangements are often classified by isomorphism type. For (unlabeled) arrangements of five lines, for example, Figure 5.2.2 illustrates the four possible isomorphism types, only one of which is simple.

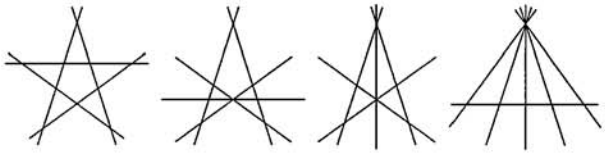


FIGURE 5.2.2
The 4 isomorphism types of arrangements of 5 lines.

There is a second classification of (numbered) line arrangements, which has proven quite useful for certain problems. If a distinguished point not on any line of the arrangement is chosen to play the part of the "vertical point at infinity," we can think of the arrangement $\mathcal{A}$ as an arrangement of nonvertical lines in the affine plane, and of P_∞ as the "upward direction." Rotating a directed line through P_∞ then amounts to sweeping a directed vertical line through $\mathcal{A}$ from left to right (say). We can then note the order in which this directed line cuts the lines of $\mathcal{A}$, and we arrive at a periodic sequence of permutations of $1, \ldots, n$, known as the ***circular sequence of permutations*** belonging to $\mathcal{A}$ (depending on the choice of P_∞ and the direction of rotation). This sequence is actually doubly infinite, since the rotation of the directed line through P_∞ can be continued in both directions. For the arrangement in Figure 5.2.3, for example, the circular sequence is

$$\mathcal{A}: \quad \ldots 12345 \overset{12,45}{\underline{\hphantom{xx}}} 21354 \overset{135}{\underline{\hphantom{xxx}}} 25314 \overset{25,14}{\underline{\hphantom{xx}}} 52341 \overset{234}{\underline{\hphantom{xxx}}} 54321 \ldots.$$

Notice how the "moves" between permutations are indicated.

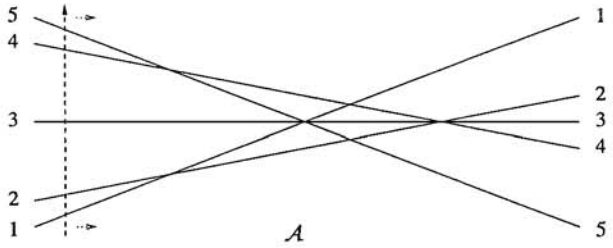


FIGURE 5.2.3
An arrangement of 5 lines.

THEOREM 5.2.1 [GP84]

A circular sequence of permutations arising from a line arrangement has the following properties:

(i) *The move from each permutation to the next consists of the reversal of one or more nonoverlapping adjacent substrings;*

(ii) *After a move in which i and j switch, they do not switch again until every other pair has switched;*

(iii) $1, \ldots, n$ *do not all switch simultaneously with each other.*

If two line arrangements are isomorphic, they may have different circular sequences, depending on the choice of P_∞ (and the direction of rotation). We do have, however,

THEOREM 5.2.2 [GP84]

If $\mathcal{A}$ and $\mathcal{A}'$ are arrangements of lines in $\mathbf{P}^2$, and Σ and Σ' are any circular sequences of permutations corresponding to $\mathcal{A}$ and $\mathcal{A}'$, then $\mathcal{A}$ and $\mathcal{A}'$ are isomorphic if and only if Σ and Σ' are locally equivalent.

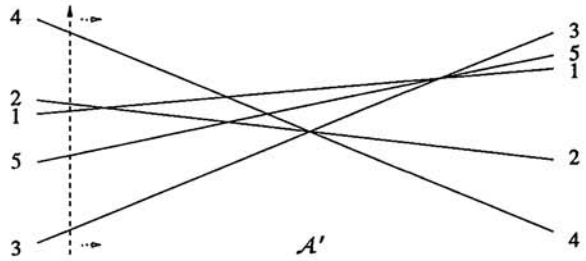


FIGURE 5.2.4
Another arrangement of 5 lines.

Theorem 5.2.2 is illustrated in Figure 5.2.4. Here, the circular sequence of the arrangement $\mathcal{A}'$, which (as an arrangement in $\mathbf{P}^2$) is isomorphic to arrangement $\mathcal{A}$ of Figure 5.2.3, is

$$\mathcal{A}': \quad \ldots 35124 \overset{12}{=} 35214 \overset{52,14}{=\!=} 32541 \overset{54}{=} 32451 \overset{324}{=} 42351 \overset{351}{=} 42153 \ldots$$

Reading off the local sequences of unordered switches of each, we get:

	1:	2:	3:	4:	5:
$\mathcal{A}$:	$\ldots;2;3,5;4;\ldots$	$\ldots;1;5;3,4;\ldots$	$\ldots;1,5;2,4;\ldots$	$\ldots;5;1;2,3;\ldots$	$\ldots;4;1,3;2;\ldots$
$\mathcal{A}'$:	$\ldots;2;4;3,5;\ldots$	$\ldots;1;5;3,4;\ldots$	$\ldots;2,4;1,5;\ldots$	$\ldots;1;5;2,3;\ldots$	$\ldots;2;4;1,3;\ldots$

We see that the 2-, 3-, and 5-sequences agree, while the 1- and 4-sequences are reversed.

CONFIGURATIONS OF POINTS

Under projective duality, arrangements of lines in $\mathbf{P}^2$ correspond to configurations of points. Some questions seem more natural in this setting of points, however, such as the Sylvester-Erdős problem about the existence of an ordinary line in a noncollinear configuration of points, and Scott's conjecture that the minimum number of directions determined by n noncollinear points is $2\lfloor n/2 \rfloor$.

Corresponding to the classification of line arrangements by isomorphism type, it turns out that the "dual" classification of point configurations is by order type.

THEOREM 5.2.3

If $\mathcal{A}$ and $\mathcal{A}'$ are arrangements of lines in $\mathbf{P}^2$ and $\mathcal{S}$ and $\mathcal{S}'$ the point sets dual to them, then $\mathcal{A}$ and $\mathcal{A}'$ are isomorphic if and only if $\mathcal{S}$ and $\mathcal{S}'$ have the same (or opposite) order types.

From a configuration of points one also derives a circular sequence of permutations in a natural way, by projecting the points onto a rotating line; this gives a finer classification than order type. The sequence for the arrangement in Figure 5.2.3 comes from the configuration in Figure 5.2.5 in this way.

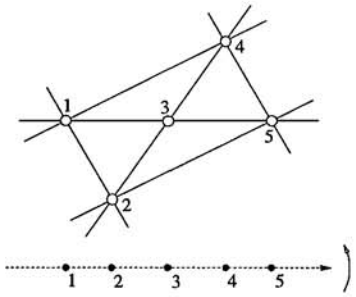


FIGURE 5.2.5
A configuration of 5 points.

In fact, it follows from projective duality that

THEOREM 5.2.4 [BLS+93]

A sequence of permutations is realizable by points if and only if it is realizable by lines.

The circular sequence of a point configuration can be reconstructed from the *set* of permutations obtained by projecting it onto all lines in general position.

THEOREM 5.2.5 [GP84]

Two configurations have the same circular sequences if and only if they are combinatorially equivalent.

This becomes useful in higher dimensions (where the circular sequence generalizes to a somewhat unwieldy cell decomposition of a sphere with a permutation associated with every cell), since it means that all one really needs to know is the *set* of permutations; how they fit together can then be determined.

See Chapter 1 of this Handbook for some recent results and some unsolved problems on point configurations.

GENERALIZED CONFIGURATIONS

Just as pseudoline arrangements generalize arrangements of straight lines, generalized configurations provide the corresponding generalization of configurations of points.

The two classifications described above for point configurations, by order type and by circular sequence of permutations, extend in a natural way to generalized configurations. For example, a circular sequence for the generalized configuration in Figure 5.2.1, which is determined by the cyclic order in which the connecting pseudolines meet a distinguished pseudoline (in this case the "pseudoline at infinity"), is

$$\ldots 12345^{\underline{34}}12435^{\underline{12}}21435^{\underline{14}}24135^{\underline{35}}24153^{\underline{15}}24513^{\underline{24}}42513^{\underline{25}}45213^{\underline{13}}45231^{\underline{23}}45321^{\underline{45}}54321\ldots$$

ALLOWABLE SEQUENCES

An allowable sequence of permutations is a combinatorial abstraction of the circular sequence of permutations associated with an arrangement of lines or a configuration of points. We can define, in a natural way, a number of geometric concepts for allowable sequences, such as *collinearity*, *betweenness*, *orientation*, *extreme point*, *convex hull*, *semispace*, *convex n-gon*, *parallel*, etc [GP80]. Not all allowable sequences are realizable, however, the smallest example being the sequence corresponding to Figure 5.2.1. A realization of this sequence would have to be a drawing of Figure 5.2.1 with straight lines, and it is not hard to prove that this is impossible.

More generally, we have

THEOREM 5.2.6 [GP80]

Suppose Σ is an allowable sequence with extreme points $1,\ldots n$ in counterclockwise order such that, for every i, side $i, i+1$ extended past vertex $i+1$ meets diagonal $i-1, i+2$ extended past vertex $i+2$ (the numbering is modulo n). Then Σ is not realizable by a configuration of points.

Allowable sequences provide a means of rephrasing many geometric problems about point configurations or line arrangements in combinatorial terms. For example, Scott's conjecture on the minimum number of directions determined by n lines has the simple statement: "Every allowable sequence of permutations of $1,\ldots,n$ has at least $2\lfloor n/2\rfloor$ moves in a half-period." It was proved in this more general form by Ungar [BLS+93], and the proof of the original Scott conjecture follows as a corollary; see also [Jam85].

The Erdős-Szekeres problem (see Chapter 1 of this Handbook) looks as follows in this more general combinatorial formulation:

PROBLEM 5.2.7 *Generalized Erdős-Szekeres Problem*

What is the minimum n such that for every simple allowable sequence Σ on $1,\ldots,n$, there are k indices with the property that each occurs before the other $k-1$ in some term of Σ?

Allowable sequences arise from pseudoline arrangements by way of wiring diagrams (see Theorem 5.1.4 above), from which they can be read off by sweeping a line across from left to right, just as with an arrangement of straight lines, and they arise as well from generalized configurations just as from configurations of points. In fact, the following theorem is just a restatement of Theorem 5.1.5.

THEOREM 5.2.8 [GP84]

Every allowable sequence of permutations can be realized both by an arrangement of pseudolines and by a generalized configuration of points.

WIRING DIAGRAMS

Wiring diagrams provide the simplest "geometric" realizations of allowable sequences. To realize the sequence

$$\mathcal{A}: \quad \ldots 12345 \overset{12,45}{\underline{\quad}} 21354 \overset{135}{\underline{\quad}} 25314 \overset{25,14}{\underline{\quad}} 52341 \overset{234}{\underline{\quad}} 54321 \ldots,$$

for example, simply start with horizontal "wires" labeled $1, \ldots, n$ in (say) increasing order from bottom to top, and, for each move in the sequence, let the corresponding wires cross. This gives the wiring diagram of Figure 5.1.2, and at the end the wires have all reversed order. (It is then easy to extend the curves in both directions to the "line at infinity," thereby arriving at a pseudoline arrangement in $\mathbf{P}^2$.)

We have the following isotopy theorem for wiring diagrams.

THEOREM 5.2.9 [GP85]

If two wiring diagrams numbered $1, \ldots, n$ in order are isomorphic as labeled pseudoline arrangements, then one can be deformed continuously to the other (or to its reflection) through wiring diagrams isomorphic as pseudoline arrangements.

HIGHER DIMENSIONS

Just as isomorphism classes of pseudoline arrangements correspond to oriented matroids of rank 3, the corresponding fact holds for higher-dimensional arrangements, known as arrangements of pseudohyperplanes: they correspond to oriented matroids of rank $d+1$ (see Theorem 6.2.4 in Chapter 6 of this Handbook).

It turns out, however, that in dimensions > 2, generalized configurations of points are (surprisingly) more restrictive than such oriented matroids; thus it is only in the plane that "projective duality" works fully in this generalized setting; see [BLS+93, Section 5.3].

5.3 STRETCHABILITY

STRETCHABLE AND NONSTRETCHABLE ARRANGEMENTS

Stretchability can be described in either combinatorial or topological terms:

THEOREM 5.3.1

Given an arrangement $\mathcal{A}$ or pseudolines in $\mathbf{P}^2$, the following are equivalent.

(i) *The cell decomposition induced by $\mathcal{A}$ is combinatorially isomorphic to that induced by some arrangement of sraight lines;*

(ii) *Some homeomorphism of $\mathbf{P}^2$ to itself maps every $L_i \in \mathcal{A}$ to a straight line.*

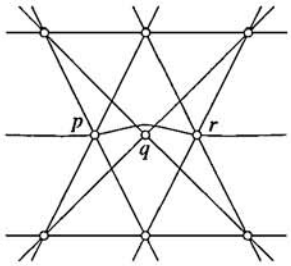


FIGURE 5.3.1
An arrangement that violates the theorem of Pappus.

Among the first examples observed of a nonstretchable arrangement of pseudolines was the non-Pappus arrangement of 9 pseudolines constructed by Levi: see Figure 5.3.1. Since Pappus's theorem says that points p, q, and r must be collinear if the pseudolines are straight, the arrangement in Figure 5.3.1 is clearly nonstretchable. A second example, involving 10 pseudolines, can be constructed similarly by violating Desargues's theorem.

Ringel showed how to convert the non-Pappus arrangement into a *simple* arrangement that was still nonstretchable. A symmetric drawing of it is shown in Figure 5.3.2.

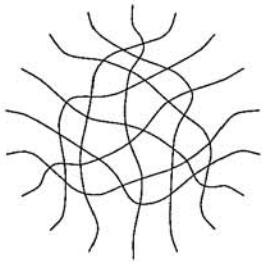


FIGURE 5.3.2
A simple nonstretchable arrangement of 9 pseudolines.

Using allowable sequences, Goodman and Pollack proved the conjecture of Grünbaum that the non-Pappus arrangement has the smallest size possible for a nonstretchable arrangement:

THEOREM 5.3.2 [BLS+93]

Every arrangement of 8 or fewer pseudolines is stretchable.

In addition, Richter-Gebert proved that the non-Pappus arrangement is unique among simple arrangements of the same size.

THEOREM 5.3.3 [BLS+93]

Every simple arrangement of 9 pseudolines is stretchable, with the exception of the simple non-Pappus arrangement.

The "bad pentagon," with extra points inserted to "pin down" the intersections of the sides and corresponding diagonals, provides another example of a nonstretchable arrangement; and Theorem 5.2.6, with extra points, provides, after dualizing, an infinite family of nonstretchable arrangements that were proved, by Bokowski and Sturmfels, to be "minor-minimal." This shows that stretchability of simple arrangements cannot be guaranteed by the exclusion of a finite number of "forbidden" subarrangements. A similar example was found by Haiman and Kahn. See [BLS+93, Section 8.3].

As for arrangements of more than 8 pseudolines, we have

THEOREM 5.3.4 [GPWZ94]

Let $\mathcal{A}$ be an arrangement of n pseudolines. If some face of $\mathcal{A}$ is bounded by at least $n-1$ pseudolines, then $\mathcal{A}$ is stretchable.

Finally, Shor shows in [Sho91] that even if a stretchable pseudoline arrangement has a symmetry, it may be impossible to realize this symmetry in any stretching.

THEOREM 5.3.5 [Sho91]

There exists a stretchable, simple pseudoline arrangement with a combinatorial symmetry such that no isomorphic arrangement of straight lines has the same combinatorial symmetry.

GENERALIZATIONS OF STRETCHABILITY

While not every pseudoline arrangement is isomorphic to an arrangement of straight lines, every pseudoline arrangement is ***d-stretchable***, i.e., realizable by an arrangement of graphs of polynomial functions of sufficiently high degree d. The following result gives the best bounds known on this degree.

THEOREM 5.3.6 *Goodman and Pollack* [BLS+93]

Let d_n be the smallest number d such that every simple arrangement of n pseudolines is d-stretchable. Then, for appropriate $c_1, c_2 > 0$, we have $c_1\sqrt{n} \leq d_n \leq c_2 n^2$.

In several recent papers, Pocchiola and Vegter explore another kind of realizability of pseudoline arrangements, by what they call arrangements of pseudotriangles. A ***pseudotriangle*** is a simply connected, bounded subset T of $\mathbb{R}^2$, bounded by 3 convex arcs pairwise tangent at their endpoints, such that T is contained in the triangle formed by these endpoints. The set T^* of directed tangent lines to the boundary of T can be identified by duality with a pseudoline in $\mathbf{P}^2$. Because two disjoint pseudotriangles share exactly one common tangent, if $\mathcal{T} = \{T_1, \ldots, T_n\}$ is an arrangement of pairwise disjoint pseudotriangles, the curves $T_1^*, \ldots, T_n^*$ form an arrangement of pseudolines which is "realized" by the arrangement $\mathcal{T}$. They prove

THEOREM 5.3.7 [PV94]

(i) *Every arrangement of straight lines is isomorphic to one realizable by an arrangement of disjoint pseudotriangles.*

(ii) *Every arrangement of pseudolines is isomorphic to one realizable by an arrangement of pseudotriangles.*

CONJECTURE 5.3.8 [PV94]

Every arrangement of pseudolines is isomorphic to one realizable by disjoint pseudotriangles.

5.4 COMBINATORIAL RESULTS

Although there are exceptions (see below), most combinatorial results known for line arrangements hold for pseudoline arrangements as well. We survey these in this section, including a number of results that update Grünbaum's comprehensive 1972 survey [Grü72]. For a discussion of *levels in arrangements* (dually, *k-sets*), see Chapters 21 and 1, respectively.

GLOSSARY

Simplicial arrangement: An arrangement of lines or pseudolines in which every cell is a triangle.

Near-pencil: An arrangement with all but one line (or pseudoline) concurrent.

Projectively unique: A line arrangement $\mathcal{A}$ with the property that every isomorphic line arrangement is the image of $\mathcal{A}$ under a projective transformation.

x-monotone path: In an arrangement of lines in $\mathbb{R}^2$, or in a wiring diagram, a path monotonic in the first coordinate, each step following a line (or wire) from one vertex to another.

SYLVESTER-TYPE RESULTS

CONJECTURE 5.4.1 [Grü69]

Every arrangement of n pseudolines has at least $\lfloor n/2 \rfloor$ ordinary vertices.

The strongest result to date on Conjecture 5.4.1 is the following theorem of Csima and Sawyer (cf. Chapter 1), which uses previous work of Hansen and improves a long-standing result of Kelly and Moser.

THEOREM 5.4.2 [CS93]

Every arrangement of n pseudolines, with the exception of the one shown in Figure 1.1.1(b), has at least $6n/13$ ordinary vertices.

The arrangement shown in Figure 1.1.1(a) shows that this result is sharp (see Chapter 1 of this Handbook for more details).

Using (complex) algebraic-geometric methods, Hirzebruch was able to prove the following result about the number t_i of vertices of multiplicity exactly i in an arrangement of *straight* lines.

THEOREM 5.4.3 [Hir83]

If an arrangement of n lines is not a near-pencil, then

$$t_2 + \frac{3}{4}t_3 \geq n + t_5 + 2t_6 + 3t_7 + \dots.$$

RELATIONS AMONG NUMBERS OF VERTICES, EDGES, AND FACES

THEOREM 5.4.4 *Euler*

If $f_i(\mathcal{A})$ is the number of faces of dimension i in the cell decomposition of $\mathbf{P}^2$ induced by an arrangement $\mathcal{A}$, then $f_0(\mathcal{A}) - f_1(\mathcal{A}) + f_2(\mathcal{A}) = 1$.

In addition to ***Euler's formula***, the following inequalities are satisfied for arbitrary pseudoline arrangements (here, $n(\mathcal{A})$ is the number of pseudolines in the arrangement $\mathcal{A}$).

THEOREM 5.4.5 [Grü72, SE88]

(i) $1+f_0(\mathcal{A}) \leq f_2(\mathcal{A}) \leq 2f_0(\mathcal{A})-2$, *with equality on the left for precisely the simple arrangements, and on the right for precisely the simplicial arrangements;*

(ii) $n(\mathcal{A}) \leq f_0(\mathcal{A}) \leq \binom{n(\mathcal{A})}{2}$, *with equality on the left for precisely the near-pencils, and on the right for precisely the simple arrangements;*

(iii) *For $n >> 0$, every f_0 satisfying $n^{3/2} \leq f_0 \leq \binom{n}{2}$, with the exceptions of $\binom{n}{2}-3$ and $\binom{n}{2}-1$, is the number of vertices of some arrangement of n pseudolines (in fact, of straight lines);*

(iv) $2n(\mathcal{A}) - 2 \leq f_2(\mathcal{A}) \leq \binom{n(\mathcal{A})}{2} + 1$, *with equality on the left for precisely the near-pencils, and on the right for precisely the simple arrangements;*

(v) $f_2(\mathcal{A}) \geq 3n(\mathcal{A}) - 6$ *if $\mathcal{A}$ is not a near-pencil.*

There are gaps in the possible values for $f_2(\mathcal{A})$, as shown by Theorem 5.4.6, which proves a conjecture posed by Grünbaum and generalized by Purdy, refining Theorem 5.4.5(iv).

THEOREM 5.4.6 *Martinov* [Mar93]

There exists an arrangement $\mathcal{A}$ of n pseudolines with $f_2(\mathcal{A}) = f$ if and only if, for some integer k with $1 \leq k \leq n-2$, we have $(n-k)(k+1) + \binom{k}{2} - \min(n-k, \binom{k}{2}) \leq f \leq (n-k)(k+1) + \binom{k}{2}$. Moreover, if $\mathcal{A}$ exists, it can be chosen to consist of straight lines.

THE NUMBER OF CELLS OF DIFFERENT SIZES

It is easy to see by induction that a *simple* arrangement of more than 3 pseudolines must have at least one nontriangular cell. This observation leads to many questions about numbers of cells of different types in both simple and nonsimple arrangements, some of which have not yet been answered satisfactorily.

The best result on the maximum number of triangles is the following.

THEOREM 5.4.7 [Grü72, Har85, Rou96]

The maximum number of triangles in an arrangement of n pseudolines is bounded above by $\lfloor n(n-1)/3 \rfloor$, with this bound achieved by infinitely many values of n.

PROBLEM 5.4.8 [Grü72]

Are there infinitely many arrangements of straight *lines with $n(n-1)/3$ triangles?*

PROBLEM 5.4.9 [Grü72]

What is the maximum number of k-sided cells in an arrangement of n pseudolines, for $k > 3$?

On the minimum number of triangles, we have

THEOREM 5.4.10 [Lev26]

In any arrangement of pseudolines, every pseudoline borders at least 3 triangles. Hence every arrangement of n pseudolines determines at least n triangles.

This minimum is achieved by the "cyclic arrangements" of lines generated by regular polygons, as in Figure 5.4.1.

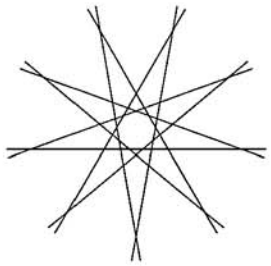


FIGURE 5.4.1
A cyclic arrangement of 9 lines.

The following result distinguishes line from pseudoline arrangements.

THEOREM 5.4.11 *Roudneff* [BLS+93]

An arrangement of n lines with only n triangles is simple. However, there exist nonsimple arrangements of n pseudolines with only n triangles.

An example of the second assertion of Theorem 5.4.11 is obtained by "collapsing" the central triangle in Figure 5.3.2.

A similar result for quadrilaterals is the following.

THEOREM 5.4.12 *Grünbaum* [Grü72], *Roudneff* [BLS+93]

(i) *Every arrangement of ≥ 5 pseudolines contains at most $n(n-3)/2$ quadrilaterals. For straight-line arrangements, this bound is achieved by a unique simple arrangement.*

(ii) *A pseudoline arrangement containing $n(n-3)/2$ quadrilaterals must be simple.*

There are infinitely many simple pseudoline arrangements with no quadrilaterals, contrary to what was once believed. The following result implies, in fact, that there must be many quadrilaterals or pentagons in *every* simple arrangement.

THEOREM 5.4.13 *Roudneff* [BLS+93]

Every pseudoline in a simple arrangement of $n > 3$ pseudolines borders at least 3 quadrilaterals or pentagons. Hence, if p_4 is the number of quadrilaterals and p_5 the number of pentagons, we must have $4p_4 + 5p_5 \geq 3n$.

The following result was proved after the opposite had been conjectured.

THEOREM 5.4.14 *Ljubić, Roudneff, and Sturmfels* [BLS+93]

There is a simple arrangement of straight lines containing no two adjacent triangles.

The proof involved finding a pseudoline arrangement with this property, then showing (algebraically, using Bokowski's "inequality reduction method"—see Section 5.6) that the arrangement, which consists of 12 pseudolines, is stretchable.

SIMPLICIAL ARRANGEMENTS

In addition to 90 "sporadic" examples of simplicial arrangements of straight lines, the following infinite families are known.

THEOREM 5.4.15 [Grü72]

Each of the following arrangements is simplicial:

(i) *the near-pencil of n lines;*
(ii) *the sides of a regular n-gon, together with its n axes of symmetry;*
(iii) *the arrangement in* (ii)*, together with the line at infinity, for n even.*

On the other hand, additional infinite families of (nonstretchable) simplicial arrangements of pseudolines are known, which are constructible from regular polygons by extending sides, diagonals, and axes of symmetry and modifying the resulting arrangement appropriately. For example, Figure 5.4.2 shows a member of such a family having 31 pseudolines, constructed from a decagon in this way.

One of the most important problems on arrangements is the following.

PROBLEM 5.4.16 [Grü72]

Classify all simplicial arrangements of pseudolines. Which of these are stretchable? In particular, are there any infinite families of simplicial line arrangements besides the three of Theorem 5.4.14*?*

It has apparently not been disproved that every (pseudo)line arrangement is a subarrangement of a simplicial (pseudo)line arrangement.

CONJECTURE 5.4.17 *Grünbaum* [Grü72]

Except for near-pencils, every simplicial arrangement of straight lines is projectively unique.

Finally, putting together results of Strommer [BLS+93] and of Csima and Sawyer [CS93], we get the following theorem; part (ii) is only a slight improvement over the corresponding result for nonsimplicial arrangements.

THEOREM 5.4.18

(i) *For every even n, there is a simplicial arrangement of n lines with a total of $(n^2 + 10n - 8)/8$ cells;*

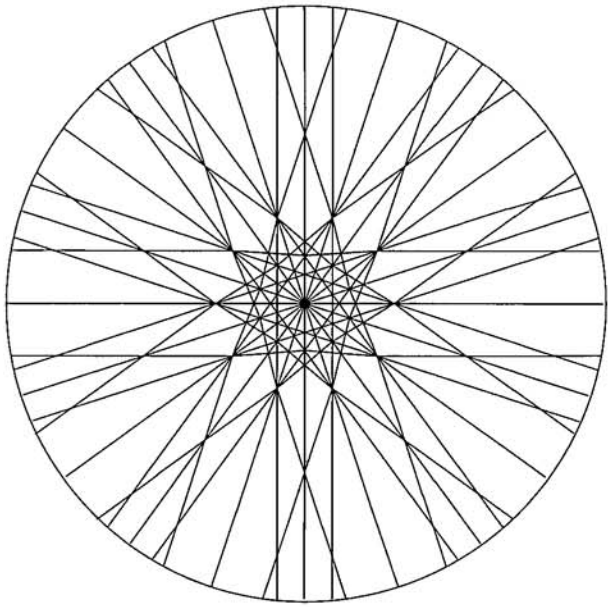


FIGURE 5.4.2
A simplicial arrangement of 31 *pseudolines. (The line at infinity, where "parallel" lines meet, is shown as a circle.)*

(ii) *Except for the arrangement of Figure* 1.1.1(b)*, the number of cells in a simplicial arrangement of n pseudolines is $\leq n(n-1)/3 + 4 - 4n/13$.*

PATHS IN PSEUDOLINE ARRANGEMENTS

The following result is most easily stated in terms of wiring diagrams.

THEOREM 5.4.19 [Mat91]

The maximum number of steps in an x-monotone path in a wiring diagram of size n is $\Omega(n^2/\log n)$, and in an arrangement of n lines is $\Omega(n^{5/3})$.

The only upper bound known for the lengths of such paths is the trivial one, $O(n^2)$, but Matoušek conjectures that the bound for lines is tight. (If so, this would distinguish line from pseudoline arrangements.)

COMPLEXITY OF SETS OF CELLS IN AN ARRANGEMENT

For cells that "line up" in an arrangement, the best result is

THEOREM 5.4.20 *Zone Theorem* [BEPY91]

The sum of the numbers of sides in all the cells of an arrangement of $n+1$ pseudolines that are supported by one of the pseudolines is $\leq 19n/20 - 1$.

For general sets of faces, on the other hand, Canham proved

THEOREM 5.4.21 *Canham* [Grü72]

If $F_1, \ldots, F_k$ are any k distinct faces of an arrangement of n pseudolines, then $\sum_{i=1}^{k} p(F_i) \leq n + 2k(k-1)$, where $p(F)$ is the number of sides of a face F. This is tight for $2k(k-1) \leq n$.

For $2k(k-1) > n$, this was improved by Clarkson et al. to the following result; the tightness follows from a result of Szemerédi and Trotter, proved independently by Edelsbrunner and Welzl.

THEOREM 5.4.22 [CEG+90]

The total number of sides in any k distinct cells of an arrangement of n pseudolines is $O(k^{2/3}n^{2/3} + n)$. This bound is (asymptotically) tight in the worst case.

There are a number of results of this kind for arrangements of objects in the plane and in higher dimensions; see Chapter 21, as well as [CEG+90].

5.5 TOPOLOGICAL PROPERTIES

GLOSSARY

Spread: Given the projective plane $\mathbf{P}^2$ with a distinguished line L_∞, a spread of pseudolines is a family $\mathcal{L} = \{L_x\}_{x \in L_\infty}$ of pseudolines varying continuously with $x = L_x \cap L_\infty$, any two of which meet at a single point (at finite distance).

Topological projective plane: $\mathbf{P}^2$, with a distinguished family $\mathcal{L}$ of pseudolines (its "lines"), is a topological projective plane if, for each $p, q \in \mathbf{P}^2$, exactly one $L_{p,q} \in \mathcal{L}$ passes through p and q, with $L_{p,q}$ varying continuously with p and q.

(There are other notions of both "spread" and "projective plane" [Grü72], but the ones defined here have the closest connection with pseudoline arrangements.)

Isomorphism of topological projective planes: A homeomorphism that maps "lines" to "lines."

Universal topological projective plane: One containing an isomorphic copy of every pseudoline arrangement.

Topological sweep: If $\mathcal{A}$ is a pseudoline arrangement in the affine plane and $L \in \mathcal{A}$, a topological sweep of $\mathcal{A}$ "starting at L" is a continuous family of pseudolines including L, each compatible with $\mathcal{A}$, which forms a partition of the plane.

Basic semialgebraic set: The set of solutions to a finite number of equations and strict inequalities in $\mathbb{R}^d$. (This term is sometimes used even if the inequalities are not necessarily strict.)

Stable equivalence: A relation on semialgebraic sets that preserves homotopy type. A precise definition appears in [Ric96].

EMBEDDING IN LARGER STRUCTURES

In [Grü72], Grünbaum asked a number of questions about extending pseudoline arrangements to more elaborate structures, in particular to spreads and topological planes. The strongest result known about such extendibility is the following, which extends results of Goodman, Pollack, Wenger, and Zamfirescu [GPWZ94].

THEOREM 5.5.1 [GPW96]

There exist uncountably many pairwise nonisomorphic universal topological projective planes.

In particular, this implies the following statements, together with the corresponding statements about spreads, all of which had been conjectured in [Grü72].

(i) Every pseudoline arrangement can be extended to a topological projective plane.

(ii) There exists a universal topological projective plane.

(iii) There are nonisomorphic topological projective planes such that every arrangement in each is isomorphic to some arrangement in the other.

Theorem 5.5.1 also implies the following result, established earlier by Snoeyink and Hershberger (and implicitly by Edmonds, Fukuda, and Mandel—see [BLS+93]).

THEOREM 5.5.2 *Sweeping Theorem* [SH91]

A pseudoline arrangement $\mathcal{A}$ in the affine plane can be swept by a pseudoline, starting at any $L \in \mathcal{A}$.

PROBLEM 5.5.3 [Grü72]

Which arrangements are present (up to isomorphism) in every *topological projective plane?*

MOVING FROM ONE ARRANGEMENT TO ANOTHER

In [Rin56], Ringel asked whether an arrangement $\mathcal{A}$ of straight lines could always be moved continuously to a given isomorphic arrangement $\mathcal{A}'$ (or to its reflection) so that all intermediate arrangements remained isomorphic. This question, which became known as the "isotopy problem" for arrangements, was eventually solved by Mnëv in [Mnë85], and (independently, since news of Mnëv's results had not yet reached the West) by White in the nonsimple case, then by Jaggi and Mani-Levitska in the simple case [BLS+93]. Mnëv's results are, however, by far stronger.

THEOREM 5.5.4 *Mnëv's Universality Theorem*

If V is any basic semialgebraic set defined over $\mathbb{Q}$, there is a configuration $\mathcal{S}$ of points in the plane such that the space of all configurations of the same order type as $\mathcal{S}$ is stably equivalent to V. If V is open in some $\mathbb{R}^n$, then there is a simple configuration $\mathcal{S}$ with this property.

From this it follows that the space of line arrangements isomorphic to a given one may have the homotopy type of *any* semialgebraic variety, and in particular may be disconnected, which gives a (very strongly) negative answer to the isotopy question. For a further generalization of Theorem 5.5.4, see [Ric96].

The line arrangement of smallest size known for which the isotopy conjecture fails consists of 14 lines in general position and was found by Suvorov [BLS+93].

Special cases where the isotopy conjecture *does* hold include:

(i) every arrangement of 9 or fewer lines in general position [Ric89], and

(ii) an arrangement of n lines containing a cell bounded by at least $n-1$ of them.

There are also results of a more combinatorial nature about the possibility of transforming one pseudoline arrangement to another. In [Rin56, Rin57], Ringel proved

THEOREM 5.5.5 *Ringel's Homotopy Theorem*

If $\mathcal{A}$ and $\mathcal{A}'$ are simple arrangements of pseudolines, then $\mathcal{A}$ can be transformed to $\mathcal{A}'$ by a finite sequence of steps each consisting of moving one pseudoline continuously across the intersection of two others. If $\mathcal{A}$ and $\mathcal{A}'$ are simple arrangements of lines, this can be done within the space of line arrangements.

The second part of Theorem 5.5.5 has been generalized by Roudneff and Sturmfels [BLS+93] to arrangements of planes; the first half is still open in higher dimensions.

Ringel also observed that the isotopy property does hold for pseudoline arrangements.

THEOREM 5.5.6 [Rin56]

If $\mathcal{A}$ and $\mathcal{A}'$ are isomorphic arrangements of pseudolines, then $\mathcal{A}$ can be deformed continuously to $\mathcal{A}'$ through isomorphic arrangements.

Ringel did not provide a proof of this observation, but one method of proving it is via Theorem 5.2.9, together with the following isotopy result.

THEOREM 5.5.7 *Goodman and Pollack* [BLS+93]

Every arrangement of pseudolines can be continuously deformed (through isomorphic arrangements) to a wiring diagram.

5.6 COMPLEXITY ISSUES

GLOSSARY

λ-matrix: The matrix with entries λ_{ij} = the number of points of the (generalized) configuration $\{p_1, \ldots, p_n\}$ to the left of the directed (pseudo)line $\overrightarrow{p_i p_j}$. (λ_{ii} is undefined.)

THE NUMBER OF ARRANGEMENTS

Various exact values, as well as bounds, are known for the number of equivalence classes of the structures discussed in this chapter. For low values of n, some of these are given in Table 5.6.1 [BKLR, BLS+93, Fel, GP80, Grü72, Knu92, Ric89].

The only exact formula known for arbitrary n follows from Stanley's formula:

TABLE 5.6.1 Exact numbers known for low n.

EQUIVALENCE CLASS	3	4	5	6	7	8	9	10	11	12	13	14	15
Isom classes of arr's of n lines	1	2	4	17									
" " " simple " " " "	1	1	1	4	11	135	4381	312114					
" " " simplicial " " " "	1	1	1	2	2	2	2	4	2	4	5	5	6
" " " arr's of n pseudolines	1	2	4	17									
" " " simple " " " "	1	1	1	4	11	135	4382	312356					
" " " simplicial " " " "	1	1	1	2	2	2							
Isom classes of simple gen'd config's	1	2	3	16	135	3315	158830						
Comb'l equiv classes of allow seq's	1	2	20										
" " " " realizable " "	1	2	19										
Simple allow seq's cont'ing $123\ldots n$	2	16	768	...	[see Theorem 5.6.1]								
Simple allow seq's	2	32	4608	...	[see Corollary 5.6.2]								

THEOREM 5.6.1 [BLS+93]

The number of simple allowable sequences on $1,\ldots,n$ *containing the permutation* $123\ldots n$ *is*

$$\frac{\binom{n}{2}!}{1^{n-1}3^{n-2}5^{n-3}\cdots(2n-3)^1}.$$

COROLLARY 5.6.2

The total number of simple allowable sequences on $1,\ldots,n$ *is*

$$\frac{(n-2)!\binom{n}{2}!}{1^{n-1}3^{n-2}5^{n-3}\cdots(2n-3)^1}.$$

For n arbitrary, Table 5.6.2 indicates the known asymptotic bounds [BLS+93, Fel, GP91, GP93, Knu92].

TABLE 5.6.2 Asymptotic bounds for large n (all logarithms are base 2).

EQUIVALENCE CLASS	LOWER BOUND	UPPER BOUND
Isom classes of (labeled) arr's of n pseudolines	$2^{n^2/6-5n/2}$	$2^{1.0850n^2}$
" " " " simple " " " "	"	$2^{.6974n^2}$
Order types of (labeled) n pt configs (simple or not)	$2^{4n\log n+\Omega(n)}$	$2^{4n\log n+O(n)}$
Isotopy classes of (labeled) n pt configs	"	"
Comb'l equiv classes of (labeled) n pt configs	$2^{7n\log n}$	$2^{8n\log n}$

CONJECTURE 5.6.3 [Knu92]

The number of isomorphism classes of simple pseudoline arrangements is $\leq 2^{\binom{n}{2}}$.

HOW MUCH SPACE IS NEEDED TO SPECIFY AN ARRANGEMENT?

A configuration or generalized configuration $\mathcal{S}$ is described, up to isomorphism, by the set of points lying to the left (say) of each line or pseudoline joining a pair of

points. The following theorem, which extends to higher dimensions, allows one to encode the order type of $\mathcal{S}$ in essentially one order of magnitude less space.

THEOREM 5.6.4 *Goodman and Pollack, Cordovil* [BLS+93]
If $\mathcal{S}$ is a configuration or generalized configuration in the plane, the order type of $\mathcal{S}$ is determined by its λ-matrix.

COROLLARY 5.6.5
The order type of an arrangement of pseudolines can be encoded in space $O(n^2 \log n)$.

A modification by Felsner of the λ-matrix encoding for planar arrangements improves this, giving an encoding of wiring diagrams in space $O(n^2)$:

THEOREM 5.6.6 [Fel]
Given a wiring diagram $\mathcal{A} = \{L_1, \ldots, L_n\}$, let $t_j^i = 1$ if the jth crossing along L_i is with L_k for $k > i$, 0 otherwise. Then the mapping that associates to each wiring diagram $\mathcal{A}$ the binary $n \times (n-1)$ matrix (t_j^i) is injective.

The number of stretchable pseudoline arrangements is much smaller than the total number, which suggests that it should be possible to encode these more compactly. The following result (stated here for the dual case of point configurations) shows, however, that the "naive" encoding, by coordinates of an integral representative, is doomed to be inefficient.

THEOREM 5.6.7 *Goodman, Pollack, and Sturmfels* [BLS+93]
For each configuration $\mathcal{S}$ of points (x_i, y_i) in the integer grid $\mathbb{Z}^2$, let

$$\nu(\mathcal{S}) = \min \max\{|x_1|, \ldots, |x_n|, |y_1|, \ldots, |y_n|\},$$

the minimum being taken over all configurations $\mathcal{S}'$ of the same order type as $\mathcal{S}$, and let $\nu^*(n) = \max \nu(\mathcal{S})$ over all n-point configurations. Then, for some $c_1, c_2 > 0$,

$$2^{2^{c_1 n}} \leq \nu^*(n) \leq 2^{2^{c_2 n}}.$$

It is conceivable, however, that a reasonably small integral representative may still be used to "approximate" an order type, perhaps in the following sense.

PROBLEM 5.6.8 *Goodman, Pollack, and Sturmfels*
Viewing a simple order type of n points as a binary $\binom{n}{3}$-vector, and measuring the distance between two order types by the Hamming metric, let $S(n)$ be the set of simple planar order types of n points that can be represented on an $n \times n$ grid. Find the smallest integer $r = r(n)$ such that every simple order type of n points has Hamming distance at most r from some order type in $S(n)$.

REALIZABILITY

Along with the Universality Theorem of Section 5.5, Mnëv proved that the problem of determining whether a given pseudoline arrangement is stretchable is NP-hard, in fact as hard as the problem of solving general systems of polynomial equations and inequalities over $\mathbb{R}$ (cf. Chapter 29 of this Handbook):

THEOREM 5.6.9 [Mnë85, Mnë88]

The stretchability problem for pseudoline arrangements is polynomially equivalent to the "existential theory of the reals" decision problem.

Shor [Sho91] presents a more compact proof of the NP-hardness result, by encoding a so-called "monotone 3-SAT" formula in a family of suitably modified Pappus and Desargues configurations that turn out to be stretchable if and only if the corresponding formula is satisfiable. (See also [Ric96].)

The following result provides an upper bound for the realizability problem.

THEOREM 5.6.10 [BLS$^+$93]

The stretchability problem for pseudoline arrangements can be decided in singly exponential time and polynomial space in the Turing machine model of complexity. The number of arithmetic operations needed is bounded above by $2^{4n \log n + O(n)}$.

The NP-hardness does not mean, however, that it is pointless to look for algorithms to determine stretchability, particularly in special cases. Indeed, a good deal of work has been done on this problem by Bokowski, in collaboration with Guedes de Oliveira, Pock, Richter-Gebert, and Sturmfels. Four main algorithmic methods have been developed to test for the realizability (or nonrealizability) of an oriented matroid, i.e., in the rank 3 case, the stretchability (respectively nonstretchability) of a pseudoline arrangement:

(i) The ***inequality reduction method:*** this attempts to find a relatively small system of inequalities that still carries all the information about a given oriented matroid;

(ii) The ***solvability sequence method:*** this attempts to find an elimination order with special properties for the coordinates in a potential realization of an order type;

(iii) The ***final polynomial method:*** this attempts to find a bracket polynomial (cf. Chapter 48) whose existence will imply the *non*realizability of an order type;

(iv) Pock's ***rubber-band method:*** an elementary heuristic that has proven surprisingly effective in finding realizations [Poc91].

Not every realizable order type has a solvability sequence, but it turns out that every nonrealizable one does have a final polynomial, and an algorithm due to Lombardi can be used to find one [BLS$^+$93].

All of these methods extend to higher dimensions. For details about the first three, see [BS89].

CONSTRUCTING ARRANGEMENTS

An $O(n^2)$ algorithm is given in [EOS86, ESS93] to "construct" an arrangement $\mathcal{A}$ of lines (hyperplanes, in general, in time $O(n^d)$), i.e., to construct its face lattice. This algorithm is used as a subroutine in a number of other algorithms in computational geometry (see [Ede87]). From this one can find the λ-matrix of $\mathcal{A}$ in time $O(n^2)$, which is optimal.

SORTING INTERSECTIONS OF LINES OR PSEUDOLINES

Steiger and Streinu consider the problem of x-sorting line or pseudoline intersections, i.e., determining the order of the x-coordinates of the intersections of the lines in an arrangement or of the pseudolines in a wiring diagram. They prove:

THEOREM 5.6.11 [SS94]

(i) *There is a decision tree of depth $O(n^2)$ to x-sort the vertices of a simple arrangement of n lines;*

(ii) $\Omega(n^2 \log n)$ *comparisons are necessary to x-sort the vertices of a simple arrangement of n pseudolines.*

(The second statement is a corollary of Theorem 5.6.1.)

Even though this is only a "pseudo-algorithmic" distinction, since it holds in the decision-tree model of computation, nevertheless this result is one of the few known instances where there is a clear computational difference between lines and pseudolines.

5.7 APPLICATIONS

Planar arrangements of lines and pseudolines, as well as point configurations, arise in many problems of computational geometry. Here we describe several such applications involving pseudolines in particular.

GLOSSARY

Tangent visibility graph of a set of pairwise disjoint convex objects: The graph formed by the tangents to pairs of objects, cut off at their points of tangency (provided these segments do not meet any other objects) and by the arcs into which they divide the boundaries of the objects.

TOPOLOGICAL SWEEP

The original idea behind what has come to be known as *topologically sweeping an arrangement* was applied, by Edelsbrunner and Guibas, to the case of an arrangement of straight lines. In order to construct the arrangement, rather than using a line to sweep it, they used a pseudoline, and achieved a saving of a factor of $\log n$ in the time required, while keeping the storage linear.

THEOREM 5.7.1 [Ede87]

The cell complex of an arrangement of n lines in the plane can be computed in $O(n^2)$ time and $O(n)$ space by sweeping a pseudoline across it.

This result can be applied to a number of problems, and results in an improvement of known bounds on each: minimum area triangle spanned by points, visibility

graph of segments, and (in higher dimensions) enumerating faces of a hyperplane arrangement and testing for degeneracies in a point configuration.

The idea of a topological sweep was then generalized, by Snoeyink and Hershberger, to sweeping a pseudoline across an arrangement of *pseudolines*; they prove the possibility of such a sweep (Theorem 5.5.2), and show that it can be performed in the same time and space as in Theorem 5.7.1. They also apply this result to finding a short Boolean formula for a polygon with curved edges.

The topological sweep method was also used by Chazelle and Edelsbrunner [CE92] to report all k-segment intersections in an arrangement of n line segments in (optimal) $O(n \log n + k)$ time, and has been generalized to higher dimensions.

PSEUDO-TRIANGULATIONS

Pocchiola and Vegter introduced the concept of a pseudo-triangulation (see Section 5.3 above) in order to compute the visibility graph of a collection of pairwise disjoint convex obstacles. Then they showed that a collection of disjoint pseudotriangles dualizes to a pseudoline arrangement, and that certain pseudoline arrangements could be realized in this way by collections of pseudotriangles. This enables them to generalize certain algorithms for configurations of points to configurations of more general convex objects.

Their results include the following.

THEOREM 5.7.2 [PV94]

Given a collection of n disjoint convex objects in the plane, a pseudo-triangulation can be computed in $O(n \log n)$ time, the dual arrangement in $O(n^2)$ time and space, and the tangent visibility graph in $O(n^2)$ time and linear space.

PSEUDO-VISIBILITY

In a series of recent papers, O'Rourke and Streinu introduce what they call the "vertex-edge visibility graph" of a polygon, which encodes more information than the standard vertex visibility graph, and use it to study the visibility problem in the polygon. They then generalize this concept to pseudopolygons, whose vertices and edges come from generalized configurations of points (see Section 5.2), and show that the reconstruction problem for vertex-edge visibility graphs can be solved provided pseudopolygons are permitted. They prove

THEOREM 5.7.3 [OS96]

There is a polynomial-time algorithm for the problem of deciding whether a graph is the vertex-edge pseudo-visibility graph of a pseudopolygon.

COROLLARY 5.7.4 [OS96]

The decision problem for vertex visibility graphs of pseudopolygons is in NP.

(This last result is in contrast to the fact that the same problem with straight-edge visibility is only known to be in PSPACE.)

5.8 SOURCES AND RELATED MATERIAL

FURTHER READING

[BLS+93]: A comprehensive account of oriented matroid theory, including a great many references; most references not given explicitly in this chapter can be traced through this book.

[Ede87]: An introduction to computational geometry, focusing on arrangements and their algorithms.

[GP91, GP93]: Two surveys on allowable sequences and order types and their complexity.

[Grü72]: A monograph on planar arrangements and their generalizations, with excellent problems (many still unsolved) and a very complete bibliography up to 1972.

RELATED CHAPTERS

Chapter 1: Finite point configurations
Chapter 4: Helly-type theorems and geometric transversals
Chapter 6: Oriented matroids
Chapter 21: Arrangements
Chapter 29: Computational real algebraic geometry

REFERENCES

[BEPY91] M. Bern, P. Eppstein, P. Plassmann, and F. Yao. Horizon theorems for lines and polygons. In J.E. Goodman, R. Pollack, and W. Steiger, editors, *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, pages 45–66, volume 6 of *DIMACS Series in Discrete Math. and Theor. Comput. Sci.* Amer. Math. Soc., Providence, 1991.

[BLS+93] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G.M. Ziegler. *Oriented Matroids*. Volume 46 of *Encyclopedia of Mathematics*. Cambridge University Press, 1993.

[BKLR] J. Bokowski, U. Kortenkamp, G. Laffaille, and J. Richter-Gebert. Classification of non-stretchable pseudoline arrangements and related properties. In preparation.

[BS89] J. Bokowski and B. Sturmfels. *Computational Synthetic Geometry*. Volume 1355 of *Lecture Notes in Math.* Springer-Verlag, Heidelberg, 1989.

[CE92] B. Chazelle and H. Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. *J. Assoc. Comput. Mach.* 39:1–54, 1992.

[CEG+90] K. Clarkson, H. Edelsbrunner, L. Guibas, M. Sharir, and E. Welzl. Combinatorial complexity bounds for arrangements of curves and spheres. *Discrete Comput. Geom.*, 5:99–160, 1990.

[CS93] J. Csima and E.T. Sawyer. There exist $6n/13$ ordinary points. *Discrete Comput. Geom.*, 9:187–202, 1993.

[Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer-Verlag, Berlin, 1987.

[EOS86] H. Edelsbrunner, J. O'Rourke, and R. Seidel. Constructing arrangements of lines and hyperplanes with applications. *SIAM J. Comput.*, 15:341–363, 1986.

[ESS93] H. Edelsbrunner, R. Seidel, and M. Sharir. On the zone theorem for hyperplane arrangements. *SIAM J. Comput.*, 22:418–429, 1993.

[Fel] S. Felsner. On the number of arrangements of pseudolines. *Discrete Comput. Geom.*, to appear.

[Goo80] J.E. Goodman. Proof of a conjecture of Burr, Grünbaum, and Sloane. *Discrete Math.*, 32:27–35, 1980.

[GP80] J.E. Goodman and R. Pollack. On the combinatorial classification of nondegenerate configurations in the plane. *J. Combin. Theory Ser. A*, 29:220–235, 1980.

[GP82] J.E. Goodman and R. Pollack. Helly-type theorems for pseudoline arrangements in P^2. *J. Combin. Theory Ser. A*, 32:1–19, 1982.

[GP84] J.E. Goodman and R. Pollack. Semispaces of configurations, cell complexes of arrangements. *J. Combin. Theory Ser. A*, 37:257–293, 1984.

[GP85] J.E. Goodman and R. Pollack. A combinatorial version of the isotopy conjecture. In J.E. Goodman, E. Lutwak, J. Malkevitch, and R. Pollack, editors, *Discrete Geometry and Convexity*, pages 12–19, volume 440 of *Ann. New York Acad. Sci.*, 1985.

[GP91] J.E. Goodman and R. Pollack. The complexity of point configurations. *Discrete Appl. Math.*, 31:167–180, 1991.

[GP93] J.E. Goodman and R. Pollack. Allowable sequences and order types in discrete and computational geometry. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, pages 103–134, volume 10 of *Algorithms Combin.*, Springer-Verlag, Berlin/Heidelberg, 1993.

[GPW96] J.E. Goodman, R. Pollack, and R. Wenger. There are uncountably many universal topological planes. *Geom. Dedicata*, 59:157–162, 1996.

[GPWZ94] J.E. Goodman, R. Pollack, R. Wenger, and T. Zamfirescu. Arrangements and topological planes. *Amer. Math. Monthly*, 101:866–878, 1994.

[Grü69] B. Grünbaum. The importance of being straight. In *Proc. 12th Biannual Intern. Seminar of the Canadian Math. Congress* (Vancouver, 1969), pages 243–254, 1970.

[Grü72] B. Grünbaum. *Arrangements and Spreads.* Volume 10 of *CBMS Regional Conf. Ser. in Math.* Amer. Math. Soc., Providence, 1972.

[GS93] L. Guibas and M. Sharir. Combinatorics and algorithms of arrangements. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, pages 9–36, volume 10 of *Algorithms Combin.* Springer-Verlag, Berlin/Heidelberg, 1993.

[Har85] H. Harborth. Some simple arrangements of pseudolines with a maximum number of triangles. In J.E. Goodman, E. Lutwak, J. Malkevitch, and R. Pollack, editors, *Discrete Geometry and Convexity*, pages 31–33, volume 440 of *Ann. New York Acad. Sci.*, 1985.

[Hir83] F. Hirzebruch. Arrangements of lines and algebraic surfaces. In M. Artin and J. Tate, editors, *Arithmetic and Geometry*, volume 2, pages 113–140. Birkhaüser, Boston, 1983.

[Jam85] R.E. Jamison. A survey of the slope problem. In J.E. Goodman, E. Lutwak, J. Malkevitch, and R. Pollack, editors, *Discrete Geometry and Convexity*, pages 34–51, volume 440 of *Ann. New York Acad. Sci.*, 1985.

[Knu92] D.E. Knuth. *Axioms and Hulls.* Volume 606 of *Lecture Notes in Comput. Sci.* Springer-Verlag, Berlin/Heidelberg, 1992.

[Lev26] F. Levi. Die Teilung der projektiven Ebene durch Gerade oder Pseudogerade. *Ber. Math.-Phys. Kl. Sächs. Akad. Wiss.*, 78:256–267, 1926.

[Mar93] N. Martinov. Classification of arrangements by the number of their cells. *Discrete Comput. Geom.*, 9:39–46, 1993.

[Mat91] J. Matoušek. Lower bounds on the length of monotone paths in arrangements. *Discrete Comput. Geom.*, 6:129–134, 1991.

[Mnë85] N.E. Mnëv. On manifolds of combinatorial types of projective configurations and convex polyhedra. *Soviet Math. Dokl.*, 32:335–337, 1985.

[Mnë88] N.E. Mnëv. The universality theorems on the classification problem of configuration varieties and convex polytopes varieties. In O.Ya. Viro, editor, *Topology and Geometry—Rohlin Seminar*, pages 527–544, volume 1346 of *Lecture Notes in Math.* Springer-Verlag, Berlin, 1988.

[OS96] J. O'Rourke and I. Streinu. Pseudo-visibility graphs in pseudo-polygons: Part II. Preprint, Smith College, 1996.

[PV94] M. Pocchiola and G. Vegter. Order types and visibility types of configurations of disjoint convex plane sets. Extended abstract, Tech. Report 94-4, Labo. d'Inf. de l'ENS, Paris, 1994.

[Poc91] K.P. Pock. *Entscheidungsmethoden zur Realisierbarkeit orientierter Matroide.* Diplomarbeit, TH Darmstadt, 1991.

[Ric89] J. Richter. Kombinatorische Realisierbarkeitskriterien für orientierte Matroide. *Mitt. Math. Sem. Gießen*, 194:1–112, 1989.

[Ric96] J. Richter-Gebert. *Realization Spaces of Polytopes.* Volume 1643 of *Lecture Notes in Math.*, Springer-Verlag, Berlin/Heidelberg, 1996.

[Rin56] G. Ringel. Teilungen der Ebene durch Geraden oder topologische Geraden. *Math. Z.*, 64:79–102, 1956.

[Rin57] G. Ringel. Über Geraden in allgemeiner Lage. *Elem. Math.*, 12:75–82, 1957.

[Rou88] J.-P. Roudneff. Tverberg-type theorems for pseudoconfigurations of points in the plane. *European J. Combin.*, 9:189–198, 1988.

[Rou96] J.-P. Roudneff. The maximum number of triangles in arrangements of (pseudo-) lines. *J. Combin. Theory Ser. B*, 66:44–74, 1996.

[SE88] P. Salamon and P. Erdős. The solution to a problem of Grünbaum. *Canad. Math. Bull.*, 31:129–138, 1988.

[Sho91] P. Shor. Stretchability of pseudolines is NP-hard. In P. Gritzmann and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics—The Victor Klee Festschrift*, pages 531–554, volume 4 of *DIMACS Series in Discrete Math. and Theor. Comput. Sci.* Amer. Math. Soc., Providence, 1991.

[SH91] J. Snoeyink and J. Hershberger. Sweeping arrangements of curves. In J.E. Goodman, R. Pollack, and W. Steiger, editors, *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, pages 309–349, volume 6 of *DIMACS Series in Discrete Math. and Theor. Comput. Sci.* Amer. Math. Soc., Providence, 1991.

[SS94] W. Steiger and I. Streinu. A pseudo-algorithmic separation of lines from pseudo-lines. *Proc. 6th Ann. Canad. Conf. on Comput. Geom.*, 1994, pages 7–11.

6 ORIENTED MATROIDS

Jürgen Richter-Gebert and Günter M. Ziegler

INTRODUCTION

The theory of *oriented matroids* provides a broad setting in which to model, describe, and analyze combinatorial properties of geometric configurations. Mathematical objects that are apparently totally distinct, such as *point and vector configurations*, *arrangements of hyperplanes*, *convex polytopes*, *directed graphs*, and *linear programs* find a common generalization in the language of oriented matroids.

The oriented matroid of a finite set of points $\boldsymbol{P}$ extracts "relative position" and "orientation" information from the configuration; for example, it can be given by a list of signs that encodes the orientations of all the bases of $\boldsymbol{P}$. In the passage from a concrete point configuration to its oriented matroid, metrical information is lost, but many structural properties of $\boldsymbol{P}$ have their counterparts at the—purely combinatorial—level of the oriented matroid.

We first introduce oriented matroids in the context of several models and motivations (Section 6.1). Then we present some equivalent axiomatizations (Section 6.2). Finally, we discuss concepts that play central roles in the theory of oriented matroids (Section 6.3), among them *duality*, *realizability*, the study of *simplicial cells*, and the treatment of *convexity*.

6.1 MODELS AND MOTIVATIONS

This section discusses geometric examples that are usually treated on the level of concrete coordinates, but where an "oriented matroid point of view" gives deeper insight. We also present these examples as standard models that provide intuition for the behavior of general oriented matroids.

6.1.1 ORIENTED BASES OF VECTOR CONFIGURATIONS

GLOSSARY

Vector configuration: A matrix $X = (x_1, \ldots, x_n) \in (\mathbb{R}^d)^n$, usually assumed to have full rank d.

Matroid of X: The pair $M_X = (E, \mathcal{B}_X)$, where $E := \{1, 2, \ldots, n\}$ and $\mathcal{B}_X$ is the set of all (column indices) of bases of X.

Matroid: A pair $M = (E, \mathcal{B})$, where E is a finite set, and $\mathcal{B} \subseteq 2^E$ is a nonempty collection of subsets of E (the ***bases*** of M) that satisfies the ***Steinitz exchange***

axiom: For all $B_1, B_2 \in \mathcal{B}$ and $e \in B_1 \backslash B_2$, there exists an $f \in B_2 \backslash B_1$ such that $(B_1 \backslash e) \cup f \in \mathcal{B}$.

Signs: Elements of the set $\{-, 0, +\}$, used as a shorthand for the corresponding elements of $\{-1, 0, +1\}$.

Chirotope of X: The map

$$\begin{array}{rccl} \chi_X\colon & E^d & \to & \{-,0,+\} \\ & (\lambda_1,\ldots,\lambda_d) & \mapsto & \text{sign}(\det(x_{\lambda_1},\ldots,x_{\lambda_d})). \end{array}$$

Ordinary (unoriented) *matroids*, as introduced in 1935 by Whitney (see Oxley [Oxl92]), can be considered as an abstraction of vector configurations in finite dimensional vector spaces over arbitrary fields. All the bases of a matroid M have the same cardinality d, which is called the ***rank*** of the matroid. Equivalently, we can identify M with the characteristic function of the bases $B_M\colon E^d \to \{0,1\}$, where $B_M(\lambda) = 1$ if and only if $\{\lambda_1,\ldots,\lambda_d\} \in \mathcal{B}$.

One can obtain examples of matroids as follows: take a finite set of vectors

$$X = \{x_1, x_2, \ldots, x_n\} \subseteq K^d$$

of rank d in a finite-dimensional vector space K^d and consider the set of bases of K^d formed by subsets of the points in X. In other words, the pair

$$M_X = (E, \mathcal{B}_X) = \Big(\{1,\ldots,n\},\ \big\{\{\lambda_1,\ldots,\lambda_d\} \mid \det(x_{\lambda_1},\ldots,x_{\lambda_d}) \neq 0\big\} \Big)$$

forms a matroid.

The basic information about the incidence structure of the points in X is contained in the underlying matroid M_X. However, the matroid alone presents only a weak model of a geometric configuration; for example, all configurations of n points in ***general position*** in the plane (i.e., no three points on a line) have the same matroid $M = U_{3,n}$: here no information beyond the dimension and size of the configuration, and the fact that it is in general position, is retained for the matroid.

In contrast to matroids, the theory of ***oriented matroids*** considers the structure of dependencies in vector spaces over *ordered* fields. Roughly speaking, an oriented matroid is a matroid where in addition every basis is equipped with an orientation. These oriented bases have to satisfy an oriented version of the Steinitz exchange axiom (to be described later). In other words, oriented matroids not only describe the incidence structure between the points of X and the hyperplanes spanned by points of X (this is the matroid information); they also encode the positions of the points relative to the hyperplanes: "Which points lie on the positive side of a hyperplane, which points lie on the negative side, and which lie on the hyperplane?" If $X \in (K^d)^n$ is a configuration of n points in a d-dimensional vector space K^d over an ordered field K, we can describe the corresponding oriented matroid χ_X by the function:

$$\begin{array}{rccl} \chi_X\colon & E^d & \to & \{-,0,+\} \\ & (\lambda_1,\ldots,\lambda_d) & \mapsto & \text{sign}(\det(x_{\lambda_1},\ldots,x_{\lambda_d})). \end{array}$$

This map χ_X is called the ***chirotope*** of X and is very closely related to the oriented matroid of X. It encodes much more information than the corresponding matroid, including information about the *topology* and the *convexity* of the underlying configurations.

6.1.2 ARRANGEMENTS OF POINTS

GLOSSARY

Affine point configuration: A matrix $(p_1, \ldots, p_n) \in (\mathbb{R}^{d-1})^n$, usually assumed to have full rank $d-1$.

Associated vector configuration: The matrix $X \in (\mathbb{R}^d)^n$ obtained from a point configuration by adding a row of ones. This corresponds to the embedding of the affine space $\mathbb{R}^{d-1}$ into the linear vector space $\mathbb{R}^d$ via $p \mapsto x = \binom{p}{1}$.

Oriented matroid of a point configuration: The oriented matroid of the associated vector configuration.

Covector of a vector configuration X: Partition of $X = (x_1, \ldots, x_n)$ induced by a linear hyperplane, into points on the hyperplane, on the positive side, and on the negative side.

Oriented matroid of X: The collection $\mathcal{L} \subseteq \{-, 0, +\}^n$ of all covectors of X.

Let $X := (x_1, \ldots, x_n) \in (\mathbb{R}^d)^n$ be an $n \times d$ matrix and let $E := \{1, \ldots, n\}$. We interpret the columns of X as n vectors in the d-dimensional real vector space $\mathbb{R}^d$. For a linear functional $y^T \in (\mathbb{R}^d)^*$ we set

$$C_X(y) = (\text{sign}(y^T x_1), \ldots, \text{sign}(y^T x_n)).$$

Such a sign vector is called a ***covector*** of X. We denote the collection of all covectors of X by

$$\mathcal{L}_X := \{C_X(y) \mid y \in \mathbb{R}^d\}.$$

The pair $\mathcal{M}_X = (E, \mathcal{L}_X)$ is called the ***oriented matroid*** of X. Here each sign vector $C_X(y) \in \mathcal{L}_X$ describes the positions of the vectors $x_1, \ldots, x_n$ relative to the linear hyperplane $H_y = \{x \in \mathbb{R}^d \mid y^T x = 0\}$: the sets

$$\begin{aligned} C_X(y)^0 &:= \{e \in E \mid C_X(y)_e = 0\} \\ C_X(y)^+ &:= \{e \in E \mid C_X(y)_e > 0\} \\ C_X(y)^- &:= \{e \in E \mid C_X(y)_e < 0\} \end{aligned}$$

describe how H_y partitions the set of points X. Here $C_X(y)^0$ contains the points on H_y, while $C_X(y)^+$ and $C_X(y)^-$ contain the points on the positive and on the negative side of H_y, respectively. In particular, if $C_X(y)^- = \emptyset$, then all points not on H_y lie on the positive side of H_y. In other words, in this case H_y determines a face of the positive cone

$$\text{pos}(x_1, \ldots, x_n) := \left\{ \lambda_1 x_1 + \lambda_2 x_2 + \ldots + \lambda_n x_n \mid 0 \leq \lambda_i \in \mathbb{R} \text{ for } 1 \leq i \leq n \right\}$$

of all points of X. The face lattice of the cone $\text{pos}(X)$ can be recovered from $\mathcal{L}_X$. It is simply the set $\mathcal{L}_X \cap \{+, 0\}^E$, partially ordered by the order induced from the relation "$0 < +$."

If in the configuration X we have $x_{i,d} = 1$ for all $1 \leq i \leq n$, then we can consider X as representing homogeneous coordinates of an *affine* point set X' in $\mathbb{R}^{d-1}$.

Here the affine points correspond to the original points x_i after removal of the d-th coordinate. The face lattice of the convex polytope $\text{conv}(X') \subseteq \mathbb{R}^{d-1}$ is then identical to the face lattice of $\text{pos}(X)$. Hence, $\mathcal{M}_X$ can be used to recover the *convex hull* of X'.

Thus oriented matroids are generalizations of point configurations in linear or affine spaces. For general oriented matroids we weaken the assumption that the hyperplanes spanned by points of the configuration are really flat to the assumption that they only satisfy certain topological incidence properties. Nonetheless, this kind of picture is sometimes misleading since not all oriented matroids have this type of representation (compare the "Type II representations" of [BLS+93, Section 5.3]).

6.1.3 ARRANGEMENTS OF HYPERPLANES AND OF HYPERSPHERES

GLOSSARY

Hyperplane arrangement $\mathcal{H}$: Collection of (oriented) linear hyperplanes in $\mathbb{R}^d$, given by normal vectors $x_1, \ldots, x_n$.

Hypersphere arrangement induced by $\mathcal{H}$: Intersection of $\mathcal{H}$ with the unit sphere S^{d-1}.

Covectors of $\mathcal{H}$: Sign vectors of the cells in $\mathcal{H}$; equivalently, $\mathbf{0}$ together with the sign vectors of the cells in $\mathcal{H} \cap S^{d-1}$.

We obtain a different picture if we polarize the situation and consider *hyperplane arrangements* rather than arrangements of points. For a real matrix $X := (x_1, \ldots, x_n) \in (\mathbb{R}^d)^n$ consider the system of hyperplanes $\mathcal{H}_X := (H_1, \ldots, H_n)$ with

$$H_i := \{y \in \mathbb{R}^d \mid y^T x_i = 0\}.$$

Each vector x_i induces an orientation on H_i by defining

$$H_i^+ := \{y \in \mathbb{R}^d \mid y^T x_i > 0\}$$

to be the ***positive side*** of H_i. We define H_i^- analogously to be the ***negative side*** of H_i. To avoid degenerate cases we assume that X contains at least one proper basis (i.e., the matrix X has rank d). The hyperplane arrangement $\mathcal{H}_X$ subdivides $\mathbb{R}^d$ into polyhedral cones. Without loss of information we can intersect with the unit sphere S^{d-1} and consider the sphere system

$$\mathcal{S}_X := (H_1 \cap S^{d-1}, \ldots, H_n \cap S^{d-1}) = \mathcal{H}_X \cap S^{d-1}.$$

Our assumption that X contains at least one proper basis translates to the fact that the intersection of all $H_1 \cap \ldots \cap H_n \cap S^{d-1}$ is empty. $\mathcal{H}_X$ induces a cell decomposition $\Gamma(\mathcal{S}_X)$ on S^{d-1}. Each face of $\Gamma(\mathcal{S}_X)$ corresponds to a sign vector in $\{-, 0, +\}^E$ that indicates the position of the cell with respect to the $(d-2)$-spheres $H_i \cap S^{d-1}$ (and therefore with respect to the hyperplanes H_i) of the arrangement. The list of all these sign vectors is exactly the set $\mathcal{L}_X$ of covectors of $\mathcal{H}_X$.

While the visualization of oriented matroids by sets of points in $\mathbb{R}^n$ does not fully generalize to the case of nonrepresentable oriented matroids, the picture of

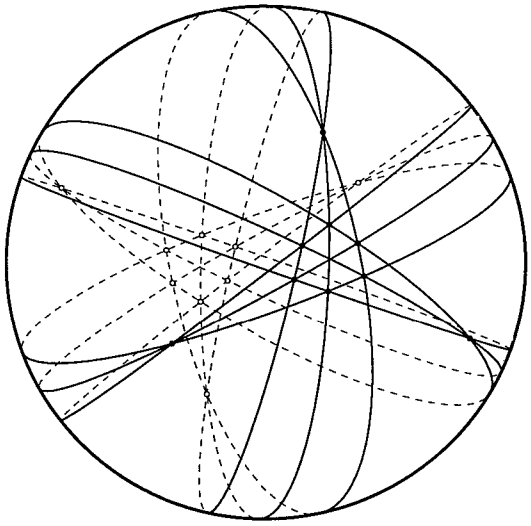


FIGURE 6.1.1
An arrangement of nine great circles on S^2. The arrangement corresponds to a Pappus configuration.

hyperplane arrangements has a well-defined analogue that also covers all the non-realizable cases. We will see that as a consequence of the topological representation theorem of Folkman and Lawrence (Section 6.2.4) every rank d oriented matroid can be represented as an arrangement of oriented *pseudospheres* (or pseudohyperplanes) embedded in the S^{d-1} (resp. in $\mathbb{R}^d$). Arrangements of pseudospheres are systems of topological $(d-2)$-spheres embedded in S^{d-1} that satisfy certain intersection properties that clearly hold in the case of "straight" arrangements.

6.1.4 ARRANGEMENTS OF PSEUDOLINES

GLOSSARY

Pseudoline: Simple closed curve p in the projective plane $\mathbb{R}\mathrm{P}^2$ that is topologically equivalent to a line (i.e., there is a self-homeomorphism of $\mathbb{R}\mathrm{P}^2$ mapping p to a straight line).

Arrangement of pseudolines: Collection of pseudolines $\mathcal{P} := (p_1, \ldots, p_n)$ in the projective plane, where any two of them intersect exactly once.

Simple arrangement: No three pseudolines meet in a common point. (Equivalently, the associated oriented matroid is ***uniform***.)

Equivalent arrangements: Arrangements $\mathcal{P}_1$ and $\mathcal{P}_2$ that generate isomorphic cell decompositions of $\mathbb{R}\mathrm{P}^2$. (In this case there exists a self-homeomorphism of $\mathbb{R}\mathrm{P}^2$ mapping $\mathcal{P}_1$ to $\mathcal{P}_2$.)

Stretchable arrangement of pseudolines: An arrangement that is equivalent to an arrangement of projective lines.

An *arrangement of pseudolines* in the projective plane is a collection of pseudolines such that any two pseudolines intersect in exactly one point, where they

cross. (See Grünbaum [Grü72] and Richter [Ric89].) We will always assume that $\mathcal{P}$ is ***essential***, i.e., that the intersection of all the pseudolines p_i is empty.

An arrangement of pseudolines behaves in many respects just like an arrangement of n lines in the projective plane. (In fact, there are only very few combinatorial theorems known that are true for straight arrangements, but not true in general for pseudoarrangements.)

Figure 6.1.2 shows a small example of a nonstretchable arrangement of pseudolines. (It is left as a challenging exercise to the reader to prove the nonstretchability.) Up to isomorphism this is the only simple nonstretchable arrangement of 9 pseudolines [Ric89] [Knu92]; every arrangement of 8 (or fewer) pseudolines is stretchable.

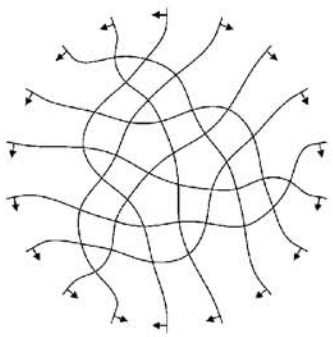


FIGURE 6.1.2
A nonstretchable arrangement of nine pseudolines. It was obtained by Ringel [Rin56] *as a perturbation of the Pappus configuration.*

To associate with a projective arrangement $\mathcal{P}$ an oriented matroid we represent the projective plane (as customary) by the 2-sphere on which antipodal points are identified. With this every arrangement of pseudolines gives rise to an arrangement of *great pseudocircles* on S^2. For each great pseudocircle on S^2 we choose a positive side. Each cell induced by $\mathcal{P}$ on S^2 now corresponds to a unique sign vector. The collection of all these sign vectors again forms a set of covectors $\mathcal{L}_{\mathcal{P}} \backslash \mathbf{0}$ of an oriented matroid of rank 3. Conversely, as a special case of the topological representation theorem, *every* oriented matroid of rank 3 has a representation by an *oriented* pseudoline arrangement.

In this way we can use pseudoline arrangements as a standard picture to represent rank 3 oriented matroids. The easiest picture is obtained when we restrict ourselves to the upper hemisphere of S^2 and assume w.l.o.g. that each pseudoline crosses the equator exactly once, and that the crossings are distinct (i.e., no intersection of the great pseudocircles lies on the equator). Then we can represent this upper hemisphere by an arrangement of mutually crossing, oriented affine pseudolines in the plane $\mathbb{R}^2$. (We did this implicitly while drawing Figure 6.1.2.)

By means of this equivalence, all problems concerning pseudoline arrangements can be translated to the language of oriented matroids. For instance, the problem of stretchability is equivalent to the realizability problem for oriented matroids.

6.2 AXIOMS AND REPRESENTATIONS

In this section we define oriented matroids formally. It is one of the main features of oriented matroid theory that the same object can be viewed under quite dif-

ferent aspects. This results in the fact that there are many different equivalent axiomatizations, and it is sometimes very useful to "jump" from one point of view to another. Statements that are difficult to prove in one language may be easy in another. For this reason we present here several different axiomatizations. We also give a (partial) dictionary of how to translate among them. For a complete version of the basic equivalence proofs—which are highly nontrivial—see [BLS+93, Chapters 3 and 5].

We will give axiomatizations of oriented matroids for the following four types of representations:

- Collections of covectors,
- Collections of cocircuits,
- Signed bases,
- Arrangements of pseudospheres.

In the last part of this section these concepts are illustrated by an example.

GLOSSARY

Sign vector: Vector C in $\{-,0,+\}^E$, where E is a finite index set, usually $\{1,\dots,n\}$. For $e \in E$, the e-component of C is denoted by C_e.

Positive, negative, and zero part of C:

$$\begin{aligned} C^+ &:= \{e \in E \mid C_e = +\}, \\ C^- &:= \{e \in E \mid C_e = -\}, \\ C^0 &:= \{e \in E \mid C_e = 0\}. \end{aligned}$$

Support of C: $\underline{C} := \{e \in E \mid C_e \neq 0\}$.

Zero vector: $\mathbf{0} := (0,\dots,0) \in \{-,0,+\}^E$.

Negative of a sign vector: $-C$, defined by $(-C)^+ := C^-$, $(-C)^- := C^+$ and $(-C)^0 = C^0$.

Composition of C and D: $(C \circ D)_e := \begin{cases} C_e & \text{if } C_e \neq 0, \\ D_e & \text{otherwise.} \end{cases}$

Separation set of C and D: $S(C,D) := \{e \in E \mid C_e = -D_e \neq 0\}$.

We partially order the set of sign vectors by "$0 < +$" and "$0 < -$." The partial order on sign vectors, denoted by $C \leq D$, is understood componentwise; equivalently, we have

$$C \leq D \iff \left[C^+ \subseteq D^+ \text{ and } C^- \subseteq D^- \right].$$

For instance, if $C := (+,+,-,0,-,+,0,0)$ and $D := (0,0,-,+,+,-,0,-)$, then we have:

$$C^+ = \{1,2,6\}, \quad C^- = \{3,5\}, \quad C^0 = \{4,7,8\}, \quad \underline{C} = \{1,2,3,5,6\},$$

$$C \circ D = (+,+,-,+,-,+,0,-), \quad C \circ D \geq C, \quad S(C,D) = \{5,6\}.$$

Furthermore, for $x \in \mathbb{R}^n$, we denote by $\sigma(x) \in \{-,0,+\}^E$ the image of x under the componentwise sign function σ that maps $\mathbb{R}^n$ to $\{-,0,+\}^E$.

6.2.1 COVECTOR AXIOMS

Definition: An ***oriented matroid*** given in terms of its covectors is a pair $\mathcal{M} := (E, \mathcal{L})$, where $\mathcal{L} \in \{-, 0, +\}^E$ satisfies

(CV0) $\mathbf{0} \in \mathcal{L}$,

(CV1) $C \in \mathcal{L} \implies -C \in \mathcal{L}$

(CV2) $C, D \in \mathcal{L} \Longrightarrow C \circ D \in \mathcal{L}$

(CV3) $C, D \in \mathcal{L}$, $e \in S(C, D) \Longrightarrow$
there is a $Z \in \mathcal{L}$ with $Z_e = 0$ and with $Z_f = (C \circ D)_f$ for $f \in E \backslash S(C, D)$.

It is not difficult to check that these covector axioms are satisfied by the sign vector system $\mathcal{L}_X$ of the cells in a hyperplane arrangement $\mathcal{H}_X$, as defined in the last section. The first two axioms are satisfied trivially. For (CV2) assume that x_C and x_D are points in $\mathbb{R}^d$ with $\sigma(x_C^T \cdot X) = C \in \mathcal{L}_X$ and $\sigma(x_D^T \cdot X) = D \in \mathcal{L}_X$. Then (CV2) is implied by the fact that for sufficiently small $\epsilon > 0$ we have $\sigma((x_C + \epsilon x_D)^T \cdot X) = C \circ D$. The geometric content of (CV3) is that if $H_e := \{y \in \mathbb{R}^d \mid y^T x_e = 0\}$ is a hyperplane separating x_C and x_D then there exists a point x_Z on H_e with the property that x_Z is on the same side as x_C and x_D for all hyperplanes not separating x_C and x_D. We can find such a point by intersecting H_e with the line segment that connects x_C and x_D.

As we will see later the partially ordered set $(\mathcal{L}, \leq)$ describes the face lattice of a cell decomposition of the sphere S^{d-1} by pseudohyperspheres. Each sign vector corresponds to a face of the cell decomposition. We define the ***rank*** d of $\mathcal{M} = (E, \mathcal{L})$ to be the (unique) length of the maximal chains in $(\mathcal{L}, \leq)$ minus one. In the case of realizable arrangements $\mathcal{S}_X$ of hyperspheres, the lattice $(\mathcal{L}_X, \leq)$ equals the face lattice of $\Gamma(\mathcal{S}_X)$.

6.2.2 COCIRCUITS

The covectors of (inclusion) minimal support in $\mathcal{L} \backslash \mathbf{0}$ correspond to the 0-faces (=vertices) of the cell decomposition. We call the set $\mathcal{C}^*(\mathcal{M})$ of all such minimal covectors the ***cocircuits*** of $\mathcal{M}$. Oriented matroids can be described by their set of cocircuits, as shown by the following theorem.

THEOREM 6.2.1 *Cocircuit Characterization*
A collection $\mathcal{C}^* \in \{-, 0, +\}^E$ is the set of cocircuits of an oriented matroid $\mathcal{M}$ if and only if it satisfies

(CC0) $\mathbf{0} \not\in \mathcal{C}^*$

(CC1) $C \in \mathcal{C}^* \implies -C \in \mathcal{C}^*$

(CC2) *For all $C, D \in \mathcal{C}^*$ we have: $\underline{C} \subseteq \underline{D} \Longrightarrow C = D$ or $C = -D$.*

(CC3) *$C, D \in \mathcal{C}^*$, $C \neq -D$, and $e \in S(C, D) \implies$*
there is a $Z \in \mathcal{C}^*$ with $Z^+ \subseteq (C^+ \cup D^+) \backslash \{e\}$ and $Z^- \subseteq (C^- \cup D^-) \backslash \{e\}$.

THEOREM 6.2.2 *Covector/Cocircuit Translation*

For every oriented matroid $\mathcal{M}$, one can uniquely determine the set $\mathcal{C}^*$ of cocircuits from the set $\mathcal{L}$ of covectors of $\mathcal{M}$, and conversely, as follows:

(i) *$\mathcal{C}^*$ is the set of vectors with minimal support in $\mathcal{L}\backslash\mathbf{0}$:*
$\mathcal{C}^* = \{C \in \mathcal{L}\backslash\{\mathbf{0}\} \mid C' \leq C \Longrightarrow C' \in \{\mathbf{0}, C\}\}$

(ii) *$\mathcal{L}$ is the set of all sign vectors obtained by successive composition of a finite number of cocircuits from $\mathcal{C}^*$:*
$\mathcal{L} = \{C_1 \circ \ldots \circ C_k \mid k \geq 0,\ C_1, \ldots, C_k \in \mathcal{C}^*\}$.

6.2.3 CHIROTOPES

GLOSSARY

Alternating sign map: A map $\chi: E^d \longrightarrow \{-, 0, +\}$ such that any transposition of two components changes the sign: $\chi(\tau_{ij}(\lambda)) = -\chi(\lambda)$.

Chirotope: An alternating sign map χ that encodes the basis orientations of an oriented matroid $\mathcal{M}$ of rank d.

We now present an axiom system for *chirotopes*, which characterizes oriented matroids in terms of basis orientations. Here an algebraic connection to determinant identities becomes obvious. Chirotopes are the main tool for translating problems in oriented matroid theory to an algebraic setting [BSt89]. They also form a description of oriented matroids that is very practical for many algorithmic purposes (for instance in computational geometry, see Knuth [Knu92]).

Definition: Let $E := \{1, \ldots, n\}$ and $0 \leq d \leq n$. A ***chirotope of rank d*** is an alternating sign map $\chi: E^d \to \{-, 0, +\}$ that satisfies

(CHI1) The map $|\chi|: E^d \to \{0, 1\}$ that maps λ to $|\chi(\lambda)|$ is a matroid, and

(CHI2) For every $\lambda \in E^{d-2}$ and $a, b, c, d \in E\backslash\lambda$ the set

$$\left\{ \chi(\lambda, a, b) \cdot \chi(\lambda, c, d),\ -\chi(\lambda, a, c) \cdot \chi(\lambda, b, d),\ \chi(\lambda, a, d) \cdot \chi(\lambda, b, c) \right\}$$

either contains $\{-1, +1\}$, or it equals $\{0\}$.

Where does the motivation of this axiomatization come from? If we again consider a configuration $X := (x_1, \ldots, x_n)$ of vectors in $\mathbb{R}^d$, we can observe the following identity among the $d \times d$ submatrices of X:

$$\begin{aligned} &\det(x_{\lambda_1}, \ldots, x_{\lambda_{d-2}}, x_a, x_b) \cdot \det(x_{\lambda_1}, \ldots, x_{\lambda_{d-2}}, x_c, x_d) \\ -\ &\det(x_{\lambda_1}, \ldots, x_{\lambda_{d-2}}, x_a, x_c) \cdot \det(x_{\lambda_1}, \ldots, x_{\lambda_{d-2}}, x_b, x_d) \\ +\ &\det(x_{\lambda_1}, \ldots, x_{\lambda_{d-2}}, x_a, x_d) \cdot \det(x_{\lambda_1}, \ldots, x_{\lambda_{d-2}}, x_b, x_c) \quad = \quad 0 \end{aligned}$$

for all $\lambda \in E^{d-2}$ and $a, b, c, d \in E\backslash\lambda$. Such a relation is called a ***three-term Grassmann-Plücker identity***. If we compare this identity to our axiomatization, we see that (CHI2) implies that

$$\begin{aligned} \chi_X: \quad E^d &\to \{-, 0, +\} \\ (\lambda_1, \ldots, \lambda_d) &\mapsto \text{sign}(\det(x_{\lambda_1}, \ldots, x_{\lambda_d})) \end{aligned}$$

is consistent with these identities. More precisely, if we consider χ_X as defined above for a vector configuration X, the above Grassmann-Plücker identities imply that (CHI2) is satisfied. (CHI1) is also satisfied since for the vectors of X the Steinitz exchange axiom holds. (In fact the exchange axiom is a consequence of higher order Grassmann-Plücker identities.)

Consequently, χ_X is a chirotope for every $X \in (\mathbb{R}^d)^n$. Thus chirotopes can be considered as a combinatorial model of the determinant values on vector configurations. The following is not easy to prove, but essential.

THEOREM 6.2.3 *Chirotope/Cocircuit Translation*
For each chirotope χ of rank d on $E := \{1, \ldots, n\}$ the set

$$\mathcal{C}^*(\chi) \;=\; \Big\{ \big(\chi(\lambda,1), \chi(\lambda,2), \ldots, \chi(\lambda,n)\big) \;\Big|\; \lambda \in E^{d-1} \Big\}$$

forms the set of cocircuits of an oriented matroid. Conversely, for every oriented matroid $\mathcal{M}$ with cocircuits $\mathcal{C}^*$ there exists a unique pair of chirotopes $\{\chi, -\chi\}$ such that $\mathcal{C}^*(\chi) = \mathcal{C}^*(-\chi) = \mathcal{C}^*$.

The retranslation of cocircuits into signs of bases is straightforward but needs extra notation. It is omitted here.

6.2.4 ARRANGEMENTS OF PSEUDOSPHERES

GLOSSARY

The $(d-1)$-sphere: The standard unit sphere $S^{d-1} := \{x \in \mathbb{R}^d \mid ||x|| = 1\}$, or any homeomorphic image of it.

Pseudosphere: The image $s \subseteq S^{d-1}$ of the equator $\{x \in S^{d-1} \mid x_d = 0\}$ in the unit sphere under a self homeomorphism $\phi\colon S^{d-1} \to S^{d-1}$. (This definition describes topologically *tame* embeddings of a $(d-2)$-sphere in S^{d-1}. Pseudospheres behave "nicely" in the sense that they divide S^{d-1} into two sides homeomorphic to open $(d-1)$-balls.)

Oriented pseudosphere: A pseudosphere together with a choice of a positive side s^+ and a negative side s^-.

Arrangement of pseudospheres: A set of n pseudospheres in S^{d-1} with the extra condition that any subset of $d+2$ or fewer pseudospheres is ***realizable***: it defines a cell decomposition of S^{d-1} that is isomorphic to a decomposition by an arrangement of $d+2$ linear hyperplanes.

Essential arrangement: Arrangement such that the intersection of all the pseudospheres is empty.

Rank: The codimension in S^{d-1} of the intersection of all the pseudospheres. For an essential arrangement in S^{d-1}, the rank is d.

Topological representation of $\mathcal{M} = (E, \mathcal{L})$: An essential arrangement of oriented pseudospheres such that $\mathcal{L}$ is the collection of sign vectors associated with the cells of the arrangement.

One of the most important interpretations of oriented matroids is given by the topological representation theorem of Folkman and Lawrence [FL78] [BLS+93,

Chapters 4 and 5]. It states that oriented matroids are in bijection to (combinatorial equivalence classes of) *arrangements of oriented pseudospheres.* Arrangements of pseudospheres are a topological generalization of hyperplane arrangements, in the same way in which arrangements of pseudolines generalize line arrangements. Thus every rank d oriented matroid describes a certain cell decomposition of the $(d-1)$-sphere. Arrangements of pseudospheres are collections of pseudospheres that have intersection properties just like those satisfied by arrangements of proper subspheres.

Definition: A finite collection $\mathcal{P} = (s_1, s_2, \ldots, s_n)$ of pseudospheres in S^{d-1} is an *arrangement of pseudospheres* if the following conditions hold (we set $E := \{1, \ldots, n\}$):

(PS1) For all $A \subseteq E$ the set $S_A = \bigcap_{e \in A} s_e$ is a topological sphere.

(PS2) If $S_A \not\subseteq s_e$, for $A \subseteq E, e \in E$, then $S_A \cap s_e$ is a pseudosphere in S_A with sides $S_A \cap s_e^+$ and $S_A \cap s_e^-$.

Notice that this definition permits two pseudospheres of the arrangement to be identical. An entirely different, but equivalent, definition is given in the Glossary.

We see that every essential arrangement of pseudospheres $\mathcal{P}$ partitions the $(d-1)$-sphere into a regular cell complex $\Gamma(\mathcal{P})$. Each cell of $\Gamma(\mathcal{P})$ is uniquely determined by a sign vector in $\{-, 0, +\}^E$ encoding the relative position with respect to each pseudosphere s_i. Conversely, $\Gamma(\mathcal{P})$ characterizes $\mathcal{P}$ up to homeomorphism. $\mathcal{P}$ is ***realizable*** if there exists an arrangement of proper spheres $\mathcal{S}_X$ with $\Gamma(\mathcal{P}) \cong \Gamma(\mathcal{S}_X)$.

The translation of arrangements of pseudospheres to oriented matroids is given by the topological representation theorem of Folkman and Lawrence [FL78]. (For the definition of "loop," see Section 6.3.1.)

THEOREM 6.2.4 *Topological Representation Theorem (pseudosphere-covector translation)*

If $\mathcal{P}$ is an essential arrangement of pseudospheres on S^{d-1} then $\Gamma(\mathcal{P}) \cup \mathbf{0}$ forms the set of covectors of an oriented matroid of rank d. Conversely, for every oriented matroid $(E, \mathcal{L})$ of rank d (without loops) there exists an essential arrangement of pseudospheres $\mathcal{P}$ on S^{d-1} with $\Gamma(\mathcal{P}) = \mathcal{L} \backslash \mathbf{0}$.

6.2.5 DUALITY

GLOSSARY

Orthogonality: Two sign vectors $C, D \in \{-, 0, +\}^E$ are *orthogonal* if the set

$$\{C_e \cdot D_e \mid e \in E\}$$

either equals $\{0\}$ or contains $\{+, -\}$. We then write $C \perp D$.

Vector of $\mathcal{M}$: Sign vector that is orthogonal to all covectors of $\mathcal{M}$; covector of the dual oriented matroid $\mathcal{M}^*$.

Circuit of $\mathcal{M}$: Vector of minimal nonempty support; cocircuit of the dual oriented matroid $\mathcal{M}^*$.

There is a natural duality structure relating oriented matroids of rank d on n elements to oriented matroids of rank $n-d$ on n elements. It is an amazing fact that the existence of such a duality relation can be used to give another axiomatization of oriented matroids (see [BLS+93, Section 3.4]). Here we restrict ourselves to the definition of the dual of an oriented matroid $\mathcal{M}$.

THEOREM 6.2.5 *Duality*

For every oriented matroid $\mathcal{M} = (E, \mathcal{L})$ of rank d there is a unique oriented matroid $\mathcal{M}^* = (E, \mathcal{L}^*)$ of rank $|E| - d$ given by

$$\mathcal{L}^* \quad = \quad \Big\{ D \in \{-,0,+\}^E \;\Big|\; C \perp D \text{ for every } C \in \mathcal{L} \Big\}.$$

*$\mathcal{M}^*$ is called the **dual** of $\mathcal{M}$. In particular, $(\mathcal{M}^*)^* = \mathcal{M}$.*

In particular, the cocircuits of the dual oriented matroid $\mathcal{M}^*$, which we call the *circuits* of $\mathcal{M}$, also determine $\mathcal{M}$. Hence the collection $\mathcal{C}(\mathcal{M})$ of all circuits of an oriented matroid $\mathcal{M}$, given by

$$\mathcal{C}(\mathcal{M}) \quad := \quad \mathcal{C}^*(\mathcal{M}^*),$$

is characterized by the *the same* cocircuit axioms. Analogously, the *vectors* of $\mathcal{M}$ are obtained as the covectors of $\mathcal{M}^*$; they are characterized by the covector axioms.

An oriented matroid $\mathcal{M}$ is realizable if and only if its dual $\mathcal{M}^*$ is realizable. The reason for this is that a matrix $(I_d|A)$ represents $\mathcal{M}$ if and only if $(-A^T|I_{n-d})$ represents $\mathcal{M}^*$. (Here I_d denotes a $d \times d$ identity matrix, $A \in \mathbb{R}^{d\times(n-d)}$, and $A^T \in \mathbb{R}^{(n-d)\times d}$ denotes the transpose of A.)

Thus for a realizable oriented matroid $\mathcal{M}_X$ the vectors represent the linear dependencies among the columns of X, while the circuits represent minimal linear dependencies. Similarly, in the pseudoarrangements picture, circuits correspond to minimal systems of closed hemispheres that cover the whole sphere, while vectors correspond to consistent unions of such covers that never require the use of both hemispheres determined by a pseudosphere. This provides a direct geometric interpretation of circuits and vectors.

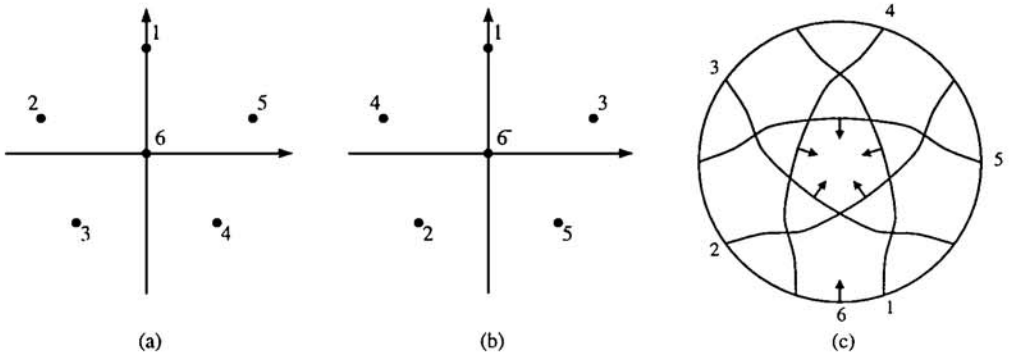
6.2.6 AN EXAMPLE

We close this section with an example that demonstrates the different representations of an oriented matroid. Consider the planar point configuration X given in Figure 6.2.1(a).

Homogeneous coordinates for X are given by

$$X := \begin{pmatrix} 0 & 3 & 1 \\ -3 & 1 & 1 \\ -2 & -2 & 1 \\ 2 & -2 & 1 \\ 3 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

FIGURE 6.2.1
An example of an oriented matroid on 6 elements.



The chirotope χ_X of $\mathcal{M}$ is given by the orientations:

$$\begin{array}{lllll}
\chi(1,2,3)=+ & \chi(1,2,4)=+ & \chi(1,2,5)=+ & \chi(1,2,6)=+ & \chi(1,3,4)=+ \\
\chi(1,3,5)=+ & \chi(1,3,6)=+ & \chi(1,4,5)=+ & \chi(1,4,6)=- & \chi(1,5,6)=- \\
\chi(2,3,4)=+ & \chi(2,3,5)=+ & \chi(2,3,6)=+ & \chi(2,4,5)=+ & \chi(2,4,6)=+ \\
\chi(2,5,6)=- & \chi(3,4,5)=+ & \chi(3,4,6)=+ & \chi(3,5,6)=+ & \chi(4,5,6)=+
\end{array}$$

Half of the cocircuits of $\mathcal{M}$ are given in the table below (the other half is obtained by negating the data):

$$\begin{array}{lll}
(0,0,+,+,+,+) & (0,-,0,+,+,+) & (0,-,-,0,+,-) \\
(0,-,-,-,0,-) & (0,-,-,+,+,0) & (+,0,0,+,+,+) \\
(+,0,-,0,+,+) & (+,0,-,-,0,-) & (+,0,-,-,+,0) \\
(+,+,0,0,+,+) & (+,+,0,-,0,+) & (+,+,0,-,-,0) \\
(+,+,+,0,0,+) & (-,+,+,0,-,0) & (-,-,+,+,0,0)
\end{array}$$

Observe that the cocircuits correspond to the point partitions produced by hyperplanes spanned by points. Half of the circuits of $\mathcal{M}$ are given in the next table. The circuits correspond to sign patterns induced by minimal linear dependencies on the rows of the matrix X. It is easy to check that every pair consisting of a circuit and a cocircuit fulfills the orthogonality condition.

$$\begin{array}{lll}
(+,-,+,-,0,0) & (+,-,+,0,-,0) & (+,-,+,0,0,-) \\
(+,-,0,+,-,0) & (+,+,0,+,0,-) & (+,-,0,0,-,+) \\
(+,0,-,+,-,0) & (+,0,+,+,0,-) & (+,0,+,0,+,-) \\
(+,0,0,+,-,-) & (0,+,-,+,-,0) & (0,+,-,+,0,-) \\
(0,+,+,0,+,-) & (0,+,0,+,+,-) & (0,0,+,-,+,-)
\end{array}$$

An affine picture of a realization of the dual oriented matroid is given in Figure 6.2.1(b). The minus-sign at point 6 indicates that a reorientation at point 6 has taken place. It is easy to check that the circuits and the cocircuits interchange their roles when dualizing the oriented matroid.

Figure 6.2.1(c) shows the corresponding arrangement of pseudolines. The circle bounding the configuration represents the projective line at infinity representing line 6.

6.3 IMPORTANT CONCEPTS

In this section we briefly introduce some very basic concepts in the theory of oriented matroids. The list of topics treated here is tailored toward some areas of oriented matroid theory that are particularly relevant for applications. Thus many other topics of great importance are left out. In particular, see [BLS+93, Section 3.3] for minors of oriented matroids, and [BLS+93, Chapter 7] for basic constructions.

6.3.1 SOME BASIC CONCEPTS

In the following glossary, we list some fundamental concepts of oriented matroid theory. Each of them can be expressed in terms of any one of the representations of oriented matroids that we have introduced (covectors, cocircuits, chirotopes, pseudoarrangements), but for each of these concepts some representations are much more convenient than others. Also, each of these concepts has some interesting properties with respect to the duality operator—which may be more or less obvious, depending on the representation that one uses.

GLOSSARY

Direct sum: An oriented matroid $\mathcal{M} = (E, \mathcal{L})$ has a ***direct sum decomposition***, denoted by $\mathcal{M} = \mathcal{M}(E_1) \oplus \mathcal{M}(E_2)$, if E has a partition into nonempty subsets E_1 and E_2 such that $\mathcal{L} = \mathcal{L}_1 \times \mathcal{L}_2$ for two oriented matroids $\mathcal{M}_1 = (E_1, \mathcal{L}_1)$ and $\mathcal{M}_2 = (E_2, \mathcal{L}_2)$. If $\mathcal{M}$ has no direct sum decomposition, then it is ***irreducible***.

Loops and coloops: A loop of $\mathcal{M} = (E, \mathcal{L})$ is an element $e \in E$ that satisfies $C_e = 0$ for all $C \in \mathcal{L}$. A coloop satisfies $\mathcal{L} \cong \mathcal{L}' \times \{-, 0, +\}$, where $\mathcal{L}'$ is obtained by deleting the e-components from the vectors in $\mathcal{L}$. If $\mathcal{M}$ has a direct sum decomposition with $E_2 = \{e\}$, then e is either a loop or a coloop.

Acyclic oriented matroid: An oriented matroid $\mathcal{M} = (E, \mathcal{L})$ for which $(+, \ldots, +) \in \mathcal{L}$ is a covector; equivalently, the union of the supports of all nonnegative cocircuits is E.

Totally cyclic oriented matroid: An oriented matroid without nonnegative cocircuits; equivalently, $\mathcal{L} \cap \{0, +\}^E = \{\mathbf{0}\}$.

Uniform: An oriented matroid $\mathcal{M}$ of rank d on E is ***uniform*** if all of its cocircuits have size $|E| - d + 1$. Equivalently, $\mathcal{M}$ is uniform if it has a chirotope with values in $\{+, -\}$.

$\mathcal{M}$ is realizable: There is a vector configuration X with $\mathcal{M}_X = \mathcal{M}$.

Realization of $\mathcal{M}$: A vector configuration X with $\mathcal{M}_X = \mathcal{M}$.

THEOREM 6.3.1 *Duality II*

Let $\mathcal{M}$ be an oriented matroid on the ground set E, and $\mathcal{M}^*$ its dual.

- *$\mathcal{M}$ is acyclic if and only if $\mathcal{M}^*$ is totally cyclic. (However, "most" oriented matroids are neither acyclic nor totally cyclic!)*

- *$e \in E$ is a loop of $\mathcal{M}$ if and only if it is a coloop of $\mathcal{M}^*$.*
- *$\mathcal{M}$ is uniform if and only if $\mathcal{M}^*$ is uniform.*
- *$\mathcal{M}$ is a direct sum $\mathcal{M}(E) = \mathcal{M}(E_1) \oplus \mathcal{M}(E_2)$ if and only if $\mathcal{M}^*$ is a direct sum $\mathcal{M}^*(E) = \mathcal{M}^*(E_1) \oplus \mathcal{M}^*(E_2)$.*

Duality of oriented matroids captures, among other things, the concepts of linear programming duality [BK92] [BLS+93, Chapter 10] and the concept of Gale diagrams for polytopes [Zie95, Lect. 6]. For the latter, we note here that the vertex set of a d-dimensional convex polytope P with $d+k$ vertices yields a configuration of $d+k$ vectors in $\mathbb{R}^{d+1}$, and thus an oriented matroid of rank $d+1$ on $d+k$ points. Its dual is a realizable oriented matroid of rank $k-1$, the ***Gale diagram*** of P. It can be modeled by an affine point configuration of dimension $k-2$, called an ***affine Gale diagram*** of P. Hence, for "small" k, we can represent a (possibly high-dimensional) polytope with "few vertices" by a low-dimensional point configuration. In particular, this is beneficial in the case $k = 4$, where polytopes with "universal" behavior can be analyzed in terms of their 2-dimensional affine Gale diagrams. For further details, see Chapter 13 of this Handbook.

6.3.2 REALIZABILITY AND REALIZATION SPACES

GLOSSARY

Realization space: Let $\chi: E^d \to \{-, 0, +\}$ be a chirotope with $\chi(1, \ldots, d) = +$. The *realization space* $\mathcal{R}(\chi)$ is the set of all matrices $X \in \mathbb{R}^{d \cdot n}$ with $\chi_X = \chi$ and $x_i = e_i$ for $i = 1, \ldots, d$, where e_i is the i-th unit vector. If $\mathcal{M}$ is the corresponding oriented matroid, we write $\mathcal{R}(\mathcal{M}) = \mathcal{R}(\chi)$.

Rational realization: A realization $X \in \mathbb{Q}^{d \cdot n}$; that is, a point in $\mathcal{R}(\chi) \cap \mathbb{Q}^{d \cdot n}$.

Basic primary semialgebraic set: The (real) solution set of an arbitrary finite system of polynomial equations and strict inequalities with integer coefficients.

Existential theory of the reals: The problem of solving arbitrary systems of polynomial equations and inequalities with integer coefficients.

Stable equivalence: A strong type of arithmetic and homotopy equivalence. Two semialgebraic sets are stably equivalent if they can be connected by a sequence of rational coordinate changes, together with certain projections with contractible fibers. (See [Ric96] for details.) In particular, two stably equivalent semialgebraic sets have the same number of components, they are homotopy-equivalent, and either both or neither of them have rational points.

It is one of the most exciting problems in oriented matroid theory to design algorithms that find a realization of a given oriented matroid if it exists. However, for oriented matroids with large numbers of points, one cannot be too optimistic, since the realizability problem for oriented matroids is NP-hard. This is one of the consequences of Mnëv's universality theorem below. An upper bound for the worst case complexity of the realizability problem is given by the following theorem. It follows from general complexity bounds for algorithmic problems about

semialgebraic sets by Basu, Pollack, and Roy [BPR97] (see also Chapter 29 of this Handbook).

THEOREM 6.3.2 *Complexity of the Best General Algorithm Known*

The realizability of a rank d oriented matroid on n points can be decided by solving a system of $S = \binom{n}{d}$ real polynomial equations and strict inequalities of degree at most $D = d - 1$ in $K = (n - d - 1)(d - 1)$ variables. Thus, with the algorithms of [BPR97] *the number of bit operations needed to decide realizability is (in the Turing machine model of complexity) bounded by $(S/K)^K \cdot S \cdot D^{O(K)}$.*

THE UNIVERSALITY THEOREM

A basic observation is that all oriented matroids of rank 2 are realizable. In particular, up to change of orientations and permuting the elements in E there is only one uniform oriented matroid of rank 2. The realization space of an oriented matroid of rank 2 is always stably equivalent to $\{0\}$; in particular, if $\mathcal{M}$ is uniform of rank 2 on n elements, then $\mathcal{R}(\mathcal{M})$ is isomorphic to an open subset of $\mathbb{R}^{2n-4}$.

In contrast to the rank 2 case, Mnëv's universality theorem states that for oriented matroids of rank 3, the realization space can be "arbitrarily complicated." Here is the first glimpse of this:

- The realization spaces of all realizable uniform oriented matroids of rank 3 and at most 9 elements are contractible (Richter [Ric89]).
- There is a realizable rank 3 oriented matroid on 9 elements that has no realization with rational coordinates (Perles).
- There is a realizable rank 3 oriented matroid on 14 elements with disconnected realization space (Suvorov).

The universality theorem is a fundamental statement with various implications for the configuration spaces of various types of combinatorial objects.

THEOREM 6.3.3 *Mnëv's Universality Theorem* [Mnë88]

For every basic primary semialgebraic set V defined over $\mathbb{Z}$ there is a chirotope χ of rank 3 such that V and $\mathcal{R}(\chi)$ are stably equivalent.

Although some of the facts in the following list were proved earlier than Mnëv's universality theorem, they all can be considered as consequences of the construction techniques used by Mnëv.

CONSEQUENCES OF THE UNIVERSALITY THEOREM

1. The full field of algebraic numbers is needed to realize all oriented matroids of rank 3.
2. The realizability problem for oriented matroids is NP-hard (Mnëv, Shor).
3. The realizability problem for oriented matroids is (polynomial time) equivalent to the "Existential Theory of the Reals" (Mnëv).
4. For every finite simplicial complex Δ, there is an oriented matroid whose realization space is homotopy-equivalent to Δ.

5. Realizability of rank 3 oriented matroids cannot be characterized by excluding a finite set of "forbidden minors" (Bokowski and Sturmfels).
6. In order to realize all combinatorial types of integral rank 3 oriented matroids on n elements, even uniform ones, in the integer grid $\{1, 2, \ldots, f(n)\}^3$, the "coordinate size" function $f(n)$ has to grow doubly exponentially in n (Goodman, Pollack, and Sturmfels).
7. The ***isotopy problem*** for oriented matroids (Can one given realization of $\mathcal{M}$ be continuously deformed, through realizations, to another given one?) has a negative solution in general, even for uniform oriented matroids of rank 3.

6.3.3 TRIANGLES AND SIMPLICIAL CELLS

There is a long tradition of studying *triangles* in arrangements of pseudolines. In his 1926 paper [Lev26], Levi already considered them to be important structures. There are good reasons for this. On the one hand, they form the simplest possible cells of full dimension, and are therefore of basic interest. On the other hand, if the arrangement is simple, triangles locate the regions where a "smallest" local change of the combinatorial type of the arrangement is possible. Such a change can be performed by taking one side of the triangle and "pushing" it over the vertex formed by the other two sides. It was observed by Ringel [Rin56] that any two simple arrangements of pseudolines can be deformed into one another by performing a sequence of such "triangle flips."

Moreover, the realizability of a pseudoline arrangement may depend on the situation at the triangles. For instance, if any one of the triangles in the nonrealizable example of Figure 6.1.2 other than the central one is flipped, the whole configuration becomes realizable.

TRIANGLES IN ARRANGEMENTS OF PSEUDOLINES

Let $\mathcal{P}$ be any arrangement of n pseudolines.

1. For any pseudoline ℓ in $\mathcal{P}$ there are at least 3 triangles adjacent to ℓ.
   Either the $n-1$ pseudolines different from ℓ intersect in one point (i.e., $\mathcal{P}$ is a ***near-pencil***), or there are at least $n-3$ triangles that are not adjacent to ℓ. Thus $\mathcal{P}$ contains at least n triangles (Levi).
2. $\mathcal{P}$ is ***simplicial*** if all its regions are bounded by exactly 3 (pseudo)lines.
   Except for the near-pencils, there are two infinite classes of simplicial line arrangements and 91 additional "sporadic" simplicial line arrangements (and many more simplicial pseudoarrangements) known (Grünbaum).
3. If $\mathcal{P}$ is simple, then it contains at most $\frac{n(n-1)}{3}$ triangles.
   For infinitely many values of n, there exists a simple arrangement with $\frac{n(n-1)}{3}$ triangles (Roudneff, Harborth).
4. Any two simple arrangements $\mathcal{P}_1$ and $\mathcal{P}_2$ can be deformed into one another by a sequence of simplicial flips (Ringel [Rin56]).

Every arrangement of pseudospheres in S^{d-1} has a centrally symmetric representation. Thus we can always derive an arrangement of projective pseudohyperplanes (pseudo $(d-2)$-planes in $\mathbb{RP}^{d-1}$) by identifying antipodal points. The proper

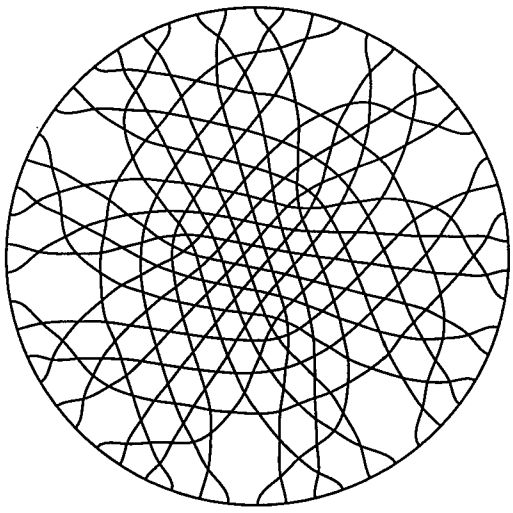


FIGURE 6.3.1
A simple arrangement of 28 *pseudolines with a maximal number of* 252 *triangles.*

analogue for the triangles in rank 3 are the $(d-1)$-simplices in projective arrangements of pseudohyperplanes in rank d, i.e., the regions bounded by the minimal number of d pseudohyperplanes. We call an arrangement ***simple*** if no more than $d-1$ planes meet in a point.

It was conjectured by Las Vergnas in 1980 [L80] that (as in the rank 3 case) any two simple arrangements can be transformed into each other by a sequence of flips of simplicial regions. In particular this requires that every simple arrangement contains *at least one* simplicial region (which was also conjectured by Las Vergnas). If we consider the case of realizable arrangements only, it is not difficult to prove that any two members in this subclass can be connected by a sequence of flips of simplicial regions and that each realizable arrangement contains at least one simplicial cell. In fact, Shannon proved that every arrangement (even the nonsimple) of n projective hyperplanes in rank d contains at least n simplicial regions. More precisely, for every hyperplane h there are at least d simplices adjacent to h and at least $n-d$ simplices not adjacent to h. The contrast between the Las Vergnas conjecture and the results known for the nonrealizable case is dramatic:

SIMPLICIAL CELLS IN PSEUDOARRANGEMENTS

1. There is an arrangement of 8 pseudoplanes in rank 4 having only 7 simplicial regions (Roudneff and Sturmfels, Altshuler and Bokowski).
2. Every rank 4 arrangement with $n < 13$ pseudoplanes has at least one simplicial region (Bokowski).
3. For every $k > 2$ there is a rank 4 arrangement of $4k$ pseudoplanes having only $3k+1$ simplicial regions (Richter-Gebert).
4. There is a rank 4 arrangement consisting of 20 pseudoplanes for which one plane is not adjacent to any simplicial region (Richter-Gebert).

OPEN PROBLEMS

The topic of simplicial cells is interesting and rich in structure even in rank 3. The case of higher dimensions is full of unsolved problems and challenging conjectures. These problems are relevant for various problems of great geometric and topological interest, such as the structure of spaces of triangulations. Three key problems are:

1. Classify simplical arrangements. Is it true, at least, that there are only finitely many types of simplicial arrangements of straight lines outside the three known infinite families?
2. Does every arrangement of pseudohyperplanes contain at least one simplicial region?
3. Is it true that any two simple arrangements can be transformed into one another by a sequence of triangle flips?

6.3.4 MATROID POLYTOPES

The convexity properties of a point configuration X are modeled superbly by the oriented matroid $\mathcal{M}_X$. The combinatorial versions of many theorems concerning convexity also hold on the level of general (including nonrealizable) oriented matroids. For instance, there are purely combinatorial versions of Carathedory's, Radon's, and Helly's theorems [BLS+93, Section 9.2].

In particular, oriented matroid theory provides us with an entirely combinatorial model of convex polytopes, known as "matroid polytopes." The following definition provides this context in terms of face lattices.

Definition: The face lattice of an acyclic oriented matroid $\mathcal{M} = (E, \mathcal{L})$ is the set

$$\mathrm{FL}(\mathcal{M}) := \{C^0 \mid C \in \mathcal{L} \cap \{0,+\}^E\},$$

partially ordered by inclusion. The elements of $\mathrm{FL}(\mathcal{M})$ are the ***faces*** of $\mathcal{M}$. $\mathcal{M}$ is a ***matroid polytope*** if $\{e\}$ is a face for every $e \in E$.

Every polytope gives rise to a matroid polytope: if $P \subseteq \mathbb{R}^d$ is a d-polytope with n vertices, then the canonical embedding $x \mapsto \binom{x}{1}$ creates a vector configuration X_P of rank $d+1$ from the vertex set of P. The oriented matroid of X_P is a matroid polytope $\mathcal{M}_P$, whose face lattice $\mathrm{FL}(\mathcal{M})$ is canonically isomorphic to the face lattice of P.

Matroid polytopes provide a very precise model of (the combinatorial structure of) convex polytopes. In particular, the topological representation theorem implies that *every* matroid polytope of rank d is the face lattice of a regular piecewise linear (PL) cell decomposition of a $(d-2)$-sphere. Thus matroid polytopes form an excellent combinatorial model for convex polytopes: in fact, much better than the model of PL spheres (which does not have an entirely combinatorial definition).

However, the construction of a polar fails in general for matroid polytopes. The cellular spheres that represent matroid polytopes have dual cell decompositions (because they are piecewise linear), but this dual cell decomposition is not in general a matroid polytope, even in rank 4 (Billera and Munson; Bokowski and Schuchert [BSc93]). In other words, the order dual of the face lattice of a matroid

polytope (as an abstract lattice) is *not in general* the face lattice of a matroid polytope. (Matroid polytopes form an important tool for polytope theory, not only because of the parts of polytope theory that work for them, but also because of those that fail.)

For every matroid polytope one has the dual oriented matroid (which is totally cyclic, hence not a matroid polytope). In particular, the set-up for Gale diagrams generalizes to the framework of matroid polytopes; this makes it possible to also include nonpolytopal spheres in a discussion of the realizability properties of polytopes. This amounts to perhaps the most powerful single tool ever developed for polytope theory. It leads to, among other things, the classification of d-dimensional polytopes with at most $d+3$ vertices, the proof that all matroid polytopes of rank $d+1$ with at most $d+3$ vertices are realizable, the construction of nonrational polytopes, as well as of nonpolytopal spheres with $d+4$ vertices, etc.

ALGORITHMIC APPROACH TO POLYTOPE CLASSIFICATION

A powerful approach, via matroid polytopes, to the problem of classifying all convex polytopes with given parameters is largely due to Bokowski and Sturmfels [BSt89]. Here we restrict our attention to the simplicial case—there are additional technical problems to deal with in the nonsimplicial case, and very little work has been done there as yet. However, the program has been successfully completed for the classification of all simplicial 3-spheres with 9 vertices (Altshuler, Bokowski, and Steinberg) and of all neighborly 5-spheres with 10 vertices (Bokowski and Shemer) into polytopes and nonpolytopes. At the core of the matroidal approach lies the following hierarchy:

$$\begin{pmatrix}\text{simplicial}\\ \text{spheres}\end{pmatrix} \longrightarrow \begin{pmatrix}\text{uniform}\\ \text{matroid polytopes}\end{pmatrix} \longrightarrow \begin{pmatrix}\text{convex}\\ \text{polytopes}\end{pmatrix}.$$

The plan of attack is the following. First, one enumerates all isomorphism types of simplicial spheres with given parameters. Then, for each sphere, one computes *all* (uniform) matroid polytopes that have the given sphere as their face lattices. Finally, for each matroid polytope, one tries to decide realizability.

At both of the steps of this hierarchy there are considerable subtleties involved that lead to important insights. For a given simplicial sphere, there may be

- *no* matroid polytope that supports it. In this case the sphere is called ***non-matroidal***. The Barnette sphere [BLS+93, Proposition 9.5.3] is an example.
- *exactly one* matroid polytope. In this (important) case the sphere is called ***rigid***. That is, a matroid polytope $\mathcal{M}$ is rigid if $\mathrm{FL}(\mathcal{M}') = \mathrm{FL}(\mathcal{M})$ already implies $\mathcal{M}' = \mathcal{M}$. For rigid matroid polytopes the face lattice uniquely defines the oriented matroid, and thus every statement about the matroid polytope yields a statement about the sphere. In particular, the matroid polytope and the sphere have the same realization space.

  Rigid matroid polytopes are a priori rare; however, the *Lawrence construction* [BLS+93, Section 9.3] [Zie95, Section 6.6] associates with every oriented matroid $\mathcal{M}$ on n elements in rank d a rigid matroid polytope $\Lambda(\mathcal{M})$ with $2n$ vertices of rank $n+d$. The realizations of $\Lambda(\mathcal{M})$ can be retranslated into realizations of $\mathcal{M}$.
- or *many* matroid polytopes.

The situation is similarly complex for the second step, from matroid polytopes to convex polytopes. In fact, for each matroid polytope there may be

- *no* convex polytope—this is the case for a nonrealizable matroid polytope. These exist already with relatively few vertices; namely in rank 5 with 9 vertices [BSc93], and in rank 4 with 10 vertices [BLS+93, Proposition 9.4.5].
- essentially *only one*—this is the rare case where the matroid polytope is "projectively unique."
- or *many* convex polytopes—the space of all polytopes for a given matroid polytope is the realization space of the oriented matroid, and this may be arbitrarily complicated. In fact, a combination of Mnëv's universality theorem, the Lawrence construction, and a scattering technique [BSt89, Thm 6.2] (in order to obtain the simplicial case) yields the following amazing universality theorem.

THEOREM 6.3.4 *Mnëv's Universality Theorem for Polytopes* [Mnë88]
For every [open] basic primary semialgebraic set V defined over $\mathbb{Z}$ there is an integer d and a [simplicial] d-dimensional polytope P on $d+4$ vertices such that V and the realization space of P are stably equivalent.

6.4 SOURCES AND RELATED MATERIAL

FURTHER READING

The basic theory of oriented matroids was introduced in two fundamental papers, Bland and Las Vergnas [BL78] and Folkman and Lawrence [FL78]. We refer to the monograph by Björner, Las Vergnas, Sturmfels, White, and Ziegler [BLS+93] for a broad introduction, and for an extensive development of the theory of oriented matroids. An up-to-date bibliography is given in [Zie96+]. Other introductions and basic sources of information include Bachem and Kern [BK92], Bokowski [Bok93], Bokowski and Sturmfels [BSt89], and Ziegler [Zie95, Lect. 6 and 7].

RELATED CHAPTERS

Chapter 5: Pseudoline arrangements
Chapter 13: Basic properties of convex polytopes
Chapter 21: Arrangements
Chapter 29: Computational real algebraic geometry
Chapter 39: Mathematical programming
Chapter 48: Geometric applications of the Cayley-Grassmann algebra

REFERENCES

[BK92] A. Bachem and W. Kern. *Linear Programming Duality: An Introduction to Oriented Matroids. Universitext.* Springer-Verlag, Berlin/Heidelberg, 1992.

[BPR97] S. Basu, R. Pollack, and M.-F. Roy. A new algorithm to find a point in every cell defined by a family of polynomials. In B. Caviness and J. Johnson, editors, *Quantifier Elimination and Cylindric Algebraic Decomposition*, pages 341–349. *Texts Monographs Symbol. Comput.*. Springer-Verlag, Vienna/New York, 1997.

[BLS+93] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G.M. Ziegler. *Oriented Matroids*. Volume 46 of *Encyclopedia Math. Appl.*. Cambridge University Press, 1993.

[BL78] R.G. Bland and M. Las Vergnas. Orientability of matroids. *J. Combin. Theory Ser. B*, 24:94–123, 1978.

[Bok93] J. Bokowski. Oriented matroids. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 555–602. North-Holland, Amsterdam, 1993.

[BSc93] J. Bokowski and P. Schuchert. Altshuler's sphere M^9_{963} revisited. Preprint, 1993.

[BSt89] J. Bokowski and B. Sturmfels. *Computational Synthetic Geometry*. Volume 1355 of *Lecture Notes in Math.* Springer-Verlag, Berlin/Heidelberg, 1989.

[EM82] J. Edmonds and A. Mandel. *Topology of Oriented Matroids*. Ph.D. thesis of A. Mandel, Univ. of Waterloo, 1982.

[FL78] J. Folkman and J. Lawrence. Oriented matroids. *J. Combin. Theory Ser. B*, 25:199–236, 1978.

[Grü72] B. Grünbaum. *Arrangements and Spreads*. Volume 10 of *CBMS Regional Conf. Ser. in Math.* Amer. Math. Soc., Providence, 1972.

[Knu92] D.E. Knuth. *Axioms and Hulls*. Volume 606 of *Lecture Notes in Comput. Sci.* Springer-Verlag, Berlin/Heidelberg, 1992.

[L80] M. Las Vergnas. Convexity in oriented matroids. *J. Combin. Theory Ser. B*, 29:231–243, 1980.

[Lev26] F. Levi. Die Teilung der projektiven Ebene durch Gerade oder Pseudogerade. *Ber. Math.-Phys. Kl. Sächs. Akad. Wiss.*, 78:256–267, 1926.

[Mnë88] N.E. Mnëv. The universality theorems on the classification problem of configuration varieties and convex polytopes varieties. In O.Ya. Viro, editor, *Topology and Geometry—Rohlin Seminar*, pages 527–544, volume 1346 of *Lecture Notes in Math.* Springer-Verlag, Berlin/Heidelberg, 1988.

[Oxl92] J. Oxley. *Matroid Theory*. Oxford University Press, Oxford, 1992.

[Ric89] J. Richter. Kombinatorische Realisierbarkeitskriterien für orientierte Matroide. *Mitt. Math. Sem. Gießen*, 194:1–112, 1989.

[Ric96] J. Richter-Gebert. *Realization Spaces of Polytopes*. Volume 1643 of *Lecture Notes in Math.* Springer-Verlag, Berlin/Heidelberg, 1996.

[Rin56] G. Ringel. Teilungen der Ebene durch Geraden oder topologische Geraden. *Math. Z.*, 64:79–102, 1956.

[Zie95] G.M. Ziegler. *Lectures on Polytopes*. Volume 152 of *Graduate Texts in Math.*, Springer-Verlag, New York, 1995.
[*Updates, corrections, etc.* available at `http://www.math.tu-berlin.de/~ziegler`.]

[Zie96+] G.M. Ziegler. Oriented matroids today: Dynamic survey and updated bibliography. *Electron. J. Combin.*, 3:DS#4, 1996+.

7 LATTICE POINTS AND LATTICE POLYTOPES

Alexander Barvinok

INTRODUCTION

Lattice polytopes arise naturally in number theory, algebraic geometry, optimization, combinatorics, and analysis. They possess a very rich structure arising from the interaction of algebraic, convex, and combinatorial properties. In this chapter we concentrate on the theory of lattice polytopes and only sketch their numerous applications. We briefly discuss their role in optimization and polyhedral combinatorics (Section 7.1). In Section 7.2 we discuss the *decision problem*, the problem of finding whether a given polytope contains a lattice point. In Section 7.3 we address the *counting problem*, the problem of counting all lattice points in a given polytope. The *asymptotic problem* (Section 7.4) explores the behavior of the number of lattice points in a varying polytope (for example, if a dilatation is applied to the polytope). Finally, in Section 7.5 we discuss *problems with quantifiers*. These problems are natural generalizations of the decision problem, which may be interpreted as a problem with existential quantifiers only. Whenever appropriate we address algorithmic issues. For general references in the area of computational complexity/algorithms see [AHU74]. We summarize the computational complexity status of our problems in Table 7.0.1.

TABLE 7.0.1 Computational complexity of basic problems.

PROBLEM NAME	BOUNDED DIMENSION	UNBOUNDED DIMENSION
Decision problem	polynomial	NP-hard
Counting problem	polynomial	#P-hard
Asymptotic problem	polynomial	#P-hard*
Problems with quantifiers	unknown; polynomial for $\forall\exists$	NP-hard

* in bounded codimension this reduces polynomially to volume computation

7.1 INTEGRAL POLYTOPES IN POLYHEDRAL COMBINATORICS

We describe some combinatorial and computational properties of integral polytopes. General references are [GK94], [GLS88], [GW93], [Sch86], and [Lag95].

GLOSSARY

$\mathbb{R}^d$: Euclidean d-dimensional space with scalar product $\langle x, y\rangle = x_1y_1 + \ldots + x_dy_d$, where $x = (x_1, \ldots, x_d)$ and $y = (y_1, \ldots, y_d)$.

$\mathbb{Z}^d$**:** The subset of $\mathbb{R}^d$ consisting of the points with integral coordinates.

Polytope: The convex hull of finitely many points in $\mathbb{R}^d$.

Face of a polytope P: The intersection of P and the boundary hyperplane of a halfspace containing P.

Facet: A face of codimension 1.

Vertex: A face of dimension 0; the set of vertices of P is denoted by Vert P.

$\mathcal{H}$-description of a polytope ($\mathcal{H}$-polytope): A representation of the polytope as the set of solutions of finitely many linear inequalities.

$\mathcal{V}$-description of a polytope ($\mathcal{V}$-polytope): The representation of the polytope by the set of its vertices.

Integral polytope or ***lattice polytope:*** A polytope with all of its vertices in $\mathbb{Z}^d$.

$(0,1)$-polytope: A polytope P such that each coordinate of any vertex of P is either 0 or 1.

An integral polytope $P \subset \mathbb{R}^d$ can be given either by its $\mathcal{H}$-description or by its $\mathcal{V}$-description or (somewhat implicitly) as the convex hull of integral points in some other polytope Q: $P = \text{conv}\{Q \cap \mathbb{Z}^d\}$. In most cases it is difficult to translate one description into another. The following examples illustrate some typical kinds of behavior.

INTEGRALITY OF $\mathcal{H}$-POLYTOPES

It is an NP-hard problem to decide whether an $\mathcal{H}$-polytope $P \subset \mathbb{R}^d$ is integral. However, if the dimension d is fixed then the straightforward procedure of generating all the vertices of P and checking their integrality has polynomial time complexity. A rare case where an $\mathcal{H}$-polytope P is a priori integral is known under the general name of "total unimodularity." Let A be an $n \times d$ integral matrix such that every minor of A is either 0 or 1 or -1. Such a matrix A is called ***totally unimodular***. If $b \in \mathbb{Z}^n$ is an integral vector then the set of solutions to the system of linear inequalities $Ax \leq b$ is an integral polytope in $\mathbb{R}^d$, provided this set is bounded. Examples of totally unimodular matrices include matrices of vertex-edge incidences of oriented graphs and of bipartite graphs. A complete characterization of totally unimodular matrices and a polynomial time algorithm for recognizing a totally unimodular matrix is provided by a theorem of P. Seymour (see [Sch86]). A family of integral polytopes, called ***transportation polytopes***, were intensively studied in the literature (see [EKK84]). The most famous of them is the ***Birkhoff polytope*** B_n, which is given in the space of $n \times n$ matrices $x = (x_{ij})$ by the system of equations $\sum_{i=1}^n x_{ij} = 1$, $j = 1, \ldots, n$, $\sum_{j=1}^n x_{ij} = 1$, $i = 1, \ldots, n$ and inequalities $x_{ij} \geq 0$, and alternatively may be described as the convex hull of the $n!$ permutation matrices $\pi(\sigma)_{ij} = \delta_{j\sigma(j)}$ for all permutations σ of the set $\{1, \ldots, n\}$.

The notion of total unimodularity has been generalized in various directions, thus leading to new classes of integral polytopes (see [CCK+94]).

$\mathcal{V}$-POLYTOPES WITH MANY VERTICES

There are several important situations where the explicit $\mathcal{V}$-description of an integral polytope is too long and a shorter description is desirable although not always

available. For example, a $(0,1)$-polytope may be given as the convex hull of the characteristic vectors

$$\chi_S(i) = \begin{cases} 1 & \text{if } i \in S, \\ 0 & \text{otherwise} \end{cases}$$

for some combinatorially interesting family $\mathcal{S}$ of subsets $S \subset \{1, \ldots, d\}$ (see [GLS88] for various examples). The most famous example is the ***traveling salesman polytope***, the convex hull TSP_n of the $(n-1)!$ permutation matrices $\pi(\sigma)$ where σ is a permutation of the set $\{1, \ldots, n\}$ containing precisely one cycle (cf. the Birkhoff polytope B_n above). The problem of the $\mathcal{H}$-description of the traveling salesman polytope has attracted a lot of attention (see [GW93] and [EKK84] for some references) because of its relevance to combinatorial optimization. As negative results in this direction we mention that it is an NP-complete problem to establish whether two given vertices of TSP_n are adjacent, i.e., connected by an edge. L. Billera and A. Sarangarajan proved that every $(0,1)$-polytope can be realized as a face of TSP_n for sufficiently large n (see [BS96]). Thus the combinatorics of TSP_n contrasts with the combinatorics of the Birkhoff polytope B_n. On the other hand, the author has shown recently that the support functions $h_P(c) = \max\{\langle c, x\rangle \mid x \in P\}$ for $P = \mathrm{TSP}_n$ and $P = B_n$ "almost coincide" as $n \longrightarrow \infty$.

Another important polytope arising in this way is the ***cut polytope***, the famous counterexample to the Borsuk conjecture (see [Zie94]).

CONVEX HULL OF INTEGRAL POINTS

Let $P \subset \mathbb{R}^d$ be a polytope. Then the convex hull P_I of the set $P \cap \mathbb{Z}^d$, if nonempty, is an integral polytope. Generally, the number of facets or vertices of P_I depends not only on the number of facets or vertices of P but also on the actual numerical size of the description of P (see [CHKM92]). Furthermore, it is an NP-complete problem to check whether a given point belongs to P_I, where P is given by its $\mathcal{H}$-description. If, however, the dimension d is fixed then the complexity of the facial description of the polytope P_I is polynomial in the complexity of the description of P. In particular, the number of vertices of P_I is bounded by a polynomial of degree $d-1$ in the input size of P (see [CHKM92]).

Integrality imposes some restrictions on the combinatorial structure of a polytope. It is known that the combinatorial type of any 2- or 3-dimensional polytope can be realized by an integral polytope. On the other hand, there are examples of higher-dimensional polytopes whose combinatorial type can not be realized by an integral polytope [Zie94]. Existence of a 4-dimensional polytope with a nonintegral (and, therefore, nonrational) combinatorial type was recently reported by J. Richter-Gebert; the smallest dimension in which an example was previously known was 8. On the other hand, it was known that for sufficiently large d there exist nonrational d-polytopes with $d+4$ vertices. The number $N_d(V)$ of classes of integral d-polytopes having volume V and nonisomorphic with respect to affine transformations of $\mathbb{R}^d$ preserving the integral lattice $\mathbb{Z}^d$ has logarithmic order [BV92]

$$c_1(d) V^{\frac{d-1}{d+1}} \leq \log N_d(V) \leq c_2(d) V^{\frac{d-1}{d+1}},$$

for some nonzero constants $c_1(d), c_2(d)$.

7.2 DECISION PROBLEM

We consider the following general decision problem: Given a polytope $P \subset \mathbb{R}^d$ and a lattice $\Lambda \subset \mathbb{R}^d$, decide whether $P \cap \Lambda = \emptyset$ and, if the intersection is nonempty, find a point in $P \cap \Lambda$. We describe the main structural and algorithmic results for this problem. General references are [GL87], [GLS88], [GW93], [Sch86], and [Lag95].

GLOSSARY

Lattice: A discrete additive subgroup Λ of $\mathbb{R}^d$, i.e., $x - y \in \Lambda$ for any $x, y \in \Lambda$ and Λ does not contain limit points.

Basis of a lattice: A set of linearly independent vectors $u_1, \ldots, u_k$ such that every vector $y \in \Lambda$ can be (uniquely) represented in the form $y = m_1 u_1 + \ldots + m_k u_k$ for some integers $m_1, \ldots, m_k$.

Rank of a lattice: The cardinality of any basis of the lattice. If $\Lambda \subset \mathbb{R}^d$ has rank d, Λ is said to be of ***full rank.***

Determinant of a lattice: For a lattice of rank k the k-volume of the parallelopiped spanned by any basis of the lattice.

Polyhedron: An intersection of finitely many halfspaces in $\mathbb{R}^d$.

Convex body: A compact convex set in $\mathbb{R}^d$ with nonempty interior.

Applying a suitable linear transformation one can reduce the decision problem to the particular case in which $\Lambda = \mathbb{Z}^k$ and $P \subset \mathbb{R}^k$ is a full-dimensional polytope, $k = \text{rank } \Lambda$.

The decision problem is known to be NP-complete for $\mathcal{H}$-polytopes as well as for $\mathcal{V}$-polytopes, although some special cases admit a polynomial time algorithm. In particular, if one fixes the dimension d then the decision problem becomes polynomially solvable. The main tool is provided by the so-called "flatness results."

FLATNESS THEOREM

THEOREM 7.2.1

There exists a function $f : \mathbb{N} \to \mathbb{R}$ such that for every convex body $P \subset \mathbb{R}^d$ such that $P \cap \mathbb{Z}^d = \emptyset$ there exists a nonzero vector $l \in \mathbb{Z}^d$ for which

$$\max\{\langle l, x\rangle \mid x \in P\} - \min\{\langle l, x\rangle \mid x \in P\} \leq f(d).$$

There are two types of results relating to the flatness theorem. First, one may be interested in making $f(d)$ as small as possible. It is known that one can choose $f(d) = O(d^2)$ [KL88] and it is conjectured that one can choose $f(d)$ as small as $O(d)$. If P is centrally symmetric it is proven that one can choose $f(d) = O(d \log d)$ [Ban95]. Second, one may be interested in the smallest possible $f(d)$ for which the actual vector $l \in \mathbb{Z}^d$ can be computed in polynomial time. The best bound known for such f is $f(d) = 2^{O(d)}$, where the corresponding vector l is polynomially computable even if the dimension d varies. For some small values of d the best

possible bound for d may be found. For example, one can choose $f(3) = 1$ (see [Sca85]).

ALGORITHMS FOR THE DECISION PROBLEMS

The flatness theorem allows one to reduce the dimension in the decision problem: Assuming that the body P does not contain an integral point, one constructs a vector $l \in \mathbb{Z}^d$ by Theorem 7.2.1 and reduces the d-dimensional decision problem to at most $f(d) + 1$ instances of the $(d-1)$-dimensional decision problem $P_i = \{x \in P \mid \langle l, x\rangle = i\}$, where i ranges between $\min\{\langle l, x\rangle \mid x \in P\}$ and $\max\{\langle l, x\rangle \mid x \in P\}$. This reduction is the main idea of polynomial time algorithms in fixed dimension. The best complexity known for the decision problem in terms of the dimension d is $d^{O(d)}$.

MINKOWSKI'S CONVEX BODY THEOREM

The following classical result, known as "Minkowski's convex body theorem," provides a very useful criterion.

THEOREM 7.2.2

Suppose $B \subset \mathbb{R}^d$ is a convex body, centrally symmetric about the origin 0, and $\Lambda \subset \mathbb{R}^d$ is a lattice of full rank. If $\operatorname{vol} B \geq 2^d \det \Lambda$ *then B contains a nonzero point of Λ.*

For the proof and various generalizations see, for example, [GL87]. An important generalization concerns the existence of i linearly independent lattice points in a convex body. Namely, if $\lambda_i = \inf\{\lambda > 0 \mid \lambda B \cap \Lambda$ contains i linearly independent points$\}$ is the "ith successive minimum," then $\lambda_1 \ldots \lambda_d \leq (2^d \det \Lambda)/(\operatorname{vol} B)$.

If B is a convex body such that $\operatorname{vol} B = 2^d \det \Lambda$ but B does not contain a nonzero lattice point in its interior, then B is called ***extremal***. Every extremal body is necessarily a polytope. Moreover, this polytope contains at most $2(2^d - 1)$ facets, and therefore, for every dimension d, there exist only finitely many combinatorially different extremal polytopes. The contracted polytope $P = \{x/2 \mid x \in B\}$ has the property that its lattice translates $P + x \mid x \in \Lambda$ tile the space $\mathbb{R}^d$. Such a tiling polytope is called a ***parallelohedron***. Similarly, for every dimension d there exist only finitely many combinatorially different parallelohedra. Parallelohedra can be characterized intrinsically: a polytope is a parallelohedron if and only if it is centrally symmetric, every facet of it is centrally symmetric, and every class of parallel ridges ($(d-2)$-dimensional facets) consists of four or six ridges. If $q : \mathbb{R}^d \longrightarrow \mathbb{R}$ is a positive definite quadratic form, then the ***Dirichlet-Voronoi cell*** $P_q = \{x \mid q(x) \leq q(x - \lambda)$ for any $\lambda \in \Lambda\}$ is a parallelohedron. The problem of finding whether a centrally symmetric polyhedron P contains a nonzero lattice point is known to be NP-complete even in the case of the standard cube $P = \{(x_1, \ldots, x_d) \mid -1 \leq x_i \leq 1\}$. For fixed dimension d there exists a polynomial time algorithm since the problem obviously reduces to the decision problem (one can add the extra inequality $x_1 + \ldots + x_d \geq 1$).

VOLUME BOUNDS

An integral simplex in $\mathbb{R}^d$ containing no lattice points other than its vertices has volume $1/2$ if $d = 2$ but already for $d = 3$ can have an arbitrarily large volume (the smallest possible volume of such a simplex is $1/d!$). On the other hand, if an integral polytope P contains precisely $k > 0$ integral points then its volume is bounded by a function of k and d. The best bound known, vol $P \leq k(7(k+1))^{2^{d+1}}$, is due to J. Lagarias and G.M. Ziegler (see [Lag95]).

7.3 COUNTING PROBLEM

We consider the following problem: Given a polytope $P \subset \mathbb{R}^d$ and a lattice $\Lambda \subset \mathbb{R}^d$, compute the number of lattice points $\#(P \cap \Lambda)$ in P.

For counting in general convex bodies see [CHKM92]. For some applications in the combinatorics of generating functions and representation theory see, for example, [BZ88] and [Sta86]. For general information see the surveys [GK94] and [GW93].

GLOSSARY

Edge of a polytope: A face of dimension 1.

Polyhedral cone: A set $K \subset \mathbb{R}^d$ of the form $K = \left\{\sum_{i=1}^k \lambda_i u_i \mid \lambda_i \geq 0,\ i = 1, \ldots, k\right\}$ for some vectors $u_1, \ldots, u_k \in \mathbb{R}^d$. The vectors $u_1, \ldots u_k$ are called ***generators*** of K.

Cone of feasible directions at a point: The cone

$$K_v = \left\{x \mid v + \epsilon x \in P \text{ for all sufficiently small } \epsilon > 0\right\}$$

for a point v of a polytope P. If v is a vertex, then the cone K_v is generated by the vectors $u_i = v_i - v$, where $[v_i, v]$ is an edge of P.

Rational cone: A polyhedral cone having a set of generators belonging to $\mathbb{Z}^d$.

Simple cone: A polyhedral cone generated by linearly independent vectors.

Fundamental parallelopiped of a simple cone: The set

$$\Pi = \left\{\lambda_1 u_1 + \ldots + \lambda_k u_k \mid 0 \leq \lambda_i < 1,\ i = 1, \ldots, k\right\},$$

where $u_1, \ldots, u_k$ are linearly independent generators of the cone.

Unimodular cone: A rational simple cone $K \subset \mathbb{R}^d$ whose fundamental parallelopiped does not contain points of $\mathbb{Z}^d$ other than 0.

Simple polytope: A polytope P such that the cone K_v of feasible directions is simple for every vertex v of P.

Totally unimodular polytope: An integral polytope P such that the cone K_v of feasible directions is unimodular for any vertex v of P.

Changing the coordinates, one can always assume that $\Lambda = \mathbb{Z}^d$ and that the polytope P is full-dimensional.

GENERAL INFORMATION

The counting problem is known to be $\#P$-hard even for an integral $\mathcal{H}$- or $\mathcal{V}$-polytope. Moreover, for an $\mathcal{H}$-polytope $P \subset \mathbb{R}^d$, it is an NP-hard problem to compute the number of integral points in P even within the factor $\exp\{p(d)\}$, where $p(d)$ is a polynomial.

However, if the dimension d is fixed, one can solve the counting problem in polynomial time [Bar94b].

SOME EXPLICIT FORMULAS IN LOW DIMENSIONS

The classical Pick formula expresses the number of integral points in a convex integral polygon $P \subset \mathbb{R}^2$ in terms of its area and the number of integral points on the boundary ∂P:

$$\#(P \cap \mathbb{Z}^2) = \text{area}(P) + \frac{1}{2} \cdot \#(\partial P \cap \mathbb{Z}^2) + 1$$

(see, for example, [Mor93a], [GW93]). This formula almost immediately gives rise to a polynomial time algorithm for counting integral points in integral polygons.

An important explicit formula for the number of integral points in a lattice tetrahedron of a special kind was proven by L. Mordell (see [Pom93]). Let a, b, c be pairwise coprime positive integers and $\Delta(a, b, c) \subset \mathbb{R}^3$ be the tetrahedron with vertices $(0, 0, 0)$, $(a, 0, 0)$, $(0, b, 0)$, and $(0, 0, c)$. Then

$$\#\Big(\Delta(a, b, c) \cap \mathbb{Z}^3\Big) = \frac{abc}{6} + \frac{ab + ac + bc + a + b + c}{4} +$$

$$\frac{1}{12}\Big(\frac{ac}{b} + \frac{bc}{a} + \frac{ab}{c} + \frac{1}{abc}\Big) - s(bc, a) - s(ac, b) - s(ab, c) + 2. \qquad (7.3.1)$$

Here

$$s(p, q) = \sum_{i=1}^{q} \Big(\Big(\frac{i}{q}\Big)\Big)\Big(\Big(\frac{pi}{q}\Big)\Big), \qquad \text{where} \qquad ((x)) = x - 0.5(\lfloor x \rfloor + \lceil x \rceil),$$

is the Dedekind sum. A similar formula was found in dimension 4 (see [Pom93]). The famous reciprocity relation $s(p, q) + s(q, p) = (p/q + q/p + 1/pq - 3)/12$ allows one to compute the Dedekind sum $s(p, q)$ in polynomial time (see [Dye91]). A version of formula (7.3.1) was used in [Dye91], where polynomial time algorithms for the counting problem for $d = 3, 4$ are constructed. Formula (7.3.1) was generalized to an arbitrary tetrahedron in [Pom93]. Later a generalization to higher dimensions was suggested in [CS94].

A powerful tool for solving the counting problem is provided by ***exponential sums***, which may be regarded as generating functions for sets of integral points.

EXPONENTIAL SUMS

Let $P \subset \mathbb{R}^d$ be a polytope and $c \in \mathbb{R}^d$ be a vector. We consider the exponential sum

$\sum_{x \in P \cap \mathbb{Z}^d} \exp\{\langle c, x \rangle\}$. If $c = 0$ we get the number of integral points in P. The reason for introducing the parameter c is that for a "generic" c the exponential sums reveal some nontrivial algebraic properties that become less visible when $c = 0$. To describe these properties we need to consider exponential sums over polyhedral cones.

EXPONENTIAL SUMS OVER RATIONAL CONES

Let $K \subset \mathbb{R}^d$ be a rational cone without straight lines generated by vectors $u_1, \ldots, u_k$ in $\mathbb{Z}^d$. Then the series $\sum_{x \in K \cap \mathbb{Z}^d} \exp\{\langle c, x \rangle\}$ converges for any c such that $\langle c, u_i \rangle < 0$ for all $i = 1, \ldots, k$ and defines a meromorphic function of c which we denote by $\sigma_K(c)$. For a simple rational cone $K \subset \mathbb{R}^d$ with linearly independent generators $u_1, \ldots, u_k$ we have

$$\sigma_K(c) = \left(\sum_{x \in \Pi \cap \mathbb{Z}^d} \exp\{\langle c, x \rangle\} \right) \cdot \prod_{i=1}^{k} \frac{1}{1 - \exp\{\langle c, u_i \rangle\}},$$

where Π is the fundamental parallelopiped of K. In particular, if K is a unimodular then

$$\sigma_K(c) = \prod_{i=1}^{d} \frac{1}{1 - \exp\{\langle c, u_i \rangle\}},$$

since the corresponding sum is just the multiple geometric series. Generally speaking, the farther a given cone is from being unimodular, the more complicated the formula for $\sigma_K(c)$ will be.

These results are known in many different forms (see, for example, Section 4.6 of [Sta86]). Furthermore, the function $\sigma_K(c)$ can be extended to a finitely additive measure, defined on rational polyhedra in $\mathbb{R}^d$ and taking its values in the space of meromorphic functions in d variables, so that the measure of a rational polyhedron with a straight line is equal to 0 [Law91].

The following crucial theorem relates the exponential sum over an integral polytope and the exponential sums over cones K_v of feasible directions for the vertices v of P.

THEOREM 7.3.1 *Brion's Theorem*

Let $P \subset \mathbb{R}^d$ be an integral polytope. Then

$$\sum_{x \in P \cap \mathbb{Z}^d} \exp\{\langle c, x \rangle\} = \sum_{v \in \mathrm{Vert}\, P} \exp\{\langle c, v \rangle\} \cdot \sigma_{K_v}(c).$$

For example, if $d = 1$ and $P = [0, n]$ is an interval, we get the formula

$$\sum_{x=0}^{n} \exp\{cx\} = \exp\{c \cdot 0\} \cdot \sum_{x=0}^{+\infty} \exp\{cx\} + \exp\{cn\} \cdot \sum_{x=-\infty}^{0} \exp\{cx\} = \frac{\exp\{c(n+1)\} - 1}{\exp\{c\} - 1}.$$

For different proofs see [Bri92], [Law91].

Brion's formula allows one to reduce the counting of integral points in polytopes to the counting of points in polyhedral cones, a much easier problem. Below

we discuss two instances where the application of exponential sums and Brion's identities leads to an efficient computational solution of the counting problem.

COUNTING IN FIXED DIMENSION

The following result is proven in [Bar94b].

THEOREM 7.3.2

Let us fix the dimension d. Then there exists a polynomial time algorithm that, for any given rational polytope $P \subset \mathbb{R}^d$, computes the number $\#(P \cap \mathbb{Z}^d)$ of integral points in P.

THE IDEA OF THE ALGORITHM

We assume that the polytope is given by its $\mathcal{V}$-description. Furthermore, we may assume that the polytope is integral since the convex hull P_I of integral points of P (see Section 7.1) can be computed in polynomial time provided the dimension is fixed. Let us choose a "generic" $c \in \mathbb{Q}^d$. We can compute the number $\#(P \cap \mathbb{Z}^d)$ as the limit of the exponential sum

$$\lim_{t \longrightarrow 0} \sum_{x \in P \cap \mathbb{Z}^d} \exp\{\langle tc, x \rangle\},$$

where t is a real parameter. Using Brion's Theorem 7.3.1, we reduce the problem to the computation of the constant term in the Laurent expansion of the meromorphic function $f_v(t) = \exp\{\langle tc, v \rangle\} \cdot \sigma_{K_v}(tc)$, where v is a vertex of P and K_v is the cone of feasible directions at v. If K_v is a unimodular cone, we have an explicit formula for $\sigma_{K_v}(c)$ (see above) and thus can easily compute the desired term. However, for $d > 1$ the cone K_v does not have to be unimodular. It turns out, nevertheless, that for any given rational cone K one can construct in polynomial time a decomposition $K = \sum_{i \in I} \epsilon_i K_i$, $\epsilon_i \in \{-1, 1\}$, of the "inclusion-exclusion" type, where the cones K_i are unimodular (see below). Thus one can get an explicit expression $\sigma_{K_v}(c) = \sum_{i \in I} \epsilon_i \cdot \sigma_{K_i}(c)$ and then compute the constant term of the Laurent expansion of $f_v(t)$.

The fastest known algorithm for counting integral points in an integral simplex is described in [Bar94a]. Its complexity in terms of the dimension d is $d^{O(d)}$.

COUNTING IN TOTALLY UNIMODULAR POLYTOPES

One can efficiently count the number of integral points in a totally unimodular polytope given by its vertex description even in varying dimension.

THEOREM 7.3.3

There exists an algorithm that, for any d and any given integral vertices $v_1, \ldots, v_m \in \mathbb{Z}^d$ such that the polytope $P = \mathrm{conv}\{v_1, \ldots, v_m\}$ is totally unimodular, computes the number of integral points of P in time linear in the number m of vertices.

The algorithm uses Brion's formulas (Theorem 7.3.1) and the explicit formula above for the exponential sum over a unimodular cone.

EXAMPLE

Suppose A is an $n \times d$ totally unimodular matrix (see Section 7.1). Let us choose $b \in \mathbb{Z}^n$ such that the set P_b of solutions to the system $Ax \leq b$ of linear inequalities is a simple polytope. Then P_b is totally unimodular.

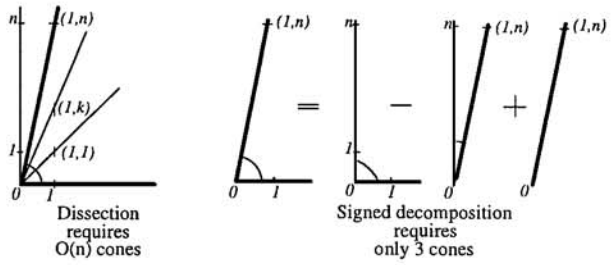
For example, if we know all the vertices of a simple transportation polytope P, we can compute the number of integral points of P in time linear in the number of vertices of P.

We discuss totally unimodular polytopes also in Section 7.4.

One can construct an efficient algorithm for counting integral points in a polytope that is somewhat "close" to totally unimodular and for which the explicit formulas for $\sigma_{K_v}(c)$ are therefore not too long.

CONNECTIONS WITH TORIC VARIETIES

It was first observed by A. Khovanskii in the 1970's, and has since then become widely known, that the number of integral points in an integral polytope is related to some algebro-geometric invariants of the associated toric variety (see [Oda88]). Naturally, for smooth toric varieties (they correspond to totally unimodular polytopes) computation is much easier. Various formulas for the number of integral points in polytopes were first obtained for totally unimodular polytopes and then, by the use of resolution of singularities, generalized to arbitrary integral polytopes (see, for example, [Pom93]). Resolution of singularities of toric varieties reduces to dissection of a polyhedral cone into unimodular cones. However, as one can see, it is impossible to subdivide a rational cone into polynomially (in the input) many unimodular cones even in dimension $d = 2$. For example (see Figure 7.3.1), the plane cone K generated by the points $(1, 0)$ and $(1, n)$ cannot be subdivided into fewer than $2n - 1$ unimodular cones, whereas a polynomial time subdivision would give a polynomial in $\log n$ cones. On the other hand, if we allow a signed linear combination of the inclusion-exclusion type, then one can easily represent this cone as a combination of 3 primitive cones: $K = K_1 - K_2 + K_3$, where K_1 is generated by the basis $(1, 0)$ and $(0, 1)$, K_2 is generated by $(0, 1)$ and $(1, n)$, and K_3 is generated by $(1, n)$. As we have mentioned above, once we allow "signed" combinations, any rational polyhedral cone can be decomposed into unimodular cones in polynomial time, provided the dimension is fixed [Bar94b].



Signed decomposition
requires
only 3 cones

FIGURE 7.3.1
Decomposition of a cone into unimodular cones.

CONNECTIONS WITH VALUATIONS

The number of integral points $\nu(P) = \#(P \cap \mathbb{Z}^d)$ in an integral polytope $P \subset \mathbb{R}^d$ is a ***valuation***, i.e., it satisfies the inclusion-exclusion principle $\nu(P_1 \cup P_2) = \nu(P_1) + \nu(P_2) - \nu(P_1 \cap P_2)$; and it is lattice-translation-invariant, i.e., $\nu(P + l) = \nu(P)$ for any $l \in \mathbb{Z}^d$. General properties of valuations and the related notion of the "polytope algebra" have been intensively studied (see, for example, [McM78], [McM93], and [Mor93b]). Various identities discovered in this area might prove useful in dealing with particular counting problems. Specifically, we mention the following result. Let $\{K_i \mid i = 0, \ldots, m\}$ be a finite family of convex polyhedral cones in $\mathbb{R}^d$ and $\{\epsilon_i \mid i = 1, \ldots, m\}$ a family of real numbers such that $K_0 = \epsilon_1 \cdot K_1 + \ldots + \epsilon_m \cdot K_m$, where the identity is understood as an identity for the indicator functions of the sets $K_0, \ldots, K_m$. Then the same identity $K_0^* = \epsilon_1 \cdot K_1^* + \ldots + \epsilon_m \cdot K_m^*$ holds for the dual cones $K_i^* = \left\{x \in \mathbb{R}^d \mid \langle x, y\rangle \leq 0 \text{ for all } y \in K_i\right\}$.

For example, if the polytope P_b of the example above is not simple, we can use the above formula to represent the cone of feasible directions at its vertex as a combination of unimodular cones in the following way. First, triangulating the normal cone at the vertex, we represent it as a combination of unimodular cones. Then, passing to the dual cones, we get the desired representation of the cone of feasible directions.

ANALYTICAL METHODS

The number $\#(P \cap \Lambda)$ of lattice points of Λ in the polytope P can be interpreted as the integral over P of the periodic delta-function

$$\sum_{y \in \Lambda} \delta_y(x) = (\det \Lambda)^{-1} \sum_{l \in \Lambda^*} \exp\{2\pi i \langle l, x\rangle\}$$

where $\Lambda^* = \left\{l \in \mathbb{R}^d \mid \langle l, y\rangle \in \mathbb{Z} \text{ for all } y \in \Lambda\right\}$ is the ***dual lattice***. Depending on the interpretation of this integral one can get various formulas. For example, if the above series is approximated as $t \longrightarrow \infty$ by the theta-series

$$\theta_t(x) = t^{d/2} \sum_{y \in \Lambda} \exp\{-t\pi \|x - y\|^2\} = (\det \Lambda)^{-1} \sum_{l \in \Lambda^*} \exp\{-\pi \|l\|^2 / t\} \exp\{2\pi i \langle l, x\rangle\},$$

then as the limit $\lim_{t \longrightarrow \infty} \int_P \theta_t(x) dx$ one gets the number of lattice points in P, each lattice point y counted with weight equal to the spherical measure of the cone K_y of feasible directions at y normalized in such a way that the spherical measure of $\mathbb{R}^d$ is equal to 1 (see [GL87] for some information about this weighted counting). To evaluate the integral one can repeatedly apply Stokes's formula to the faces of a polytope.

Applying Parseval's theorem one can get the famous Siegel identity (see [GL87])

$$2^d \det \Lambda - \text{vol } B = \frac{1}{\text{vol } B} \sum_{l \in \Lambda^* \setminus 0} \left| \int_B \exp\{-\pi i \langle l, x\rangle\} dx \right|^2,$$

where B is a 0-symmetric convex body not containing nonzero lattice points (cf. Theorem 7.2.2).

Quite a few useful inequalities for the number of lattice points can be found in [GW93], [Lag95], and [GL87]. Blichfeldt's inequality states that $\#(B \cap \Lambda) \leq \frac{d!}{\det \Lambda} \text{vol } B + d$, where B is a convex body containing at least $d+1$ affinely independent lattice points. Davenport's inequality implies that

$$\#(B \cap \mathbb{Z}^d) \leq \sum_{i=0}^{d} \binom{d}{i} V_i(B),$$

where the V_i are the intrinsic volumes. A conjectured stronger inequality, $\#(B \cap \mathbb{Z}^d) \leq V_0(K) + \ldots + V_d(K)$, was shown to be false in dimensions $d \geq 207$ although it is correct for $d = 2, 3$. Furthermore, H. Hadwiger proved that $\#(B \cap \mathbb{Z}^d) \geq \sum_{i=0}^{d} (-1)^{d-i} V_i(B)$, provided $B \subset \mathbb{R}^d$ is a convex body having a nonempty interior (see [Lag95]).

7.4 ASYMPTOTIC PROBLEMS

If $P \subset \mathbb{R}^d$ is an integral polytope then the number of integral points in the dilated polytope $nP = \{nx \mid x \in P\}$ for a natural number n is a polynomial in n, known as the Ehrhart polynomial. We review several results concerning the Ehrhart polynomial and its generalizations.

GLOSSARY

Todd polynomial: The homogeneous polynomial $\text{td}_k(x_1, \ldots, x_m)$ of degree k defined as the coefficient of t^k in the expansion

$$\prod_{i=1}^{m} \frac{tx_i}{1 - \exp\{-tx_i\}} = \sum_{k=0}^{\infty} t^k \cdot \text{td}_k(x_1, \ldots, x_m).$$

Tangent cone at a face of a polytope: The cone K_F of feasible directions at any point in the relative interior of the face $F \subset P$.

Apex of a cone: The largest linear subspace contained in the cone.

Dual cone: The cone $K^* = \{x \in \mathbb{R}^d \mid \langle x, y \rangle \leq 0 \text{ for all } y \in K\}$, where $K \subset \mathbb{R}^d$ is a given cone.

vol_k: The normalized k-volume of a k-dimensional rational polytope $P \subset \mathbb{R}^d$ computed as follows. Let $L \subset \mathbb{R}^d$ be the k-dimensional linear subspace parallel to the affine span of P. Then $\text{vol}_k(P)$ is the Euclidean k-dimensional volume of P in the affine span of P divided by the determinant of the lattice $\Lambda = \mathbb{Z}^d \cap L$.

Ridge: A $(d-2)$-dimensional face of a d-dimensional polytope.

EHRHART POLYNOMIALS

The following fundamental result was suggested by Ehrhart (see, for example, [Sta86] and [Sta83]).

THEOREM 7.4.1

Let $P \subset \mathbb{R}^d$ be an integral polytope. For a natural number n we denote by $nP = \{nx \mid x \in P\}$ the n-fold dilatation of P. Then the number of integral points in nP is a polynomial in n:

$$\#(nP \cap \mathbb{Z}^d) = E_P(n) \quad \textit{for some polynomial} \quad E_P(x) = \sum_{i=0}^{d} e_i(P) \cdot x^i.$$

Moreover, for positive integers n the value of $(-1)^{\deg E_P} E_P(-n)$ is equal to the number of integral points in the relative interior of the polytope nP (the "reciprocity law").

The polynomial E_P is called the ***Ehrhart polynomial*** and its coefficients $e_i(P)$ are called ***Ehrhart coefficients***. For various proofs of Theorem 7.4.1 see, for example, [McM78], [Sta86], and [Sta83]. The existence of the Ehrhart polynomials and the reciprocity law can be derived from the single fact that the number of integral points in a polytope is a lattice-translation-invariant valuation (see [McM93] and Section 7.3).

If P is a rational polytope, we define $e_k(P) = n^{-k} e_k(P_1)$, where n is a positive integer such that $P_1 = nP$ is an integral polytope. For an integral polytope $P \subset \mathbb{R}^d$, one has $\#(P \cap \mathbb{Z}^d) = e_0(P) + e_1(P) + \ldots + e_d(P)$. (This formula is no longer true, however, if P is a general rational polytope.) The Ehrhart coefficients constitute a basis of all additive functions (valuations) ν on rational polytopes that are invariant under unimodular transformations (see [McM93] and [GW93]).

GENERAL PROPERTIES

It is known that $e_0(P) = 1$, $e_d(P) = \mathrm{vol}_d(P)$, and $e_{d-1}(P) = \frac{1}{2} \sum_F \mathrm{vol}_{d-1} F$, where the sum is taken over all the facets of P. Thus, computation of the two highest coefficients reduces to computation of the volume. In fact, the computation of any fixed number of the highest Ehrhart coefficients of an $\mathcal{H}$-polytope reduces in polynomial time to the computation of the volumes of faces [Bar94a]; see also below.

EXISTENCE OF LOCAL FORMULAS

The Ehrhart coefficients can be decomposed into a sum of "local" summands. The following theorem was proven by P. McMullen (see [McM78], [McM93], and [Mor93b]).

THEOREM 7.4.2

For any natural numbers k and d there exists a real valued function $\mu_{k,d}$, defined on the set of all rational polyhedral cones $K \subset \mathbb{R}^d$, such that for every rational full-dimensional polytope $P \subset \mathbb{R}^d$ we have

$$e_k(P) = \sum_F \mu_{k,d}(K_F) \cdot \mathrm{vol}_k F,$$

where the sum is taken over all k-dimensional faces F of P and K_F is the tangent

cone at the face F. Moreover, one can choose $\mu_{k,d}$ to be an additive measure on polyhedral cones.

The function $\mu_{k,d}$ that satisfies the conditions of Theorem 7.4.2 is not unique and it is a difficult problem to choose a computationally efficient $\mu_{k,d}$ (see also Morelli's formulas, below). However, for some specific values of k and d a "canonical" choice of $\mu_{k,d}$ has long been known.

EXAMPLE

For a polyhedral cone $K \subset \mathbb{R}^d$ let

$$\mu(K) = \int_K \exp\{-\pi\|x\|^2\}dx,$$

where dx is the Euclidean measure in $\mathbb{R}^d$ and $\|\cdot\|$ is the Euclidean norm. Thus $\mu(K) = 1$ if $K = \mathbb{R}^d$ and $\mu(K) = 0.5$ if K is a halfspace. One can choose $\mu_{d,d} = \mu_{d-1,d} = \mu$ because of the formulas for $e_d(P)$ and $e_{d-1}(P)$ (see above).

On the other hand, one can choose $\mu_{0,d}(K) = \mu(K^*)$, where K^* is the dual cone, since it is known that $e_0(P) = 1$. We note that if $\mu(K)$ is an additive measure on polyhedral cones then $\nu(K) = \mu(K^*)$ is also an additive measure on polyhedral cones (see Section 7.3).

TOTALLY UNIMODULAR POLYTOPES

Let $P \subset \mathbb{R}^d$ be a full-dimensional totally unimodular polytope. Let $\{l_i \mid i = 1,\ldots m\}$ be the set of integral outer normals to the facets of P. We assume that the l_i are *primitive*, i.e., $\alpha l_i \notin \mathbb{Z}^d$ for any i and any $0 < \alpha < 1$. Say $P = \{x \in \mathbb{R}^d \mid \langle l_i, x\rangle \le b_i \text{ for } i = 1,\ldots,m\}$ for some $b_1,\ldots,b_m \in \mathbb{Z}$. Let $h = (h_1,\ldots,h_m) \in \mathbb{R}^m$ be a vector. If $\|h\|$ is small enough, then the "perturbed" polytope $P_h = \{x \in \mathbb{R}^d \mid \langle l_i, x\rangle \le b_i + h_i\}$ has the same "shape" as P and the volume of P_h is a polynomial function of h.

The following expression for the Ehrhart coefficient $e_k(P)$ was found in [KP92]:

$$e_{d-k}(P) = \mathrm{td}_k\Big(\frac{\partial}{\partial h_1},\ldots,\frac{\partial}{\partial h_m}\Big)\mathrm{vol}_d(P_h)\big|_{h=0}.$$

Thus $\mathrm{td}_0 = 1$, $\mathrm{td}_1(x_1,\ldots,x_m) = (x_1 + \ldots + x_m)/2$, etc.

A FORMULA FOR $e_{d-2}(P)$.

Let $P \subset \mathbb{R}^d$ be a full-dimensional simple integral polytope. As above, we can formally define some coefficients

$$b_{d-k}(P) = \mathrm{td}_k\Big(\frac{\partial}{\partial h_1},\ldots,\frac{\partial}{\partial h_m}\Big)\mathrm{vol}_d(P_h)\big|_{h=0}.$$

However, $b_{d-k}(P)$ are no longer Ehrhart coefficients if P is not totally unimodular. To get $e_{d-2}(P)$ one should introduce a correction term for every ridge of P.

Let F be a ridge of P. There exist exactly two facets, G_1 and G_2, that contain F. Let $l_1, l_2 \in \mathbb{Z}^d$ be primitive outer normals to G_1, G_2 respectively (see above).

Finally, let $\Lambda = \mathbb{Z}^d \cap L$, where L is a two-dimensional linear subspace orthogonal to F. Choosing an appropriate basis in Λ we may write $l_1 = (1,0)$ and $l_2 = (p,q)$ for coprime $0 \le p < q$. Let us define a function $\tau(K_F)$ on the tangent cone at the ridge F:

$$\tau(K_F) = s(p,q) + \frac{1}{4} - \frac{1}{4q}, \quad \text{where } s(p,q) \text{ is the Dedekind sum (see Section 7.3).}$$

Then $e_{d-2}(P) = b_{d-2}(P) + \sum_F \tau(K_F) \cdot \mathrm{vol}_{d-2}(F)$, where the summation is taken over all ridges F of P. This formula was found in [KK93] (see also [Pom93]). The appearance of the Dedekind sums arises from an explicit procedure for the resolution of singularities of toric varieties in codimension 2 [Oda88]; see also Section 7.3.

MORELLI'S FORMULAS

General formulas for $e_k(P)$ were obtained in [Mor93a]. Morelli constructed an explicit measure $\mu_{k,d}(K)$ as in Theorem 7.4.2, which, however, is not a real number but a real-valued rational function on the Grassmanian $\mathbf{G}_{k+1}(\mathbb{R}^d)$ of all $(k+1)$-dimensional subspaces in $\mathbb{R}^d$. Let K be a full-dimensional cone whose apex is a k-dimensional subspace (if K is not such a cone then $\mu_{k,d}(K) = 0$). There is an explicit formula for $\mu_{k,d}(K) : \mathbf{G}_{k+1}(\mathbb{R}^d) \longrightarrow \mathbb{R}$ when the dual k-dimensional cone $K^* \subset \mathbb{R}^d$ is unimodular. If K^* is not unimodular, then we define $\mu_{k,d}(K)$ using the additivity of $\mu_{k,d}$ (cf. the discussion in Section 7.3 about decomposing a polyhedral cone into unimodular cones). The cone K contains $d-k$ $(k+1)$-dimensional halfspaces ("edges") whose intersection is the k-dimensional apex V of K. Let E_s, $s = 1, \ldots, d-k$, be the linear spans of these edges. For every s we choose an oriented basis $(b_1^s, \ldots, b_{k+1}^s)$ of the $(k+1)$-dimensional lattice $(E_s \cap \mathbb{Z}^d)$, so that all these orientations are coherent with some fixed orientation of the apex V. Let $A \in \mathbf{G}_{k+1}(\mathbb{R}^d)$ be a $(k+1)$-dimensional subspace. We define the value of $\mu_{k,d}(K)$ on A as follows: Choose any basis $u_1, \ldots, u_{k+1}$ of A. Define a $(k+1)\times(k+1)$ matrix M^s by the formula $M^s_{ij} = \langle b_i^s, u_j \rangle$. Let $f_s = \det M^s$ and define $\mu_{k,d}(K)$ on A to be equal to

$$\frac{\mathrm{td}_{d-k}(f_1, \ldots, f_{d-k})}{f_1 \cdots f_{d-k}}.$$

If $d-k$ is fixed then the function $\mu_{k,d}(K) : \mathbf{G}_{k+1}(\mathbb{R}^d) \longrightarrow \mathbb{R}$ is polynomially computable [Bar94a]. Therefore, computation of any fixed number of the highest Ehrhart coefficients reduces in polynomial time to computation of the volumes of faces for a an $\mathcal{H}$-polytope [Bar94a].

THE h^*-VECTOR

General properties of generating functions (see [Sta86]) imply that for every integral d-dimensional polytope P there exist integers $h_0^*(P), \ldots, h_d^*(P)$ such that

$$\sum_{n=0}^{\infty} E_P(n) x^n = \frac{h_0^*(P) + h_1^*(P)x + \ldots + h_d^*(P)x^d}{(1-x)^{d+1}}.$$

The $(d+1)$-vector $h^*(P) = \big(h_0^*(P), \ldots, h_d^*(P)\big)$ is called the h^*-vector of P. It is clear that $h^*(P)$ is a (vector-valued) valuation on the set of integral polytopes and that $h^*(P)$ is invariant under a unimodular transformation of $\mathbb{Z}^d$. Moreover, the functions $h_k^*(P)$ constitute a basis of all valuations on integral polytopes that are invariant under unimodular transformations. Unlike the Ehrhart coefficients $e_k(P)$, the numbers $h_k^*(P)$ are not homogeneous. However, $h_k^*(P)$ are monotone (and, therefore, nonnegative): if $Q \subset P$ are two integral polytopes then $h_k^*(P) \geq h_k^*(Q)$ ([Sta93]). This property follows from the fact that polytopes admit triangulations that are Cohen-Macaulay complexes (see Chapter 15). If these complexes are Gorenstein then one gets the ***Dehn-Sommerville equations*** $h_k^*(P) = h_{d-k}^*(P)$. For example, the h^*-vector of the Birkhoff polytope B_n (see Section 7.1) satisfies the Dehn-Sommerville equations (see [Sta83]).

In principle, there is a combinatorial way to calculate $h^*(P)$. Namely, let Δ be a triangulation of P such that every d-dimensional simplex of Δ is integral and has volume $1/d!$ (see Section 7.2). Let $f_k(\Delta)$ be the number of k-dimensional faces of the triangulation Δ. Then

$$h_k^*(P) = \sum_{i=0}^{k} (-1)^{k-i} \binom{d-i}{d-k} f_{i-1}(\Delta),$$

where we let $f_{-1}(\Delta) = 1$. Such a triangulation may not exist for the polytope P but it exists for mP, where m is a sufficiently large integer (see [Mor93a]). Generally, this triangulation Δ would be too big, but for some special polytopes with nice structure (for example, for the so-called *poset polytopes*) it may provide a very good way to compute $h^*(P)$ and hence the Ehrhart polynomial E_P.

Since the number of integral points in a polytope is a valuation, we get the following result (see [McM93]).

THEOREM 7.4.3

Let $P_1, \ldots, P_m$ be integral polytopes in $\mathbb{R}^d$. For an m-tuple of natural numbers $\boldsymbol{n} = (n_1, \ldots, n_m)$, let us define the polytope

$$P(\boldsymbol{n}) = \{n_1 x_1 + \ldots + n_m x_m \mid x_1 \in P_1, \ldots x_m \in P_m\}$$

(using "+" for Minkowski addition one can also write $P(\boldsymbol{n}) = n_1 P_1 + \ldots + n_m P_m$). Then there exists a polynomial $p(x_1, \ldots, x_m)$ of degree at most d such that

$$\#\Big(P(\boldsymbol{n}) \cap \mathbb{Z}^d\Big) = p(n_1, \ldots, n_m).$$

An interpretation of the values $p(n_1, \ldots, n_m)$ for nonpositive integer values of $n_1, \ldots, n_m$ can be obtained by using the polytope algebra identities (see [McM93]).

More generally, the existence of local formulas for the Ehrhart coefficients implies that the number of integral points in an integral polytope $P_h = \{x \in \mathbb{R}^d \mid Ax \leq b + h\}$ is a polynomial in h provided P_h is an integral polytope combinatorially isomorphic to the integral polytope P_0. In other words, if we move the facets of an integral polytope so that it remains integral and has the same facial structure, then the number of integral points varies polynomially.

INTEGRAL POINTS IN RATIONAL POLYTOPES

If P is a rational (not necessarily integral) polytope then $\#(nP \cap \mathbb{Z}^d)$ is not a polynomial but a ***quasi-polynomial*** (a function of n whose value cycles through the values of a finite list of polynomials). The following result was independently proven by P. McMullen and R. Stanley (see [McM78] and [McM93]).

THEOREM 7.4.4

Let $P \subset \mathbb{R}^d$ be a rational polytope. For every r, $0 \leq r \leq d$, let ind_r be the smallest natural number k such that all r-dimensional faces of kP are integral polytopes. Then, for every $n \in \mathbb{N}$,

$$\#(nP \cap \mathbb{Z}^d) = \sum_{r=0}^{d} e_r\big(P, n(\mathrm{mod}\ \ \mathrm{ind}_r)\big) \cdot n^r$$

for suitable rational numbers $e_r(P,k)$, $0 \leq k < \mathrm{ind}_r$.

A generalization of the "reciprocity law" also appears in [McM78].

7.5 PROBLEMS WITH QUANTIFIERS

A natural generalization of the decision problem (see Section 7.2) is a problem with quantifiers. We describe some known results and formulate open questions for this class of problems.

FROBENIUS PROBLEM

The most famous problem from this class is the ***Frobenius problem***:

Given k positive integers $a_1, \ldots, a_k$ with greatest common divisor 1, find the largest integer m that cannot be represented as an integer combination $a_1 n_1 + \ldots + a_k n_k$, $n_i \geq 0$.

The problem is known to be NP-hard in general, but a polynomial time algorithm is known for fixed k [Kan92].

PROBLEM WITH QUANTIFIERS

A general ***problem with quantifiers*** can be formulated as follows. Suppose that P is a Boolean combination of convex polyhedra: we start with some polyhedra $P_1, \ldots, P_k \subset \mathbb{R}^d$ given by their facet descriptions and construct P by using the set-theoretical operations of union, intersection, and complement. We want to find out if the formula

$$\exists x_1 \forall x_2 \exists x_3 \ldots \forall x_m : (x_1, \ldots, x_m) \in P \tag{7.5.1}$$

is true. Here x_i is an integral vector from $\mathbb{Z}^{d_i}$, and, naturally, $d_1 + \ldots + d_m = d$, $d_i \geq 0$. The parameters that characterize the size of (7.5.1) can be divided

into two classes. The first class consists of the parameters characterizing the *combinatorial size* of the formula. These are the dimension d, the number $m-1$ of quantifier alternations, the number of linear inequalities and Boolean operations that define the polyhedral set P. The parameters from the other class characterize the *numerical size* of the formula. Those are the bit sizes of the numbers involved in the inequalities that define P.

The following fundamental question remains open.

PROBLEM 7.5.1

Let us fix all the combinatorial parameters of the formula (7.5.1). *Does there exist a polynomial time algorithm that checks whether this formula is true?*

Naturally, "polynomial time" means that the running time of the algorithm is bounded by a polynomial in the numerical size of the formula. The answer to this question is unknown although it is widely believed that such an algorithm indeed exists. A polynomial time algorithm is known if the formula contains not more than 1 quantifier alternation, i.e., if $m \leq 2$ ([Kan90]). A related problem, whose solution is not known even for the simplest formulas with quantifiers, is to compute the number of solutions for quantifier-free variables in a formula with quantifiers.

Problems with quantifiers have some apparent connections with parametric integer programming and test sets for integer programs (see [Kan90]). Geometrically, we are seeking to describe sets of points obtained from the set of integral points in a polyhedron by using iterated operations consisting of (rational) projection and taking the complement in some lattice. An equivalent formulation: we are seeking to describe the class of sets obtained from a polyhedron by using the iterated operations of taking the union of lattice shifts and taking complements. In particular, a problem with at most one quantifier alternation reduces to testing whether $\overline{P+\Lambda} = \emptyset$, where $P \subset \mathbb{R}^d$ is a polyhedron, $\Lambda \subset \mathbb{R}^d$ is a lattice, and the overbar denotes complementation (see [KL88] and [Kan92]).

7.6 SOURCES AND RELATED MATERIAL

RELATED CHAPTERS

Chapter 3: Tilings
Chapter 13: Basic properties of convex polytopes
Chapter 15: Face numbers of polytopes and complexes
Chapter 27: Computational convexity
Chapter 39: Mathematical programming

REFERENCES

[AHU74] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, Reading, 1974.

[Ban95] W. Banaszczyk. Inequalities for convex bodies and polar reciprocal lattices in $\mathbb{R}^n$. *Discrete Comput. Geom.*, 13:217–231, 1995.

[Bar94a] A.I. Barvinok. Computing the Ehrhart polynomial of a convex lattice polytope. *Discrete Comput. Geom.*, 12:35–48, 1994.

[Bar94b] A.I. Barvinok. A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Math. Oper. Res.*, 19:769–779, 1994.

[Bri92] M. Brion. Polyèdres et réseaux. *L' Enseignement Math.*, 38:71–88, 1992.

[BS96] L.J. Billera and A. Sarangarajan. Combinatorics of permutation polytopes. In L.J. Billera, C. Greene, R. Simion, and R. Stanley, editors, *Formal Power Series and Algebraic Combinatorics*, volume 24 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 1–23. American Math. Soc., Providence, 1996.

[BV92] I. Bárány and A.M. Vershik. On the number of convex lattice polytopes. *Geom. Funct. Anal.*, 2:381–393, 1992.

[BZ88] A.D. Berenstein and A.V. Zelevinsky. Tensor product multiplicities and convex polytopes in the partition space. *J. Geom. Phys.*, 5:453–472, 1988.

[CCK+94] M. Conforti, G. Cornuéjols, A. Kapoor, K. Vušković, and M.R. Rao. Balanced matrices. In J.R. Birge and K. Murty, editors, *Mathematical Programming, State of the Art 1994*. University of Michigan, Ann Arbor, 1994.

[CHKM92] W.J. Cook, M. Hartmann, R. Kannan, and C. McDiarmid. On integer points in polyhedra. *Combinatorica*, 12:27–37, 1992.

[CS94] S.E. Cappell and J.L. Shaneson. Genera of algebraic varieties and counting of lattice points. *Bull. Amer. Math. Soc. (N.S.)*, 30:62–69, 1994.

[Dye91] M. Dyer. On counting lattice points in polyhedra. *SIAM J. Comput.*, 20:695–707, 1991.

[EKK84] V.A. Emelichev, M.M. Kovalev, and M.K. Kravtsov. *Polytopes, Graphs and Optimization*. Cambridge University Press, Cambridge, 1984.

[GK94] P. Gritzmann and V. Klee. On the complexity of some basic problems in computational convexity, II: volume and mixed volumes. In T. Bisztriczky, P. McMullen, R. Schneider, and A. Ivić Weiss, editors, *Polytopes: Abstract, Convex and Computational*, volume 440 of *Nato Adv. Sci. Inst. Ser. C: Math. Phys. Sci.*, pages 373–466. Kluwer, Dordrecht, 1994.

[GL87] P.M. Gruber and C.G. Lekkerkerker. *Geometry of Numbers*. North Holland, Amsterdam, 2nd edition, 1987.

[GLS88] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1988.

[GW93] P. Gritzmann and J.M. Wills. Lattice points. In P. M. Gruber and J. M. Wills, editors, *Handbook of Convex Geometry*, pages 765–797. Elsevier, Amsterdam, 1993.

[Kan90] R. Kannan. Test sets for integer programs, $\forall\exists$ sentences. In W. Cook and P. D. Seymour, editors, *Polyhedral Combinatorics*, volume 1 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 39–47. American Mathematical Society, Providence, 1990.

[Kan92] R. Kannan. Lattice translates of a polytope and the Frobenius problem. *Combinatorica*, 12:161–177, 1992.

[KK93] J.-M. Kantor and A. Khovanskii. Une application du théorème de Riemann-Roch combinatoire au polynôme d'Ehrhart des polytopes entiers de $\mathbb{R}^d$. *C. R. Acad. Sci. Paris Sér. I Math.*, 317:501–507, 1993.

[KL88] R. Kannan and L. Lovász. Covering minima and lattice-point-free convex bodies. *Ann. of Math.*, 128:577–602, 1988.

[KP92] A.G. Khovanskii and A.V. Puhlikov. A Riemann-Roch theorem for integrals and sums of quasipolynomials on virtual polytopes (Russian). *Algebra i Analiz*, 4:188–216, 1992. Translated in *St.-Petersb. Math. J.* 4, 1993 (789–812).

[Lag95] J.C. Lagarias. Point lattices. In R. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, pages 919–966. North Holland, Amsterdam, 1995.

[Law91] J. Lawrence. Rational-function-valued valuations on polyhedra. In J.E. Goodman, R. Pollack, and W. Steiger, editors, *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, volume 6 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 199–208. AMS-ACM, Providence/Baltimore, 1991.

[McM78] P. McMullen. Lattice invariant valuations on rational polytopes. *Arch. Math.*, 31:509–516, 1978.

[McM93] P. McMullen. Valuations and dissections. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, volume B, pages 933–988. North-Holland, Amsterdam, 1993.

[Mor93a] R. Morelli. Pick's theorem and the Todd class of a toric variety. *Adv. Math.*, 100:183–231, 1993.

[Mor93b] R. Morelli. A theory of polyhedra. *Adv. Math.*, 97:1–73, 1993.

[Oda88] T. Oda. *Convex Bodies and Algebraic Geometry: an Introduction to the Theory of Toric Varieties.* Springer-Verlag, Berlin, 1988.

[Pom93] J. Pommersheim. Toric varieties, lattice points and Dedekind sums. *Math. Ann.*, 295:1–24, 1993.

[Sca85] H.E. Scarf. Integral polyhedra in three space. *Math. Oper. Res.*, 10:403–438, 1985.

[Sch86] A. Schrijver. *The Theory of Linear and Integer Programming.* Wiley, Chichester, 1986.

[Sta83] R.P. Stanley. *Combinatorics and Commutative Algebra*, volume 41 of *Progress in Mathematics.* Birkhäuser, Boston, 1983.

[Sta86] R.P. Stanley. *Enumerative Combinatorics*, volume 1. Wadsworth and Brooks/Cole, Monterey, 1986.

[Sta93] R.P. Stanley. A monotonicity property of h-vectors and h^*-vectors. *Europ. J. Combin.*, 14:251–258, 1993.

[Zie94] G.M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Math.* Springer-Verlag, Berlin, 1994.

8 EUCLIDEAN RAMSEY THEORY

R.L. Graham

INTRODUCTION

Ramsey theory typically deals with problems of the following type. We are given a set S, a family $\mathcal{F}$ of subsets of S, and a positive integer r. We would like to decide whether or not for every partition of $S = C_1 \cup \cdots \cup C_r$ into r subsets, it is always true that some C_i contains some $F \in \mathcal{F}$. If so, we abbreviate this by writing $S \stackrel{r}{\longrightarrow} \mathcal{F}$ (and we say S is r-Ramsey). If not, we write $S \stackrel{r}{\not\longrightarrow} \mathcal{F}$. (For a comprehensive treatment of Ramsey theory, see [GRS90].)

In Euclidean Ramsey theory, S is usually taken to be the set of points in some Euclidean space $\mathbb{E}^N$, and the sets in $\mathcal{F}$ are determined by various geometric considerations. The case most studied is the one in which $\mathcal{F} = \mathrm{Cong}(X)$ consists of all *congruent* copies of a fixed finite configuration $X \subset S = \mathbb{E}^N$. In other words, $\mathrm{Cong}(X) = \{gX \mid g \in SO(N)\}$, where $SO(N)$ denotes the special orthogonal group acting on $\mathbb{E}^N$.

Further, we say that X is *Ramsey* if, for all r, $\mathbb{E}^N \stackrel{r}{\longrightarrow} \mathrm{Cong}(X)$ holds provided N is sufficiently large (depending on X and r). This we indicate by writing $\mathbb{E}^N \longrightarrow X$.

Another important case we will discuss (in Section 8.4) is that in which $\mathcal{F} = \mathrm{Hom}(X)$ consists of all *homothetic* copies $aX + \overline{t}$ of X, where a is a positive real and $\overline{t} \in \mathbb{E}^N$. Thus, in this case $\mathcal{F}$ is just the set of all images of X under the group of positive homotheties acting on $\mathbb{E}^N$.

It is easy to see that any Ramsey (or r-Ramsey) set must be finite. A standard compactness argument shows that if $\mathbb{E}^N \stackrel{r}{\longrightarrow} X$ then there is always a *finite* set $Y \subseteq \mathbb{E}^N$ such that $Y \stackrel{r}{\longrightarrow} X$. Also, if X is Ramsey (or r-Ramsey) then so is any homothetic copy $aX + \overline{t}$ of X.

GLOSSARY

$\mathbb{E}^N \stackrel{r}{\longrightarrow}$ Cong (X): For any partition $\mathbb{E}^N = C_1 \cup \cdots \cup C_r$, some C_i contains a set congruent to X. We say that X is ***r*-Ramsey**. When $\mathrm{Cong}(X)$ is understood we will usually write $\mathbb{E}^N \stackrel{r}{\longrightarrow} X$.

$\mathbb{E}^N \longrightarrow X$: For every r, $\mathbb{E}^N \stackrel{r}{\longrightarrow} \mathrm{Cong}(X)$ holds, provided N is sufficiently large. We say in this case that X is ***Ramsey***.

8.1 r-RAMSEY SETS

In this section we focus on low-dimensional r-Ramsey results. We begin by stating three conjectures.

CONJECTURE 8.1.1

For any nonequilateral triangle T (i.e., the set of 3 vertices of T),

$$\mathbb{E}^2 \xrightarrow{2} T.$$

CONJECTURE 8.1.2 (stronger)

For any partition $\mathbb{E}^2 = C_1 \cup C_2$, every triangle occurs (up to congruence) in C_1, or else the same holds for C_2, with the possible exception of a single equilateral triangle.

The partition $\mathbb{E}^2 = C_1 \cup C_2$ with

$$\begin{aligned} C_1 &= \{(x,y) \mid -\infty < x < \infty, 2m \le y < 2m+1, m = 0, \pm 1, \pm 2, \ldots\} \\ C_2 &= \mathbb{E}^2 \setminus C_1 \end{aligned}$$

into alternating half-open strips of width 1 prevents the equilateral triangle of side $\sqrt{3}$ from occurring in a single C_i. In fact, it is conjectured that except for some freedom in assigning the boundary points (x, m), m an integer, this is the only way of avoiding *any* triangle.

CONJECTURE 8.1.3

For any triangle T,

$$\mathbb{E}^2 \xrightarrow{3} T.$$

In the positive direction, we have [EGM+75b]:

THEOREM 8.1.4

(a) $\mathbb{E}^2 \xrightarrow{2} T$ *if T is a triangle satisfying:*

(i) *T has a ratio between two sides equal to $2\sin\theta/2$ with $\theta = 30°, 72°, 90°,$ or $120°$*

(ii) *T has a $30°$, $90°$, or $150°$ angle* [Sha76]

(iii) *T has angles $(\alpha, 2\alpha, 180° - 3\alpha)$ with $0 < \alpha < 60°$*

(iv) *T has angles $(180° - \alpha, 180° - 2\alpha, 3\alpha - 180°)$ with $60° < \alpha < 90°$*

(v) *T is the degenerate triangle $(a, 2a, 3a)$*

(vi) *T has sides (a, b, c) satisfying*

$$a^6 - 2a^4b^2 + a^2b^4 - 3a^2b^2c^2 + b^2c^2 = 0$$

or

$$a^4c^2 + b^4a^2 + c^4b^2 - 5a^2b^2c^2 = 0$$

(vii) *T has sides (a, b, c) satisfying*

$$c^2 = a^2 + 2b^2 \quad \textit{with} \quad a < 2b \qquad \text{[Sha76]}$$

(viii) *T has sides (a, b, c) satisfying*

$$a^2 + c^2 = 4b^2 \textit{ with } \; 3b^2 < 2a^2 < 5b^2 \qquad \text{[Sha76]}$$

(ix) *T has sides equal in length to the sides and circumradius of an isosceles triangle;*

(b) $\mathbb{E}^3 \xrightarrow{2} T$ *for any nondegenerate triangle* T

(c) $\mathbb{E}^3 \xrightarrow{3} T$ *for any nondegenerate right triangle* T [BT96]

(d) $\mathbb{E}^3 \overset{12}{\not\longrightarrow} T$, *a triangle with angles* $(30°, 60°, 90°)$ [Bón93]

(e) $\mathbb{E}^2 \overset{2}{\not\longrightarrow} Q^2$ *(4 points forming a square)*

(f) $\mathbb{E}^4 \overset{2}{\not\longrightarrow} Q^2$ [Can96]

(g) $\mathbb{E}^5 \xrightarrow{2} R^2$, *any rectangle* [Tót96]

(h) $\mathbb{E}^n \overset{4}{\not\longrightarrow}$ *for any* n *(a degenerate* $(1, 1, 2)$ *triangle)*

(i) $\mathbb{E}^n \overset{16}{\not\longrightarrow}$ *for any* n *(a degenerate* $(a, b, a + b)$ *triangle).*

It is not known whether the 4 in (h) or the 16 in (i) can be replaced by smaller values. Other results of this type can be found in [EGM+73], [EGM+75a], [EGM+75b], [Sha76], [CFG91].

The 2-point set X_2 consisting of two points a unit distance apart is the simplest set about which such questions can be asked, and has a particularly interesting history (see [Soi91] for details). It is clear that

$$\mathbb{E}^1 \overset{2}{\not\longrightarrow} X_2 \quad \text{and} \quad \mathbb{E}^2 \xrightarrow{2} X_2.$$

To see that $\mathbb{E}^2 \xrightarrow{3} X_2$, consider the 7-point Moser graph shown in Figure 8.1.1. All edges have length 1. On the other hand, $\mathbb{E}^2 \overset{7}{\not\longrightarrow} X_2$, which can be seen by an appropriate periodic 7-coloring (= partition into 7 parts) of a tiling of $\mathbb{E}^2$ by regular hexagons of diameter 0.9 (see Figure 1.3.1).

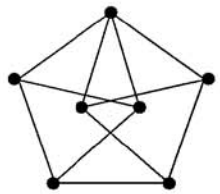


FIGURE 8.1.1
The Moser graph.

Definition: The ***chromatic number*** of $\mathbb{E}^n$, denoted by $\chi(\mathbb{E}^n)$, is the least m such that $\mathbb{E}^n \overset{m}{\not\longrightarrow} X_2$.

By the above remarks,

$$4 \leq \chi(\mathbb{E}^2) \leq 7.$$

These bounds have remained unchanged for over 45 years.

PROBLEM 8.1.5

Determine the exact value of $\chi(\mathbb{E}^2)$.

The best bounds currently known for $\mathbb{E}^n$ are:

$$(6/5 + o(1))^n < \chi(\mathbb{E}^n) < (3 + o(1))^n$$

(see [FW81], [CFG91]).

A "near miss" for showing $\chi(\mathbb{E}^2) < 7$ was recently found Soifer [Soi92]. He shows that there exists a partition $\mathbb{E}^2 = C_1 \cup \cdots \cup C_7$ where C_i contains no pair of points a distance 1 for $1 \leq i \leq 6$, while C_7 has no pair with distance $1/\sqrt{5}$.

See Section 1.3 for more details.

8.2 RAMSEY SETS

Recall that X is Ramsey (written $\mathbb{E}^N \longrightarrow X$) if, for all r, if $\mathbb{E}^N = C_1 \cup \cdots \cup C_r$ then some C_i must contain a congruent copy of X, provided only that $N \geq N_0(X, r)$.

GLOSSARY

Spherical: X is spherical if it lies on the surface of some sphere.

Rectangular: X is rectangular if it is a subset of the vertices of a rectangular parallelepiped.

Simplex: X is a simplex if it spans $\mathbb{E}^{|X|-1}$.

THEOREM 8.2.1 [EGM+73]

If X and Y are Ramsey then so is $X \times Y$.

Thus, since any 2-point set is Ramsey (for any r, consider the unit simplex S_{2r+1} in $\mathbb{E}^{2r}$ scaled appropriately), then so is any rectangular parallelepiped. This implies:

THEOREM 8.2.2

Any rectangular set is Ramsey.

Frankl and Rödl strengthen this significantly in the following way.

Definition: A set $A \subset \mathbb{E}^n$ is called ***super-Ramsey*** if there exist positive constants c and ϵ and subsets $X = X(N) \subset \mathbb{E}^N$ for every $N \geq N_0(X)$ such that:

(i) $|X| < c^n$;

(ii) $|Y| < |X|/(1+\epsilon)^n$ holds for all subsets $Y \subset X$ containing no congruent copy of A.

THEOREM 8.2.3 [FR90]

(i) *All two-element sets are super-Ramsey.*

(ii) *If A and B are super-Ramsey then so is $A \times B$.*

COROLLARY 8.2.4

If X is rectangular then X is super-Ramsey.

In the other direction we have

THEOREM 8.2.5

Any Ramsey set is spherical.

The simplest nonspherical set is the degenerate $(1, 1, 2)$ triangle.

Concerning simplices, we have the result of Frankl and Rödl:

THEOREM 8.2.6 [FR90]

Every simplex is Ramsey.

In fact, they show that for any simplex X, there is a constant $c = c(X)$ such that for all r,

$$\mathbb{E}^{c \log r} \xrightarrow{r} X,$$

which follows from their result:

THEOREM 8.2.7

Every simplex is super-Ramsey.

It was an open problem for more than 20 years as to whether the set of vertices of a regular pentagon was Ramsey. This was finally settled by Kříž [Kři91] who proved the following two fundamental results:

THEOREM 8.2.8 [Kři91]

Suppose $X \subseteq \mathbb{E}^N$ has a transitive solvable group of isometries. Then X is Ramsey.

COROLLARY 8.2.9

Any set of vertices of a regular polygon is Ramsey.

THEOREM 8.2.10 [Kři91]

Suppose $X \subseteq \mathbb{E}^N$ has a transitive group of isometries that has a solvable subgroup with at most two orbits. Then X is Ramsey.

COROLLARY 8.2.11

The vertex sets of the Platonic solids are Ramsey.

CONJECTURE 8.2.12

Any 4-point subset of a circle is Ramsey.

Kříž [Kři92] has shown this holds if a pair of opposite sides of the 4-point set are parallel (i.e., form a trapezoid).

Certainly, the outstanding open problem in Euclidean Ramsey theory is to determine the Ramsey sets. The author (bravely?) makes the following:

CONJECTURE 8.2.13

Any spherical set is Ramsey.

If true then this would imply that the Ramsey sets are exactly the spherical sets.

8.3 SPHERE-RAMSEY SETS

Since spherical sets play a special role in Euclidean Ramsey theory, it is natural that the following concept arises.

GLOSSARY

$S^N(\rho)$: A sphere in $\mathbb{E}^N$ with radius ρ.

Sphere-Ramsey: X is sphere-Ramsey if, for all r, there exist $N = N(X, r)$ and $\rho = \rho(X, r)$ such that

$$S^N(\rho) \xrightarrow{r} X.$$

In this case we write $S^N(\rho) \longrightarrow X$.

For a spherical set X, let $\rho(X)$ denote its circumradius, i.e., the radius of the smallest sphere containing X as a subset.

Remark. If X and Y are sphere-Ramsey then so is $X \times Y$.

THEOREM 8.3.1 [Gra83]

If X is rectangular then X is sphere-Ramsey.

In [Gra83], it was conjectured that in fact if X is rectangular and $\rho(X) = 1$ then $S^N(1+\epsilon) \longrightarrow X$ should hold. This was proved by Frankl and Rödl [FR90] in a much stronger "super-Ramsey" form.

Concerning simplices, Matoušek and Rödl proved the following spherical analogue of simplices being Ramsey:

THEOREM 8.3.2 [MR95]

For any simplex X with $\rho(X) = 1$, any r, and any $\epsilon > 0$, there exists $N = N(X, r, \epsilon)$ such that

$$S^N(1+\epsilon) \xrightarrow{r} X.$$

The proof uses an interesting mix of techniques from combinatorics, linear algebra, and Banach space theory.

The following results show that the "blowup factor" of $1+\epsilon$ is really needed.

THEOREM 8.3.3 [Gra83]

Let $X = \{x_1, \ldots, x_m\} \subset \mathbb{E}^N$ such that:

(i) *for some nonempty $I \subseteq \{1, 2, \ldots, m\}$, there exist nonzero a_i, $i \in I$, with*

$$\sum_{i \in I} a_i x_i = 0 \in \mathbb{E}^N$$

(ii) *for all nonempty $J \subseteq I$,*

$$\sum_{j \in J} a_j \neq 0.$$

Then X is not sphere-Ramsey.

This implies that $X \subset S^N(1)$ is not sphere-Ramsey if the convex hull of X contains the center of $S^N(1)$.

Definition: A simplex $X \subset \mathbb{E}^N$ is called ***exceptional*** if there is a subset $A \subseteq X$, $|A| \geq 2$, such that the affine hull of A translated to the origin has a nontrivial intersection with the linear span of the points of $X \setminus A$ regarded as vectors.

THEOREM 8.3.4 [MR95]

If X is a simplex with $\rho(X) = 1$ and $S^N(1) \longrightarrow X$ then X must be exceptional.

It is not known whether it is true for exceptional X that $S^N(1) \longrightarrow X$. The simplest nontrivial case is for the set of three points $\{a, b, c\}$ lying on some great circle of $S^N(1)$ (with center o) so that the line joining a and b is parallel to the line joining o and c.

We close with a fundamental conjecture:

CONJECTURE 8.3.5

If X is Ramsey, then X is sphere-Ramsey.

8.4 HOMOTHETIC RAMSEY SETS AND DENSITY THEOREMS

In this section we will survey various results of the type $\mathbb{E}^N \xrightarrow{r} \mathrm{Hom}(X)$, the set of positive homothetic images $aX + \overline{t}$ of a given set X. Thus, we are allowed to dilate and translate X but we cannot rotate it. The classic result of this type is van der Waerden's theorem, which asserts the following:

THEOREM 8.4.1 [van27]

If $X = \{1, 2, \ldots, m\}$ then $\mathbb{E} \xrightarrow{r} \mathrm{Hom}(X)$.

(Note that $\mathrm{Hom}(X)$ is just the set of m-term arithmetic progressions.)

By the compactness theorem mentioned in the Introduction there exists for each m a minimum value $W(m)$ such that

$$\{1, 2, \ldots, W(m)\} \xrightarrow{2} \mathrm{Hom}(X).$$

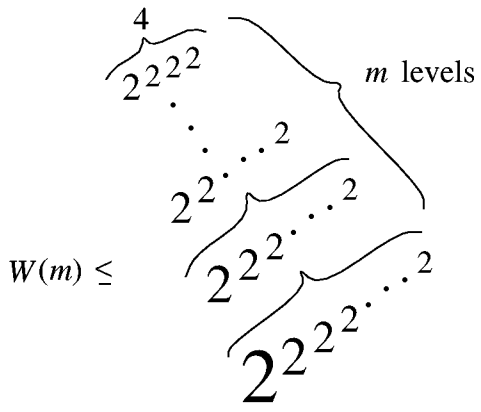
The determination or even estimation of $W(m)$ seems to be extremely difficult. The known values are:

m	1	2	3	4	5
$W(m)$	1	3	9	35	178

The best general result from below (due to Berlekamp—see [GRS90]) is

$$W(p+1) \geq p \cdot 2^p, \quad p \text{ prime.}$$

The best upper bound is given by a result of Shelah [She88]:



The following conjecture of the author has been open for more than 25 years:

CONJECTURE 8.4.2

For all m,

$$W(m) \leq \left.\begin{array}{c} 2^{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}} \\ \end{array}\right\} \ (m \text{ 2's})$$

The generalization to $\mathbb{E}^N$ is due independently to Gallai and Witt (see [GRS90]).

THEOREM 8.4.3

For any finite set $X \subset \mathbb{E}^n$,

$$\mathbb{E}^N \longrightarrow \mathrm{Hom}(X).$$

We remark here that a number of results in (Euclidean) Ramsey theory have stronger so-called *density* versions. As an example, we state the well-known theorem of Szemerédi.

GLOSSARY

$\mathbb{N}$: The set of natural numbers $\{1, 2, 3, \ldots\}$.

$\overline{\delta}(A)$: The ***upper density*** of a set $A \subseteq \mathbb{N}$ is defined by:

$$\overline{\delta}(A) = \limsup_{n \longrightarrow \infty} \frac{|A \cap \{1, 2, \ldots, n\}|}{n} .$$

THEOREM 8.4.4 (Szemerédi [Sze75])

If $A \subseteq \mathbb{N}$ *has* $\overline{\delta}(A) > 0$ *then* A *contains arbitrarily long arithmetic progressions.*

That is, $A \cap \mathrm{Hom}\{1, 2, \ldots, m\} \neq \emptyset$ for all m. This clearly implies van der Waerden's theorem since $\mathbb{N} = C_1 \cup \cdots \cup C_r \Rightarrow \max_i \overline{\delta}(C_i) \geq 1/r$.

Furstenberg [Fur77] has given a quite different proof of Szemerédi's theorem, using tools from ergodic theory and topological dynamics. This approach has proved to be very powerful, allowing Furstenberg, Katznelson, and others to prove density versions of the Hales-Jewett theorem (see [FK91]), the Gallai-Witt theorem, and many others.

There are other ways of expressing the fact that A is relatively dense in $\mathbb{N}$ besides the condition that $\overline{\delta}(A) > 0$. One would expect that these could also be used as a basis for a density version of van der Waerden or Gallai-Witt. Very little is currently known in this direction, however. We conclude this section with several conjectures of this type.

CONJECTURE 8.4.5 (Erdős)
If $A \subseteq \mathbb{N}$ satisfies $\sum_{a \in A} 1/a = \infty$ then A contains arbitrarily long arithmetic progressions.

CONJECTURE 8.4.6 (Graham)
If $A \subseteq \mathbb{N} \times \mathbb{N}$ with $\sum_{(x,y) \in A} 1/(x^2 + y^2) = \infty$ then A contains the 4 vertices of an axes-parallel square.

More generally, I expect that A will always contain a homothetic image of $\{1, 2, \ldots, m\} \times \{1, 2, \ldots, m\}$ for all m.

Finally, we mention a direction in which the group $SO(n)$ is enlarged to allow dilatations as well.

Definition: For a set $W \subseteq \mathbb{E}^k$, define the ***upper density*** $\overline{\delta}(W)$ of W by

$$\overline{\delta}(W) := \limsup_{R \longrightarrow \infty} \frac{m(B(o,R) \cap W)}{m(B(o,R))},$$

where $B(o,R)$ denotes the k-ball $\left\{ (x_1, \ldots, x_k) \in \mathbb{E}^k \;\middle|\; \sum_{i=1}^{k} x_i^2 \leq R^2 \right\}$ centered at the origin, and m denotes Lebesgue measure.

THEOREM 8.4.7 (Bourgain [Bou86])
Let $X \subseteq \mathbb{E}^k$ be a simplex. If $W \subseteq \mathbb{E}^k$ with $\overline{\delta}(W) > 0$ then there exists t_0 such that for all $t > t_0$, W contains a congruent copy of tX.

Some restrictions on X are necessary as the following result shows.

THEOREM 8.4.8 (Graham [Gra94])
Let $X \subseteq \mathbb{E}^k$ be nonspherical. Then for any N there exist a set $W \subseteq \mathbb{E}^N$ with $\overline{\delta}(W) > 0$ and a set $T \subseteq \mathbb{R}$ with $\underline{\delta}(T) > 0$ such that W contains no congruent copy of tX for any $t \in T$.

Here $\underline{\delta}$ denotes ***lower density***, defined similarly to $\overline{\delta}$ but with lim inf replacing lim sup.

It is clear that much remains to be done here.

8.5 VARIATIONS

There are quite a few variants of the preceding topics that have received attention in the literature (e.g., see [Sch93]). We mention some of the more interesting ones.

ASYMMETRIC RAMSEY THEOREMS

Typical results of this type assert that for given sets X_1 and X_2 (for example), for every partition of $\mathbb{E}^N = C_1 \cup C_2$, either C_1 contains a congruent copy of X_1, or C_2 contains a congruent copy of X_2. We can denote this by

$$\mathbb{E}^N \xrightarrow{2} (X_1, X_2).$$

Here is a sampling of results of this type (more of which can be found in [EGM+73], [EGM+75a], [EGM+75b]).

(i) $\mathbb{E}^2 \xrightarrow{2} (T_2, T_3)$ where T_i is any subset of $\mathbb{E}^2$ with i points, $i = 2, 3$.

(ii) $\mathbb{E}^2 \xrightarrow{2} (P_2, P_4)$ where P_2 is a set of two points at a distance 1, and P_4 is a set of four collinear points with distance 1 between consecutive points.

(iii) $\mathbb{E}^3 \xrightarrow{2} (T, Q^2)$ where T is an isoceles right triangle and Q^2 is a square.

(iv) $\mathbb{E}^2 \xrightarrow{2} (P_2, T_4)$ where P_2 is as in (ii) and T_4 is any set of four points [Juh79].

(v) There is a set T_8 of 8 points such that

$$\mathbb{E}^2 \overset{2}{\not\longrightarrow} (P_2, T_8) \quad \text{[CT94]}.$$

This strengthens an earlier result of Juhász [Juh79], which proved this for a certain set of 12 points.

POLYCHROMATIC RAMSEY THEOREMS

Here, instead of asking for a copy of the target set X in a single C_i, we require only that it be contained in the union of a small number of C_i, say at most m of the C_i.

Let us indicate this by writing $\mathbb{E}^N \underset{m}{\rightarrow} X$.

(i) If $\mathbb{E}^N \underset{m}{\rightarrow} X$ then X must be embeddable on the union of m concentric spheres [EGM+73].

(ii) Suppose X_i is finite and $\mathbb{E}^N \underset{m_i}{\longrightarrow} X_i$, $1 \le i \le t$. Then

$$\mathbb{E}^N \xrightarrow[m_1 m_2 \cdots m_t]{} X_1 \times X_2 \times \cdots \times X_t \qquad \text{[ERS83]}.$$

(iii) If X_6 is the 6-point set formed by taking the four vertices of a square together with the midpoints of two adjacent sides then $\mathbb{E}^2 \not\rightarrow X_6$ but $\mathbb{E}^2 \underset{2}{\longrightarrow} X_6$.

(iv) If X is the set of vertices of a regular simplex in $\mathbb{E}^N$ together with the trisection points of each of its edges then

$$\mathbb{E}^2 \not\rightarrow X_6 \quad \text{but} \quad \mathbb{E}^2 \underset{3}{\longrightarrow} X_6.$$

It is not known if $\mathbb{E}^2 \underset{2}{\longrightarrow} X_6$. Many other results of this type can be found in [ERS83].

PARTITIONS OF $\mathbb{E}^n$ WITH ARBITRARILY MANY PARTS

Since $\mathbb{E}^2 \overset{7}{\not\longrightarrow} P_2$, where P_2 is a set of two points with unit distance, one might ask whether there is any nontrivial result of the type $\mathbb{E}^2 \overset{m}{\longrightarrow} \mathcal{F}$ when m is allowed to go to infinity. Of course, if $\mathcal{F}$ is sufficiently large, then there certainly are. There are some interesting geometric examples for which $\mathcal{F}$ is not too large.

THEOREM 8.5.1 [Gra80a]

For any partition of $\mathbb{E}^n$ into finitely many parts, some part contains, for all $\alpha > 0$ and all sets of lines $L_1, \ldots, L_n$ that span $\mathbb{E}^n$, a simplex having volume α and edges through one vertex parallel to the L_i.

Many other theorems of this type are possible (see [Gra80a]).

PARTITIONS WITH INFINITELY MANY PARTS

Results of this type tend to have a strong set-theoretic flavor. For example: $\mathbb{E}^2 \overset{\aleph_0}{\not\longrightarrow} T_3$ where T_3 is an equilateral triangle [Ced69]. In other words, $\mathbb{E}^2$ can be partitioned into countably many parts so that no part contains the vertices of an equilateral triangle. In fact, this was very recently strengthened by Schmerl [Sch94b] who showed that for all N,

$$\mathbb{E}^N \overset{\aleph_0}{\not\longrightarrow} T_3.$$

In fact, this result holds for *any* fixed triangle T in place of T_3 [Sch94b]. Schmerl also has shown [Sch94a] that there is a partition of $\mathbb{E}^N$ into countably many parts such that no part contains the vertices of *any* isoceles triangle.

Another result of this type is this:

THEOREM 8.5.2 [Kun]

Assuming the Continuum Hypothesis, it is possible to partition $\mathbb{E}^2$ into countably many parts, none of which contains the vertices of a triangle with **rational** *area.*

COMPLEXITY ISSUES

S. Burr [Bur82] has shown that the algorithmic question of deciding if a given set $X \subset \mathbb{N} \times \mathbb{N}$ can be partitioned $X = C_1 \cup C_2 \cup C_3$ so that $x, y \in C_i \Rightarrow \text{distance}(x, y) \geq 6$, $i = 1, 2, 3$, is NP-complete. (Also, he shows that a certain infinite version of this is undecidable.)

Finally, we make a few remarks about the celebrated problem of Esther Klein (who became Mrs. Szekeres), which, in some sense, initiated this whole area (see [Sze73] for a charming history).

THEOREM 8.5.3 [ES35]

There is a minimum function $f : \mathbb{N} \longrightarrow \mathbb{N}$ such that any set of $f(n)$ points in $\mathbb{E}^2$ in general position contains the vertices of a convex n-gon.

This result of Erdős and George Szekeres actually spawned an independent genesis of Ramsey theory.

The best bounds currently known for $f(n)$ are:

$$2^{n-2} + 1 \leq f(n) \leq \binom{2n-4}{n-2} + 1.$$

CONJECTURE 8.5.4

Prove (or disprove) that $f(n) = 2^{n-2} + 1$, $n \geq 3$.

(See Chapter 1 of this Handbook for more details.)

8.6 SOURCES AND RELATED MATERIAL

SURVEYS

The principal surveys for results in Euclidean Ramsey theory are [GRS90], [Gra80b], [Gra85], and [Gra94]. The first of these is a monograph on Ramsey theory in general, with a section devoted to Euclidean Ramsey theory, while the last three are specifically about the topics discussed in the present chapter.

RELATED CHAPTERS

Chapter 1: Finite point configurations
Chapter 10: Geometric discrepancy theory and uniform distribution

REFERENCES

[Bón93] M. Bóna. A Euclidean Ramsey theorem. *Discrete Math.*, 122:349–352, 1993.

[Bou86] J. Bourgain. A Szemerédi type theorem for sets of positive density in $\mathbb{R}^k$. *Israel J. Math.*, 54:307–316, 1986.

[BT96] M. Bóna and G. Tóth. A Ramsey-type problem on right-angled triangles in space. *Discrete Math.*, 150:61–76, 1996.

[Bur82] S.A. Burr. An NP-complete problem in Euclidean Ramsey theory. In *Proc. 13th Southeastern Conf. on Combinatorics, Graph Theory and Computing*, volume 35, pages 131–138, 1982.

[Can96] K. Cantwell. Finite Euclidean Ramsey theory. *J. Combin. Theory Ser. A*, 73:273–285, 1996.

[Ced69] J. Ceder. Finite subsets and countable decompositions of Euclidean spaces. *Rev. Roumaine Math. Pures Appl.*, 14:1247–1251, 1969.

[CFG91] H.T. Croft, K.J. Falconer, and R.K. Guy. *Unsolved Problems in Geometry*. Springer-Verlag, New York, 1991.

[CT94] G. Csizmadia and G. Tóth. Note on a Ramsey-type problem in geometry. *J. Combin. Theory Ser. A*, 65:302–306, 1994.

[EGM+73] P. Erdős, R.L. Graham, P. Montgomery, B.L. Rothschild, J.H. Spencer, and E.G. Straus. Euclidean Ramsey theorems. *J. Combin. Theory Ser. A*, 14:341–63, 1973.

[EGM+75a] P. Erdős, R.L. Graham, P. Montgomery, B.L. Rothschild, J.H. Spencer, and E.G. Straus. Euclidean Ramsey theorems II. In A. Hajnal, R. Rado, and V. Sós, editors, *Infinite and Finite Sets I*, pages 529–557. North-Holland, Amsterdam, 1975.

[EGM+75b] P. Erdős, R.L. Graham, P. Montgomery, B.L. Rothschild, J.H. Spencer, and E.G. Straus. Euclidean Ramsey theorems III. In A. Hajnal, R. Rado, and V. Sós, editors, *Infinite and Finite Sets II*, pages 559–583. North-Holland, Amsterdam, 1975.

[ERS83] P. Erdős, B. Rothschild, and E.G. Straus. Polychromatic Euclidean Ramsey theorems. *J. Geom.*, 20:28–35, 1983.

[ES35] P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio Math.*, 2:463–470, 1935.

[FK91] H. Furstenberg and Y. Katznelson. A density version of the Hales-Jewett theorem. *J. Anal. Math.*, 57:64–119, 1991.

[FR90] P. Frankl and V. Rödl. A partition property of simplices in Euclidean space. *J. Amer. Math. Soc.*, 3:1–7, 1990.

[Fur77] H. Furstenberg. Ergodic behavior of diagonal measures and a theorem of Szemerédi on arithmetic progressions. *J. d'Anal. Math.*, 31:204–256, 1977.

[FW81] P. Frankl and R.M. Wilson. Intersection theorems with geometric consequences. *Combinatorica*, 1:357–368, 1981.

[Gra80a] R.L. Graham. On partitions of $\mathbb{E}^n$. *J. Combin. Theory Ser. A*, 28:89–97, 1980.

[Gra80b] R.L. Graham. Topics in Euclidean Ramsey theory. In J. Nešetřil and V. Rödl, editors, *Mathematics of Ramsey Theory*. Springer-Verlag, Heidelberg, 1980.

[Gra83] R.L. Graham. Euclidean Ramsey theorems on the n-sphere. *J. Graph Theory*, 7:105–114, 1983.

[Gra85] R.L. Graham. Old and new Euclidean Ramsey theorems. In J.E. Goodman, E. Lutwak, J. Malkevitch, and R. Pollack, editors, *Discrete Geometry and Convexity*, volume 440, Ann. New York Acad. Sci., pages 20–30. New York, 1985.

[Gra94] R.L. Graham. Recent trends in Euclidean Ramsey theory. *Discrete Math.*, 136:119–127, 1994.

[GRS90] R.L. Graham, B.L. Rothschild, and J.H. Spencer. *Ramsey Theory*. John Wiley and Sons, New York, 2nd edition, 1990.

[Juh79] R. Juhász. Ramsey type theorems in the plane. *J. Combin. Theory Ser. A*, 27:152–160, 1979.

[Kři91] I. Kříž. Permutation groups in Euclidean Ramsey theory. *Proc. Amer. Math. Soc.*, 112:899–907, 1991.

[Kři92] I. Kříž. All trapezoids are Ramsey. *Discrete Math*, 108:59–62, 1992.

[Kun] K. Kunen. Personal communication.

[MR95] J. Matoušek and V. Rödl. On Ramsey sets on spheres. *J. Combin. Theory Ser. A*, 70:30–44, 1995.

[Sch93] P. Schmitt. Problems in discrete and combinatorial geometry. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, volume A. North-Holland, Amsterdam, 1993.

[Sch94a] J.H. Schmerl. Personal communication, 1994.

[Sch94b] J.H. Schmerl. Triangle-free partitions of Euclidean space. *Bull. London Math. Soc.*, 26:483–486, 1994.

[Sha76] L. Shader. All right triangles are Ramsey in $\mathbb{E}^2$! *J. Combin. Theory Ser. A*, 20:385–389, 1976.

[She88] S. Shelah. Primitive recursive bounds for van der Waerden numbers. *J. Amer. Math. Soc.*, 1:683–697, 1988.

[Soi91] A. Soifer. Chromatic number of the plane: A historical survey. *Geombinatorics*, 1:13–14, 1991.

[Soi92] A. Soifer. A six-coloring of the plane. *J. Combin. Theory Ser. A*, 61:292–294, 1992.

[Sze73] G. Szekeres. A combinatorial problem in geometry: Reminiscences. In J. Spencer, editor, *Paul Erdős: The Art of Counting, Selected Writings*, pages xix–xxii. The MIT Press, Cambridge, 1973.

[Sze75] E. Szemerédi. On sets of integers containing no k elements in arithmetic progression. *Acta Arith.*, 27:199–245, 1975.

[Tót96] G. Tóth. A Ramsey-type bound for rectangles. *J. Graph Theory*, 23:53–56, 1996.

[van27] B.L. van der Waerden. Beweis einer Baudetschen Vermutung. *Nieuw Arch. Wisk.*, 15:212–216, 1927.

9 DISCRETE ASPECTS OF STOCHASTIC GEOMETRY

Rolf Schneider

INTRODUCTION

Stochastic geometry studies randomly generated geometric objects. The present chapter is restricted to discrete aspects of stochastic geometry. We describe work that has been done on familiar objects of discrete geometry, in particular finite or discrete point sets, but also arrangements of flats, random congruent bodies, or tessellations, under various assumptions of randomness. The emphasis will be on finite point sets. Most of the results to be mentioned concern expectations of geometrically defined random variables or probabilities of events defined by random geometric configurations. The determination of whole probability distributions of geometric random variables is mostly out of reach, and the few cases where this can be achieved may be of only peripheral interest from the viewpoint of discrete geometry.

9.1 RANDOM POINTS

The setup for most of this section is a finite number of random points in a topological space S. Often the space S is $\mathbb{R}^d$, the d-dimensional Euclidean space, with scalar product $\langle \cdot, \cdot \rangle$ and norm $\| \cdot \|$. Other spaces that occur are the sphere $S^{d-1} := \{x \in \mathbb{R}^d \,|\, \|x\| = 1\}$ or more general submanifolds of $\mathbb{R}^d$. By $B^d := \{x \in \mathbb{R}^d \,|\, \|x\| \leq 1\}$ we denote the unit ball of $\mathbb{R}^d$. The volume of B^d is denoted by κ_d.

GLOSSARY

Random point in S: A Borel measurable mapping from some probability space into S.

Distribution of a random point X in S: The probability measure μ on S such that $\mu(B)$, for a Borel set $B \subset S$, is the probability that $X \in B$.

i.i.d. random points: Stochastically independent random points (on the same probability space) with the same distribution.

9.1.1 NATURAL DISTRIBUTIONS

In geometric problems about random points, a few distributions have been considered as particularly natural, for different reasons. Such reasons may be invariance properties, or relations to measures of geometric significance, but there are also more subtle viewpoints, as explained, for example, in Section 9.1.5 or in Ruben

and Miles [RM80]. The distributions of a random point in $\mathbb{R}^d$ shown in Table 9.1.1 underlie many investigations.

TABLE 9.1.1 Natural distributions of a random point in $\mathbb{R}^d$.

NAME OF DISTRIBUTION	PROBABILITY DENSITY AT $x \in \mathbb{R}^d$
Uniform in K	$\propto$ indicator function of K
Standard normal	$\propto \exp\left(-\frac{1}{2}\|x\|^2\right)$
Beta type 1	$\propto (1-\|x\|^2)^q \times$ indicator function of B^d, $q > -1$
Beta type 2	$\propto \|x\|^{\alpha-1}(1+\|x\|)^{-(\alpha+\beta)}, \alpha, \beta > 0$
Spherically symmetric	function of $\|x\|$

Here $K \subset \mathbb{R}^d$ is a given closed set of positive, finite volume, often a convex body. Usually the name of the distribution of a random point is also associated with the random point itself. General rotationally symmetric distributions have mostly been considered under additional tail assumptions. If F is a smooth compact hypersurface in $\mathbb{R}^d$, a random point is uniform on F if its distribution is proportional to the area measure on F. This distribution is particularly natural for the unit sphere S^{d-1}, since it is the unique rotation-invariant probability measure on S^{d-1}.

For combinatorial problems about n-tuples of random points in $\mathbb{R}^d$, the following approach leads to a natural distribution. Every configuration of n numbered points in general position in $\mathbb{R}^d$ is affinely equivalent to the orthogonal projection of the set of numbered vertices of a fixed regular simplex $T^{n-1} \subset \mathbb{R}^{n-1}$ onto a unique d-dimensional linear subspace of $\mathbb{R}^{n-1}$. This establishes a one-to-one correspondence between the (orientation-preserving) affine equivalence classes of such configurations and an open dense subset of the Grassmannian $G(n-1,d)$ of oriented d-spaces in $\mathbb{R}^{n-1}$. The unique rotation-invariant probability measure on $G(n-1,d)$ thus leads to a probability distribution on the set of affine equivalence classes of n-tuples of points in general position in $\mathbb{R}^d$. References for this ***Grassmann approach***, which was proposed by Vershik and by Goodman and Pollack, are given in Affentranger and Schneider [AS92]. Baryshnikov and Vitale [BV94] proved that an affine-invariant functional of n-tuples with this distribution is stochastically equivalent to the same functional taken at an i.i.d. n-tuple of standard normal points in $\mathbb{R}^d$.

9.1.2 CONVEX HULLS OF UNIFORM RANDOM POINTS

A great deal of work has been done on convex hulls of a finite number of i.i.d. random points in $\mathbb{R}^d$. We consider this topic first for the case of uniform distributions.

NOTATION

$X_1, \ldots, X_n$	i.i.d. random points in $\mathbb{R}^d$
μ	the common probability distribution of X_i
φ	a measurable real function defined on polytopes in $\mathbb{R}^d$
$\varphi(\mu, n)$	the random variable $\varphi(\text{conv}\{X_1, \ldots, X_n\})$
$\varphi(K, n)$	$= \varphi(\mu, n)$, if μ is the uniform distribution in K

$\mathbb{E}$	expectation of a random variable
f_k	number of k-faces
ψ_j	1 on polytopes with j vertices, 0 otherwise
V_j	jth intrinsic volume (see Chapter 13); in particular:
V_d	d-dimensional volume
S	surface area

For a convex body $K \subset \mathbb{R}^d$, some of the expectations of $\varphi(K,n)$ for different functions φ listed above are connected by identities. For $n \geq d+1$ it is easy to see that

$$\mathbb{E}\psi_{d+1}(K,n) = \binom{n}{d+1} V_d(K)^{d+1-n} \mathbb{E}V_d^{n-d-1}(K, d+1), \tag{9.1.1}$$

and

$$\mathbb{E}f_0(K,n) = \frac{n}{V_d(K)}\big(V_d(K) - \mathbb{E}V_d(K, n-1)\big) \tag{9.1.2}$$

(Efron 1965, see [Sch88], [WW93]). The latter equation is one reason for considering combinatorial and metric invariants simultaneously in this section. For arbitrary distributions μ on $\mathbb{R}^d$, Buchta [Buc90] showed that

$$\mathbb{E}V_d(\mu, d+2m) = \sum_{k=1}^{m} (2^{2k}-1)\frac{B_{2k}}{k}\binom{d+2m}{2k-1}\mathbb{E}V_d(\mu, d+2m-2k+1)$$

for $m \in \mathbb{N}$, where the constants B_{2k} are the Bernoulli numbers.

About the random variables $\varphi(K,n)$, we first mention the rare instances where information on the whole distribution is available. Some special results for $d=2$ due to Alagar, Reed, and Henze are quoted in [Sch88, Section 4]. For example, Henze showed that the distribution function F_K of $V_2(K,3)$ for a convex body $K \subset \mathbb{R}^2$ satisfies $F_T \leq F_K \leq F_E$, where T is a triangle and E is an ellipse, provided that K, T, E have the same area. Results on the distribution of $V_r(B^d, r+1)$ for $r = 1, \ldots, d$ are listed in a more general context in Section 9.1.3. In the plane, a few remarkable central limit type theorems have been obtained. For a convex polygon $P \subset \mathbb{R}^2$ with k vertices, Groeneboom (1988, see [WW93]) proved that

$$\frac{f_0(P,n) - (2/3)k\log n}{\sqrt{(10/27)k\log n}} \xrightarrow{\mathcal{D}} \mathcal{N}(0,1)$$

for $n \to \infty$, where $\xrightarrow{\mathcal{D}}$ denotes convergence in distribution and $\mathcal{N}(0,1)$ is the standard normal distribution. For the circular disk, Groeneboom showed

$$\frac{f_0(B^2,n) - 2\pi a_1 n^{1/3}}{\sqrt{2\pi a_2 n^{1/3}}} \xrightarrow{\mathcal{D}} \mathcal{N}(0,1)$$

with explicitly given constants a_1, a_2. Similar results are known for the remaining area $A(K,n) := V_2(K) - V_2(K,n)$. For a polygon P with k vertices, Cabo and Groeneboom [CG94] obtained

$$\frac{A(P,n) - \frac{2}{3}\frac{\log n}{n}}{\sqrt{\frac{100}{189}k\frac{\log n}{n}}} \xrightarrow{\mathcal{D}} \mathcal{N}(0,1).$$

For the circular disk, Hsing [Hsi94] proved that the variance of $A(B^2,n)$ satisfies $\operatorname{var}(A(B^2,n)) \sim \sigma^2 n^{-5/3}$ for $n \to \infty$ with a number $\sigma^2 < \infty$ and that

$$n^{5/6}\big(A(B^2,n) - \mathbb{E}A(B^2,n)\big) \xrightarrow{\mathcal{D}} N(0,\sigma^2).$$

Küfer [Küf94] investigated the asymptotic behavior of $V_d(B^d) - V_d(B^d, n)$, in particular the second moment.

Most of the known results about the random variables $\varphi(K, n)$ concern their expectations. Explicit formulas for $\mathbb{E}\,\varphi(K, n)$ for convex bodies $K \subset \mathbb{R}^d$ and arbitrary $n \geq d + 1$ are known in the cases listed in Table 9.1.2 (for references, see [Sch88, WW93]).

TABLE 9.1.2 Expected value of $\varphi(K, n)$.

DIMENSION d	CONVEX BODY K	FUNCTIONAL φ	SOURCES
2	polygon	V_2	Buchta
2	ellipse	V_2	Buchta
3	ellipsoid	V_3	Buchta
≥ 2	ball	S, mean width, f_{d-1}	Buchta and Müller
≥ 2	ball	V_d	Affentranger

Affentranger's result is given in the form of an integral, which can be evaluated for given d and n; it implies the corresponding result for ellipsoids.

A well-known problem, popularized by Klee, is the explicit determination of $\mathbb{E}V_d(T^d, d + 1)$ for a d-simplex T^d. Klee's opinion that $\mathbb{E}V_3(T^3, 4)$ "might yield to brute force" was justified. The result

$$\mathbb{E}V_3(T^3, 4) = \frac{13}{720} - \frac{\pi^2}{15015} = 0.0173982\ldots \tag{9.1.3}$$

was announced by Buchta and Reitzner [BR93], as well as a more general formula for $\mathbb{E}V_3(T^3, n)$. Independently, (9.1.3) was established by Mannion [Man94], who made heavy use of computer algebra.

If explicit formulas for $\mathbb{E}\,\varphi(K, n)$ are not available, one can try to obtain inequalities or asymptotic results for increasing n. For $\mathbb{E}V_d(K, n)$, the following estimates are known. The quotient $\mathbb{E}V_d(K, n)/V_d(K)$, for $n \geq d + 1$, is minimal for ellipsoids. The conjecture that it is maximal for simplices is only proved for $d = 2$. If the convex body $K \subset \mathbb{R}^d$ is not a simplex, then the quotient $\mathbb{E}V_d(K, n)/V_d(K)$ is strictly less than its value for a simplex, for all $n \geq n_0(K)$ (references for these and related results are given in the survey part of [BS95]).

We turn to asymptotic results. Buchta (1984, see [Sch88]) proved for plane polygons P and the perimeter $2V_1$ that

$$V_1(P) - \mathbb{E}V_1(P, n) = c(P)\left(\frac{n}{V_2(P)}\right)^{-1/2} + o(n^{\epsilon-1})$$

for any fixed $\epsilon > 0$, where the constant $c(P)$ is given explicitly in terms of the angles of P. Work of Rényi and Sulanke for the plane is described in [Sch88, Section 5], as well as some particular results for $\mathbb{R}^d$, in part superseded by the following ones. For d-dimensional polytopes P, Bárány and Buchta [BB93] were able to show that

$$\mathbb{E}f_0(P, n) = \frac{T(P)}{(d+1)^{d-1}(d-1)!}\log^{d-1} n + O(\log^{d-2} n \log\log n), \tag{9.1.4}$$

where $T(P)$ denotes the number of chains $F_0 \subset F_1 \subset \ldots \subset F_{d-1}$ where F_i is an i-dimensional face of P. They establish a corresponding relation for the volume,

from which (9.1.4) follows by (9.1.2). This work was the culmination of a series of papers by other authors, among them Affentranger and Wieacker, who settled the case of simple polytopes, which is applied in [BB93]. Bárány and Buchta mention that their methods permit one to extend (9.1.4) to $\mathbb{E}f_k(P,n)$ for $k=0,\dots,d-1$, with the denominator replaced by a constant depending on d and k.

For convex bodies $K\subset\mathbb{R}^d$ with a boundary of class C^3 and positive Gauss-Kronecker curvature κ, Bárány [Bár92] established

$$\begin{aligned} & V_d(K)-\mathbb{E}V_d(K,n)\\ = \; & c(d)\int\limits_{\partial K}\kappa^{1/(d+1)}\,dS\left(\frac{n}{V_d(K)}\right)^{-2/(d+1)}+O(n^{-3/(d+1)}\log^2 n) \end{aligned} \tag{9.1.5}$$

with a constant $c(d)$, and for the intrinsic volumes the relation

$$\begin{aligned} & V_j(K)-\mathbb{E}V_j(K,n)\\ = \; & c(d,j)\int\limits_{\partial K}\kappa^{[(d-j)/(d-1)]+[1/(d+1)]}\,dS\left(\frac{n}{V_d(K)}\right)^{-2/(d+1)}(1+O(1)). \end{aligned} \tag{9.1.6}$$

Relation (9.1.5) was extended (as a limit without an O-term) by Schütt [Sch94] to arbitrary convex bodies, with the Gauss-Kronecker curvature generalized accordingly. For general convex bodies K, the asymptotic behavior of $\mathbb{E}V_d(K,n)$ was also investigated by Bárány and Larman (see [WW93]). Similar results for $\mathbb{E}V_j(K,n)$ and $\mathbb{E}f_k(K,n)$ were obtained by Bárány [Bár89]. The principal idea of Bárány and Larman was to compare the volume of the convex hull, say P_n, of n independent uniform random points in K, with the volume of a certain kind of floating body, K_n, derived from K. Bárány and Vitale [BV93] show that the set-valued expectation of the random polytope P_n itself is close to K_n.

Of combinatorial interest is the expectation $\mathbb{E}\psi_i(\mu,n)$, which is the probability that the convex hull of n i.i.d. random points with distribution μ has exactly i vertices. Sylvester's classical problem asked for $\mathbb{E}\psi_3(K,4)$ (or the complementary probability) for a convex body $K\subset\mathbb{R}^2$. More generally, one may ask for $\mathbb{E}\psi_{d+1}(K,n)$ for a convex body $K\subset\mathbb{R}^d$ and $n>d+1$, the probability that the convex hull of n uniform i.i.d. points in K is a simplex. By (9.1.1), this is related to the moments of $V_d(K,d+1)$ and hence, in particular, explicitly known if K is a ball (by Miles 1971, see [Sch88], [SW93], [WW93]). At the other end, $\mathbb{E}\psi_n(K,n)$ is of interest, the probability that n uniform i.i.d. points in K are "in convex position." Valtr [Val95] proved by purely combinatorial means that, for a parallelogram $P\subset\mathbb{R}^2$,

$$\mathbb{E}\psi_n(P,n)=\left(\frac{\binom{2n-2}{n-1}}{n!}\right)^2,$$

and in [Val96] he obtained a similar result for triangles. Generalizing earlier work of Buchta, it was proved by Bárány and Füredi (1988, see [WW93]) that

$$\begin{aligned} \mathbb{E}\psi_{n(d)}(B^d,n(d))&\to 1 \qquad \text{if} \quad n(d)=2^{d/2}d^{-\epsilon},\\ \mathbb{E}\psi_{m(d)}(B^d,m(d))&\to 0 \qquad \text{if} \quad m(d)=2^{d/2}d^{(3/4)+\epsilon} \end{aligned}$$

when $d\to\infty$, for every fixed $\epsilon>0$. The authors also investigated k-neighborliness of the convex hull.

In order to obtain the convex hull of n i.i.d. uniform points in a circular disk, only the points in a narrow strip about the boundary are needed, with high probability, if n is large. This was made precise by Carnal and Hüsler [CH91].

OPEN PROBLEMS

PROBLEM 9.1.1

For a convex body $K \subset \mathbb{R}^d$ with a boundary of class C^3 and positive Gauss-Kronecker curvature κ, and for the numbers of k-faces one expects that

$$\mathbb{E}f_k(K,n) = b(d,k) \int_{\partial K} \kappa^{1/(d+1)}\, dS \left(\frac{n}{V_d(K)}\right)^{(d-1)/(d+1)} (1+o(1)) \qquad (9.1.7)$$

with a constant $b(d,k)$. For $k=0$, this follows from (9.1.6); for $k=d-1$ (which implies the case $k=d-2$) the result goes back to Raynaud and Wieacker; see [Bár92] and [Sch88, p. 222] for references.

PROBLEM 9.1.2 (I. Bárány)

Is it true for a general convex body $K \subset \mathbb{R}^d$ that the surface area S satisfies

$$c_1(K)n^{-1/2} < S(K) - \mathbb{E}S(K,n) < c_2(K)n^{-2/(d+1)}$$

with positive constants $c_1(K), c_2(K)$?

PROBLEM 9.1.3 (P. Valtr [Val96])

Is it true, for a convex body $K \subset \mathbb{R}^2$ and for $n \geq 4$, that $\mathbb{E}\psi_n(K,n)$, the probability that n uniform i.i.d. points in K are in convex position, is minimal if K is a triangle and maximal if K is an ellipse?

9.1.3 CONVEX HULLS FOR OTHER DISTRIBUTIONS

Convex hulls of nonuniform i.i.d. random points have been investigated for each of the distributions mentioned in Section 9.1.1, and occasionally for more general ones. The following setup has been studied repeatedly. For $0 \leq p \leq r+1 \leq d-1$, one considers $r+1$ independent random points, of which the first p are uniform in the ball B^d and the last $r+1-p$ are uniform on the boundary sphere S^{d-1}. Precise information on the moments and the distribution of the r-dimensional volume of the convex hull is available; see the references in [Sch88, pp. 219, 224] and the work of Affentranger (1988, see [WW93]).

Among spherically symmetric distributions, the beta distributions are particularly tractable. For these, again, the r-dimensional volume of the convex hull of $r+1$ i.i.d. random points has frequently been studied. We refer to the references given in [Sch88] and Chu [Chu93]. Affentranger (1991, see [WW93]) determined the asymptotic behavior, for $n \to \infty$, of the expectation $\mathbb{E}V_j(\mu,n)$, where μ is either the beta type-1 distribution, the uniform distribution in B^d, or the standard normal distribution in $\mathbb{R}^d$. Also the asymptotic behavior of $\mathbb{E}f_{d-1}(\mu,n)$ was found for these cases.

For normally distributed points in the plane, Hueter [Hue94] obtained central

limit type results. Let μ_d denote the standard normal distribution in $\mathbb{R}^d$. Then

$$\frac{f_0(\mu_2,n)-2\sqrt{2\pi\log n}}{\left(2\sqrt{2\pi\log n}(1+c_2)\right)^{1/2}} \xrightarrow{\mathcal{D}} \mathcal{N}(0,1)$$

as $n\to\infty$, with some constant c_2, and similar results hold for the perimeter and the area of the convex hull. In higher dimensions, one has asymptotic results for expectations, for example for any given integers $0\le k<d\le n-1$ the relation

$$\mathbb{E}f_k(\mu_d,n)\sim\frac{2^d}{\sqrt{d}}\binom{d}{k+1}\beta_{k,d-1}(\pi\log n)^{(d-1)/2}$$

as $n\to\infty$, where $\beta_{k,d-1}$ is the interior angle of the regular $(d-1)$-dimensional simplex at one of its k-dimensional faces. This follows from [AS92], where the Grassmann approach was used, due to the equivalence of [BV94] mentioned in Section 9.1.1. For the Grassmann approach, Vershik and Sporyshev [VS92] made a careful study of the asymptotic behavior of the number of k-faces, if k and the dimension d grow linearly with the number n.

For more general spherically symmetric distributions μ, the asymptotic behavior of the random variables $\varphi(\mu,n)$ will essentially depend on the tail behavior of the distribution. Extending work of Carnal (1970, see [Sch88]), Dwyer [Dwy91] obtained asymptotic estimates for $\mathbb{E}f_0(\mu,n)$, $\mathbb{E}f_{d-1}(\mu,n)$, $\mathbb{E}V_n(\mu,n)$, and $\mathbb{E}S(\mu,n)$. Aldous *et al.* [AFGP91] considered an i.i.d. sequence $(X_k)_{k\in\mathbb{N}}$ in $\mathbb{R}^2$ with a spherically symmetric (or more general) distribution. Under an assumption of slowly varying tail, they determined a limiting distribution for $f_0(\mu,n)$.

As a generalization of, or counterpart to, uniform random points in a convex body K, one may consider random points in K with a given density, or points on the boundary ∂K with a given density. The approximation of K by the convex hull of n i.i.d. points on ∂K is of particular interest. Some references are given in [Sch88, p. 224]. Affentranger (1991, see [WW93]) determined the asymptotic behavior of $V_j(B^d)-\mathbb{E}V_j(\mu,n)$, where μ is the uniform distribution on ∂B^d. For a convex body $K\subset\mathbb{R}^d$ with a boundary of class C^3 and positive Gauss-Kronecker curvature κ, the following approximation result in terms of the Hausdorff metric δ was proved by Glasauer and Schneider [GS94]. Let $(X_k)_{k\in\mathbb{N}}$ be an i.i.d. sequence of random points in ∂K, whose distribution has a continuous positive density h with respect to the surface area measure. Then

$$\underset{n\to\infty}{P\text{-}\lim}\left(\frac{n}{\log n}\right)^{2/(d-1)}\delta(K,\operatorname{conv}\{X_1,\dots,X_n\})=\frac{1}{2}\left(\frac{1}{\kappa_{d-1}}\max\frac{\sqrt{\kappa}}{h}\right)^{2/(d-1)},$$

where P-lim denotes stochastic convergence. For $d=2$, similar results hold with almost sure in place of stochastic convergence, and also with the Hausdorff distance replaced by area or perimeter difference; see [Sch88].

OPEN PROBLEM

PROBLEM 9.1.4

For a convex body $K\subset\mathbb{R}^d$ with a boundary of class C^3 and positive Gauss-Kronecker curvature, determine the exact asymptotic behavior, for $n\to\infty$, of $\mathbb{E}f_k(\mu,n)$ and $\mathbb{E}V_j(\mu,n)$, where μ is a distribution on ∂K with a positive continuous density. For $\mathbb{E}V_1(\mu,n)$, this was done by Müller (1989, see [WW93]).

9.1.4 GEOMETRIC CONFIGURATIONS

For a finite set of points, the relative position of its elements may be viewed under various geometric and combinatorial aspects. For randomly generated point sets, the probabilities of particular configurations may be of interest, but are in general hard to obtain. We list some contributions to problems of this type.

Bokowski *et al.* [BRS92] made a simulation study to estimate the probabilities of certain order types, using the Grassmann approach.

Related to k-sets (see Chapter 1 of this Handbook) is the following investigation of Bárány and Steiger [BS94]. If X is a set of n points in general position in $\mathbb{R}^d$, a subset $S \subset X$ of d points is called a k-simplex if X has exactly k points on one side of the affine hull of S. The authors study $E_d(k,n)$, the expected number of k-simplices for n i.i.d. random points. For continuous spherically symmetric distributions they show that

$$E_d(k,n) \leq c(d)n^{d-1}.$$

Further results concern the uniform distribution in a convex body in $\mathbb{R}^2$.

For a given distribution μ on $\mathbb{R}^2$, let $P_1,\ldots,P_j,Q_1,\ldots,Q_k$ be i.i.d. points distributed according to μ. Let $p_{jk}(\mu)$ be the probability that the convex hull of $P_1,\ldots,P_j$ is disjoint from the convex hull of $Q_1,\ldots,Q_k$. Continuing earlier work of L.C.C. Rogers, Buchta [Buc94] investigated $p_{jk}(\mu)$. For example, he obtained estimates for the case where μ is the uniform distribution in a circular disk, and deduced that

$$\lim_{n\to\infty} \frac{p_{nn}}{2\sqrt{\pi}n^{3/2}4^{-n}} = \frac{4}{3}$$

in this case. An explicit formula for p_{jk} is obtained for the uniform distribution in a triangle.

The following result of Wendel on random points on the unit sphere S^{d-1} has proved useful on several occasions; for a proof we refer to Mycielski [Myc87]. Let μ be an even probability measure on S^{d-1} such that every great subsphere has measure zero. Let $X_1,\ldots,X_n$ be i.i.d. random points on S^{d-1} with distribution μ. Then the probability p that $\{X_1,\ldots,X_i\}$ can be separated from $\{X_{i+1},\ldots,X_n\}$ by a hyperplane through 0 does not depend on i and is given by

$$p = \frac{1}{2^{n-1}} \sum_{k=0}^{d-1} \binom{n-1}{k}.$$

Various elementary geometric questions can be asked, even about a small number of random points. For example, if three uniform i.i.d. points in a convex body K are given, what is the probability that the triangle formed by them is obtuse, or what is the probability that the circle (almost surely) determined by these points is contained in K? Known results on probabilities of these types are listed in [BS95]. The following result is due to Affentranger. The probability that the sphere spanned (almost surely) by $d+1$ i.i.d. uniform random points in a convex body K is entirely contained in K attains its maximum precisely if K is a ball. In [BS95] it is shown that the probability that the circumball of $m \geq 2$ i.i.d. uniform points in K is contained in K is maximal if and only if K is a ball. The value of this maximum is $m/(2m-1)$ if $d = 2$, but is unknown for $d > 2$.

9.1.5 SHAPE

Two subsets of $\mathbb{R}^d$ may be said to have the same shape if they differ only by a similarity. D.G. Kendall's theory of shape yields natural probability distributions on shapes of labeled n-tuples of points in $\mathbb{R}^d$. The possible shapes of such n-tuples of points (not all coincident) can canonically be put in one-to-one correspondence with points of a certain topological space, and the resulting "shape spaces" carry natural probability measures. For this extensive theory and its statistical applications, we refer to the survey given by Kendall [Ken89].

A different approach to more general notions of shape and probability distributions for them is followed by Ambartzumian [Amb90]. He uses factorization of products of invariant measures to obtain corresponding probability densities, for example, for the affine shape of a tetrad of points in the plane.

9.1.6 POINT PROCESSES

The investigations described so far concerned finite systems of random points. For randomly generated infinite discrete point sets, suitable models are provided by stochastic point processes.

GLOSSARY

Locally finite: $M \subset \mathbb{R}^d$ is locally finite if $\operatorname{card}(M \cap B) < \infty$ for every compact set $B \subset \mathbb{R}^d$.

$\mathcal{M}$: The set of all locally finite subsets of $\mathbb{R}^d$.

M: The smallest σ-algebra on $\mathcal{M}$ for which every function $M \mapsto \operatorname{card}(M \cap B)$ is measurable, where $B \subset \mathbb{R}^d$ is a Borel set.

(Simple) point process X on $\mathbb{R}^d$: A measurable map X from some probability space (Ω, A, P) into $(\mathcal{M}, \mathsf{M})$.

Distribution of X: The image measure P_X of P under X.

Intensity measure Λ of X: $\Lambda(B) = \mathbb{E}\operatorname{card}(X \cap B)$, for Borel sets $B \subset \mathbb{R}^d$.

Stationary (or ***homogeneous***): X is a stationary point process if the distribution P_X is invariant under translations.

The point process X on $\mathbb{R}^d$, with intensity measure Λ (assumed to be finite on compact sets), is a ***Poisson process*** if, for any finitely many pairwise disjoint Borel sets $B_1, \ldots, B_k$, the random variables $\operatorname{card}(X \cap B_1), \ldots, \operatorname{card}(X \cap B_k)$ are independent and Poisson distributed. Thus, a Poisson point process X satisfies

$$\operatorname{Prob}\{\operatorname{card}(X \cap B) = k\} = e^{-\Lambda(B)} \frac{\Lambda(B)^k}{k!}$$

for $k \in \mathbb{N}_0$. If it is stationary, then the intensity measure Λ is a constant multiple of Lebesgue measure. Let X be a stationary Poisson process and $C \subset \mathbb{R}^d$ a compact set, and let $k \in \mathbb{N}_0$. Under the condition that exactly k points of the process fall into C, these points are equivalent to k i.i.d. uniform points in C. This fact clearly illustrates the geometric significance of stationary Poisson point processes, as does the following. Consider n i.i.d. uniform points in the ball rB^d. The Poisson process with intensity measure the Lebesgue measure can be considered as the limit process if n and r tend to infinity in such a way that $n/V_d(rB^d) \to 1$.

A detailed study of geometric properties of stationary Poisson processes in the plane was made by Miles [Mil70].

For much of the theory of point processes, the underlying space $\mathbb{R}^d$ can be replaced by a locally compact topological space S with a countable base. Of importance for stochastic geometry are, in particular, the cases where S is the space of r-flats in $\mathbb{R}^d$ (see Section 9.2.3) or the space of convex bodies in $\mathbb{R}^d$.

9.2 RANDOM FLATS

Next to random points, randomly generated r-dimensional flats in $\mathbb{R}^d$ are the most common object of study in stochastic geometry. Like convex hulls of random points, intersections of random halfspaces yield random polytopes in a natural way. Random flats through convex bodies as well as infinite arrangements of random hyperplanes give rise to a variety of questions.

9.2.1 RANDOM HYPERPLANES AND HALFSPACES

Intersections of random halfspaces appear as solution sets of systems of linear inequalities with random coefficients. Therefore, such random polyhedra play a role in the average case analysis of linear programming algorithms (see the book by Borgwardt [Bor87] and its bibliography). Under various assumptions on the distribution of the coefficients, one has information on the expected number of vertices of the solution sets (see [Sch88] for references). Extending earlier work of Prékopa, Buchta obtained several estimates, of which the following is an example.

Let $E(v)$ be the expected number of vertices of the polyhedron given by the inequalities $\sum_{i=1}^{n} a_{ij}x_j \le b\ (i = 1, \dots, m),\ x_j \ge 0\ (j = 1, \dots, n)$. If the coefficients a_{ij} are nonnegative and distributed independently, continuously, and symmetrically with respect to the same number $c > 0$, then

$$E(v) = \frac{1}{2^{m-1}}\binom{n}{m} + \frac{m}{2^{m-1}}\binom{n}{m-1} + O(n^{m-2})$$

for $n \to \infty$. Buchta also has formulas and estimates for $E(v)$ in the case of the polyhedron given by $\sum_{j=1}^{n} a_{ij}x_j \le 1\ (i = 1, \dots, m)$, where the points $(a_{i1}, \dots, a_{in})\ (i = 1, \dots, m)$ are i.i.d. uniform on the sphere S^{d-1}.

In a certain duality to convex hulls of random points in a convex body, one may consider intersections of halfspaces containing a convex body. Let $K \subset \mathbb{R}^d$ be a convex body with a boundary of class C^3 and with positive Gauss curvature κ; suppose that $0 \in \operatorname{int} K$ and let $r > 0$. Call a random closed halfspace $H_{u,t}^- := \{x \in \mathbb{R}^d \mid \langle x, u\rangle \le t\}$ with $u \in S^{d-1}$ and $t > 0$ "(K, r)-adapted" if the unit normal vector u is uniform on S^{d-1} and the distance t is independent of u and is, for given u, uniform in the interval for which $H_{u,t}^-$ contains K but not rB^d. Let $\mathbb{E}\tilde{V}_d(K, n)$ be the expected volume of the intersection of rB^d with n i.i.d. (K, r)-adapted random halfspaces. Then Kaltenbach [Kal90] proved that

$$\mathbb{E}\tilde{V}_d(K, n) - V_d(K) =$$
$$c_1(d) \int_{\partial K} \kappa^{1/(d+1)}\, dS \left(\frac{n}{V_1(rB^d) - V_1(K)}\right)^{-2/(d+1)} + O(n^{-3/(d+1)}) + O(r^d(1-\epsilon)^n)$$

for $n \to \infty$, where $0 < \epsilon < 1$ is fixed. This relation is very similar to (9.1.5), but not strictly dual to it.

Other results of Kaltenbach [Kal90] concern the intersection of n i.i.d. halfspaces of the form $H^-_{u,t}$, with u uniform on S^{d-1} and t distributed according to some distribution function F. The behavior of F for $t \to \infty$ strongly influences the behavior of the intersection of the halfspaces for $n \to \infty$.

9.2.2 RANDOM FLATS THROUGH CONVEX BODIES

The notion of uniform points in a convex body K in $\mathbb{R}^d$ is extended by that of a uniform random r-flat through K. Let $\mathcal{E}^d_r$ be the space of r-dimensional affine subspaces of $\mathbb{R}^d$ with the usual topology and Borel structure ($r \in \{0, \ldots, d-1\}$). A random r-flat is a measurable map from some probability space into $\mathcal{E}^d_r$. It is a ***uniform*** (***isotropic uniform***) random r-flat through K if its distribution can be obtained from a translation-invariant (resp. rigid-motion-invariant) measure on $\mathcal{E}^d_r$, by restricting it to the r-flats meeting K and normalizing to a probability measure. (For details, see [SW93, Example 6.2]; see also [WW93, Section 2].)

A random r-flat E (uniform or not) through K generates the random secant $E \cap K$, which has often been studied, particularly for $r = 1$. References are in [SW92, Chapter 6] and [SW93, Section 7]. Finitely many i.i.d. random flats through K lead to combinatorial questions. Associated random variables, such as numbers of intersection points inside K if $d = 2$ and $r = 1$, are hard to attack; for work of Sulanke (1965) and Gates (1984) see [SW93]. Of special interest is the case of $i \leq d$ i.i.d. uniform hyperplanes $H_1, \ldots, H_i$ through a convex body $K \subset \mathbb{R}^d$. Let p_i denote the probability that the intersection $H_1 \cap \ldots \cap H_i$ also meets K. In some special cases, the maximum of this probability (which depends on K and on the distribution of the hyperplanes) is known, but not in general. References for this and related problems and a conjecture are found in [BS95]. If $N > d$ i.i.d. uniform hyperplanes through K are given, they give rise to a random cell decomposition of $\operatorname{int} K$. For $k \in \{0, \ldots, d\}$, the expected number, $\mathbb{E}\nu_k$, of k-dimensional cells of this decomposition is given by

$$\mathbb{E}\nu_k = \sum_{i=d-k}^{d} \binom{i}{d-k} \binom{N}{i} p_i,$$

with p_i as defined above (Schneider, see [SW93]). If the hyperplanes are isotropic uniform, then

$$\mathbb{E}\nu_k = \sum_{i=d-k}^{d} \binom{i}{d-k} \binom{N}{i} \frac{i!}{2^i} \kappa_i \frac{V_i(K)}{V_1(K)^n}.$$

OPEN PROBLEM

PROBLEM 9.2.1

For $i \leq d$ i.i.d. uniform random hyperplanes through a convex body K, find the sharp upper bound for the probability p_i that their intersection also intersects K.

9.2.3 POISSON FLATS

A suitable model for infinite discrete random arrangements of r-flats in $\mathbb{R}^d$ is provided by a point process in the space $\mathcal{E}_r^d$. Stationary Poisson processes are the simplest and geometrically most interesting examples. Basic work was done by Miles [Mil71] and Matheron [Mat75]. In the case $r = d-1$, one speaks of a ***stationary Poisson hyperplane network***. For such a hyperplane process, an ith intersection density D_i can be defined, in such a way that, for a Borel set $A \subset \mathbb{R}^d$, the expectation of the total i-dimensional volume inside A, of the intersections of any $d-i$ hyperplanes of the process, is given by $D_i\lambda^d(A)$. Given the intensity D_{d-1}, the maximal ith intersection density D_i (for an $i \in \{0,\dots,d-2\}$) is achieved if the process is isotropic (its distribution is rigid-motion-invariant); this result is due to Thomas (1984, see [WW93]). Similar questions can be asked for stationary Poisson r-flats with $r < d-1$, for example for $2r \geq d$ and intersections of any two r-flats. Here nonisotropic extremal cases occur, such as in the case $r = 2$, $d = 4$ solved by Mecke [Mec88]. Various other cases have been treated; see Mecke [Mec91], Keutel [Keu91], and the references given there.

9.3 RANDOM SETS

For defining a ***random closed set*** in $\mathbb{R}^d$, one considers the set $\mathcal{F}$ of all closed subsets of $\mathbb{R}^d$ and equips it with the topology of closed convergence and the induced σ-algebra of Borel sets. A random closed set in $\mathbb{R}^d$ is then a measurable map from some probability space into $\mathcal{F}$. For this important model of stochastic geometry we refer to Matheron [Mat75] and to Stoyan, Kendall, and Mecke [SKM87]. Random closed sets are sometimes of use in describing certain discrete structures, such as random mosaics. However, random sets appearing in discrete geometry are mostly of a very restricted type, such as random congruent or similar copies of a fixed convex body.

9.3.1 RANDOM CONGRUENT COPIES

The following is a typical question on random congruent copies. Let $K_0, K \subset \mathbb{R}^d$ be given convex bodies. An isotropic random congruent copy of K meeting K_0 is of the form gK, where g is a random element of the motion group G_d of $\mathbb{R}^d$, and the distribution of g is obtained from the Haar measure on G_d by restricting it to the set $\{g \in G_d \mid K_0 \cap gK \neq \emptyset\}$ and normalizing. Let $K_1,\dots,K_n$ be convex bodies, let g_iK_i be an isotropic random copy of K_i meeting K_0, and suppose that $g_1,\dots,g_n$ are stochastically independent. What is the probability that the random bodies $g_1K_1,\dots,g_nK_n$ have a common point inside K_0? This question and similar ones can be given explicit answers by means of integral geometry. We refer to the books of Santaló [San76] and of Schneider and Weil [SW92].

9.3.2 BALLS AND SEGMENTS

More in the spirit of discrete geometry are random sets depending only on finitely many parameters, such as balls or segments.

Generalizing convex hulls of random points, Affentranger and Dwyer [AD93] considered convex hulls of i.i.d. random balls. A facet of such a convex hull is, by definition, a $(d-1)$-dimensional intersection with a supporting hyperplane. The authors obtained asymptotic estimates for the expected number of facets of the convex hull of n i.i.d. random balls, for different natural distributions. One of their ways of generating a random ball is to take the ball bounded by the sphere through $d+1$ independent uniform random points in B^d; another is to take the ball for which two independent uniform points in B^d determine a diameter. In a third model, the center of a random ball is uniform in B^d and its radius is independently and uniformly chosen in $[0,1]$.

For these models, Dwyer [Dwy93] also investigated the expected numbers of maximal and minimal balls in a set of n i.i.d. random balls.

Motivated by the fact that computing and counting intersections of a set of line segments in the plane are fundamental problems in computational geometry, Devroye and Zhu [DZ94] considered finite collections of i.i.d. random segments in the plane and studied the expected behavior of the number of intersections. In their model, center, direction, and length of a random line segment are independently chosen according to given distributions; the directions are uniform.

9.3.3 BUFFON TYPE PROBLEMS

Buffon type problems ask for geometric probabilities of the following kind. A plane convex body K is tossed at random onto a plane that carries a regular mosaic. What is the probability that the random rigid copy of K meets none, or a given finite number, of the lines or segments defining the mosaic? In Buffon's classical needle problem, K is a segment of length L and the mosaic is a grid of parallel lines at equal distances $D > L$. One usually makes uniformity and independence assumptions. For example, in the needle problem one assumes that the distance of the center of the needle from the nearest line is uniformly distributed in the interval $[0, D/2]$ and that the angle between needle and lines is uniformly distributed and independent of the center. The probability that the needle hits a line is then given by $2L/\pi D$. For discussions of older investigations on Buffon's needle problem and its extensions we refer to Kendall and Moran [KM63] and to Solomon [Sol78].

Buffon type problems in various versions are still under investigation. Besides hitting probabilities, topics such as independence of different hitting events or hitting numbers and their asymptotic distributions have also been studied. Mostly the results depend on lengthy calculations, whereas general principles do not seem to play a major role. Investigations concern higher-dimensional or non-Euclidean spaces, balls and ellipsoids in lattices of parallelepipeds, general convex bodies, polygons in a triangular lattice, needles with nonuniformly distributed directions, nonregular lattices, and needles replaced by "noodles." A few typical papers are Ren and Zhang [RZ91], Vassallo [Vas92], Duma and Stoka [DS93], and Duma and Rizzo [DR94].

9.4 RANDOM TESSELLATIONS

By a ***tessellation*** of $\mathbb{R}^d$, or a ***mosaic*** in $\mathbb{R}^d$, we understand here a collection of d-dimensional polytopes such that their union is $\mathbb{R}^d$, any two polytopes have

no interior points in common, and any bounded set meets only finitely many of the polytopes. Such a tessellation is also determined by the hypersurface that is the complement of the interiors of the polytopes. A ***random tessellation*** can therefore be modeled either by a random hypersurface, or by a point process in the space of convex bodies, both with very special properties. General references are [SKM87, Chapter 10], [MSSW90, Chapter 3], and [WW93, Section 7].

Every stationary Poisson hyperplane network (as defined in Section 9.2.3) yields a tessellation of $\mathbb{R}^d$, with special and interesting properties. The two-dimensional case was thoroughly investigated by Miles [Mil73]. General relations between mean numbers of lower-dimensional faces, which do not require Poisson assumptions, were found by Mecke [Mec84].

An important example of a random mosaic is obtained by taking the Voronoi tessellation induced by a point process in $\mathbb{R}^d$. If X is a realization of a point process, one associates with each point x of X its Voronoi cell $\{z \,|\, \|z-x\| \leq \|z-y\| \text{ for all } y \in X\}$. A detailed study of the two-dimensional stationary Poisson case was made by Miles [Mil70]. General references are Møller [Møl94] and Okabe, Boots, and Sugihara [OBS92, Chapter 5]. Precise information on several mean values is available in the case of Voronoi tessellations induced by stationary Poisson processes in $\mathbb{R}^d$; see Møller [Møl89].

For more general random mosaics, typical questions that have been considered concern relations between various mean-value quantities, in particular the selection of "fundamental" ones (e.g. Kendall and Mecke [KM87]), or inequalities of isoperimetric type for mean values (e.g. Mecke [Mec87]).

9.5 SOURCES AND RELATED MATERIAL

SOURCES FOR STOCHASTIC GEOMETRY IN GENERAL

Stoyan, Kendall, and Mecke [SKM87]: A monograph on theoretical foundations and applications of stochastic geometry.

Matheron [Mat75]: A monograph on basic models of stochastic geometry and applications of integral geometry.

Santaló [San76]: The classical work on integral geometry and its applications to geometric probabilities.

Kendall and Moran [KM63]: A collection of problems on geometric probabilities.

Solomon [Sol78]: A selection of topics from geometric probability theory.

Ambartzumian [Amb90]: Develops a special approach to stochastic geometry via factorization of measures, with various applications.

Moran [Mor66], [Mor69], Little [Lit74], Baddeley [Bad77]: "Notes on recent research in geometrical probability," useful surveys with many references.

Baddeley [Bad82]: An introduction and reading list for stochastic geometry.

Baddeley [Bad84]: Connections of stochastic geometry with image analysis.

Weil and Wieacker [WW93]: A comprehensive handbook article on stochastic geometry.

RELATED CHAPTERS

Several topics are outside the scope of this chapter, although they could be subsumed under probabilistic aspects of discrete geometry. Among these are randomization and average-case analysis of geometric algorithms and the probabilistic analysis of combinatorial optimization problems in Euclidean spaces. Two classical topics of discrete geometry, namely packing and covering, were also excluded, for the reason that the existing probabilistic results are in a spirit rather far from discrete geometry.

Chapters of this Handbook in which these and related topics are covered are:

Chapter 1: Finite point configurations
Chapter 2: Packing and covering
Chapter 13: Basic properties of convex polytopes
Chapter 15: Face numbers of polytopes and complexes
Chapter 34: Randomized algorithms
Chapter 39: Mathematical programming

RELEVANT SURVEYS AND FURTHER SOURCES

Some of the topics treated have been the subjects of earlier surveys. The following sources contain references to the excluded topics as well as to work within the scope of this chapter.

Borgwardt [Bor87], Shamir [Sha87, Section 6], Todd [Tod89]: Information on the probabilistic analysis of linear programming algorithms under different model assumptions.

Dwyer [Dwy88] and later work: Contributions to the average-case analysis of geometric algorithms.

Hall [Hal88]: A monograph devoted to the probabilistic analysis of coverage problems.

Buchta [Buc85]: A survey on random polytopes.

Schneider [Sch88], Affentranger [Aff92]: Surveys on approximation of convex bodies by random polytopes.

Bauer and Schneider [BS95]: A collection of information on inequalities for geometric probabilities.

REFERENCES

[Aff92] F. Affentranger. Aproximación aleatoria de cuerpos convexos. *Publ. Mat.*, 36:85–109, 1992.

[AD93] F. Affentranger and R.A. Dwyer. The convex hull of random balls. *Adv. in Appl. Probab.*, 25:373–394, 1993.

[AS92] F. Affentranger and R. Schneider. Random projections of regular simplices. *Discrete Comput. Geom.*, 7:219–226, 1992.

[AFGP91] D.J. Aldous, J. Fristedt, P.S. Griffin, and W.E. Pruitt. The number of extreme points in the convex hull of a random sample. *J. Appl. Probab.*, 28:287–304, 1991.

[Amb90] R.V. Ambartzumian. *Factorization Calculus and Geometric Probability.* Volume 33 of *Encyclopedia Math. Appl.*, Cambridge University Press, 1990.

[Bad77] A.J. Baddeley. A fourth note on recent research in geometrical probability. *Adv. in Appl. Probab.*, 9:824–860, 1977.

[Bad82] A.J. Baddeley. Stochastic geometry: An introduction and reading list. *Internat. Statist. Rev.*, 50:179–193, 1982.

[Bad84] A.J. Baddeley. Stochastic geometry and image analysis. *CWI Newslett.*, 4:2–20, 1984.

[Bár89] I. Bárány. Intrinsic volumes and f-vectors of random polytopes. *Math. Ann.*, 285:671–699, 1989.

[Bár92] I. Bárány. Random polytopes in smooth convex bodies. *Mathematika*, 39:81–92, 1992.

[BB93] I. Bárány and C. Buchta. Random polytopes in a convex polytope, independence of shape, and concentration of vertices. *Math. Ann.*, 297:467–497, 1993.

[BS94] I. Bárány and W. Steiger. On the expected number of k-sets. *Discrete Comput. Geom.*, 11:243–263, 1994.

[BV93] I. Bárány and R.A. Vitale. Random convex hulls: Floating bodies and expectations. *J. Approx. Theory*, 75:130–135, 1993.

[BV94] Y.M. Baryshnikov and R.A. Vitale. Regular simplices and Gaussian samples. *Discrete Comput. Geom.*, 11:141–147, 1994.

[BS95] C. Bauer and R. Schneider. Extremal problems for geometric inequalities involving convex bodies. *Adv. in Appl. Probab.*, 27:20–34, 1995.

[BRS92] J. Bokowski, J. Richter-Gebert, and W. Schindler. On the distribution of order types. *Comput. Geom. Theory Appl.*, 1:127–142, 1992.

[Bor87] K.-H. Borgwardt. *The Simplex Method—a Probabilistic Approach.* Springer-Verlag, Berlin, 1987.

[Buc85] C. Buchta. Zufällige Polyeder—Eine Übersicht. In: E. Hlawka, editor, *Zahlentheoretische Analysis*, volume 1114 of *Lecture Notes in Math.*, pages 1–13. Springer-Verlag, Berlin, 1985.

[Buc90] C. Buchta. Distribution-independent properties of the convex hull of random points. *J. Theoret. Probab.*, 3:387–393, 1990.

[Buc94] C. Buchta. On the probability that two sets of points have disjoint convex hulls. *Rend. Circ. Mat. Palermo (2) Suppl.*, 35:67–74, 1994.

[BR93] C. Buchta and M. Reitzner. What is the expected volume of a tetrahedron whose vertices are chosen at random from a given tetrahedron? *Anz. Österreich. Akad. Wiss. Math.-Natur. Kl.*, 129:63–68, 1993.

[CG94] A.J. Cabo and P. Groeneboom. Limit theorems for functionals of convex hulls. *Probab. Theory Related Fields*, 100:31–56, 1994.

[CH91] H. Carnal and J. Hüsler. On the convex hull of n random points on a circle. *J. Appl. Probab.*, 28:231–237, 1991.

[Chu93] D.P.T. Chu. Random r-content of an r-simplex from beta-type-2 random points. *Canad. J. Statist.*, 21:285–293, 1993.

[DZ94] L. Devroye and B. Zhu. Intersections of random line segments. *Internat. J. Comput. Geom. Appl.*, 4:261–274, 1994.

[DR94] A. Duma and S. Rizzo. The Buffon's needle problem in a weak magnetic field. *Rend. Circ. Mat. Palermo (2) Suppl.*, 35:127–142, 1994.

[DS93] A. Duma and M. Stoka. Hitting probabilities for random ellipses and ellipsoids. *J. Appl. Probab.*, 30:971–974, 1993.

[Dwy88] R.A. Dwyer. *Average-Case Analysis of Algorithms for Convex Hulls and Voronoi Diagrams.* PhD Thesis, Carnegie-Mellon Univ., Pittsburgh, 1988.

[Dwy91] R.A. Dwyer. Convex hulls of samples from spherically symmetric distributions. *Discrete Appl. Math.*, 31:113–132, 1991.

[Dwy93] R.A. Dwyer. Maximal and minimal balls. *Comput. Geom. Theory Appl.*, 3:261–275, 1993.

[GS94] S. Glasauer and R. Schneider. Asymptotic approximation of smooth convex bodies by polytopes. *Forum Math.*, 8:363–377, 1996.

[Hal88] P. Hall. *Introduction to the Theory of Coverage Processes.* Wiley, New York, 1988.

[Hsi94] T. Hsing. On the asymptotic distribution of the area outside a random convex hull in a disk. *Ann. Appl. Probab.*, 4:478–493, 1994.

[Hue94] I. Hueter. The convex hull of a normal sample. *Adv. in Appl. Probab.*, 26:855–875, 1994.

[Kal90] F.J. Kaltenbach. Asymptotisches Verhalten zufälliger konvexer Polyeder. Dissertation, Univ. Freiburg i. Br., 1990.

[Ken89] D.G. Kendall. A survey of the statistical theory of shape. *Statist. Sci.*, 4:87–120, 1989.

[KM63] M.G. Kendall and P.A.P. Moran. *Geometrical Probability.* Griffin, New York, 1963.

[KM87] W.S. Kendall and J. Mecke. The range of the mean-value quantities of planar tessellations. *J. Appl. Probab.*, 24:411–421, 1987.

[Keu91] J. Keutel. Ein Extremalproblem für zufällige Ebenen und für Ebenenprozesse in höherdimensionalen Räumen. Dissertation, Univ. Jena, 1991.

[Küf94] K.-H. Küfer. On the approximation of a ball by random polytopes. *Adv. in Appl. Probab.*, 26:876–892, 1994.

[Lit74] D.V. Little. A third note on recent research in geometrical probability. *Adv. in Appl. Probab.*, 6:103–130, 1974.

[Man94] D. Mannion. The volume of a tetrahedron whose vertices are chosen at random in the interior of a parent tetrahedron. *Adv. in Appl. Probab.*, 26:577–596, 1994.

[Mat75] G. Matheron. *Random Sets and Integral Geometry.* Wiley, New York, 1975.

[Mec84] J. Mecke. Random tessellations generated by hyperplanes. In R.V. Ambartzumian and W. Weil, editors, *Stochastic Geometry, Geometric Statistics, Stereology*, pages 104–109. Teubner, Leipzig, 1984.

[Mec87] J. Mecke. Extremal properties of some geometric processes. *Acta Appl. Math.*, 9:61–69, 1987.

[Mec88] J. Mecke. An extremal property of random flats. *J. Microscopy*, 151:205–209, 1988.

[Mec91] J. Mecke. On the intersection density of flat processes. *Math. Nachr.*, 151:69–74, 1991.

[MSSW90] J. Mecke, R. Schneider, D. Stoyan, and W. Weil. *Stochastische Geometrie.* Volume 16 of *DMV Sem.*, Birkhäuser, Basel, 1990.

[Mil70] R.E. Miles. On the homogeneous planar Poisson point process. *Math. Biosci.*, 6:85–127, 1970.

[Mil71] R.E. Miles. Poisson flats in Euclidean spaces. II: Homogeneous Poisson flats and the complementary theorem. *Adv. in Appl. Probab.*, 3:1–43, 1971.

[Mil73] R.E. Miles. The various aggregates of random polygons determined by random lines in a plane. *Adv. Math.*, 10:256–290, 1973.

[Møl89] J. Møller. Random tessellations in $\mathbb{R}^d$. *Adv. in Appl. Probab.*, 21:37–73, 1989.

[Møl94] J. Møller. *Lectures on Random Voronoi Tessellations.* Volume 87 of *Lecture Notes in Statist.*, Springer-Verlag, New York, 1994.

[Mor66] P.A.P. Moran. A note on recent research in geometrical probability. *J. Appl. Probab.*, 3:453–463, 1966.

[Mor69] P.A.P. Moran. A second note on recent research in geometrical probability. *Adv. in Appl. Probab.*, 1:73–89, 1969.

[Myc87] J. Mycielski. On random convex hulls. *Colloq. Math.*, 51:263–265, 1987.

[OBS92] A. Okabe, B. Boots, and K. Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams.* Wiley, Chichester, 1992.

[RZ91] D. Ren and G. Zhang. Random convex sets in a lattice of parallelograms. *Acta Math. Sci.* (English Ed.), 11:317–326, 1991.

[RM80] H. Ruben and R.E. Miles. A canonical decomposition of the probability measure of sets of isotropic random points in $\mathbb{R}^n$. *J. Multivariate Anal.*, 10:1–18, 1980.

[San76] L.A. Santaló. *Integral Geometry and Geometric Probability.* Volume 1 of *Encyclopedia of Mathematics*, Addison-Wesley, Reading, 1976.

[Sch88] R. Schneider. Random approximation of convex sets. *J. Microscopy*, 151:211–227, 1988.

[SW92] R. Schneider and W. Weil. *Integralgeometrie.* Teubner, Stuttgart, 1992.

[SW93] R. Schneider and J.A. Wieacker. Integral geometry. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 1349–1390. Elsevier, Amsterdam, 1993.

[Sch94] C. Schütt. Random polytopes and affine surface area. *Math. Nachr.*, 170:227–249, 1994.

[Sha87] R. Shamir. The efficiency of the simplex method: a survey. *Management Sci.*, 33:301–334, 1987.

[Sol78] H. Solomon. *Geometric Probability.* Soc. Industr. Appl. Math., Philadelphia, 1978.

[SKM87] D. Stoyan, W.S. Kendall, and J. Mecke. *Stochastic Geometry and Its Applications.* Akademie-Verlag, Berlin, 1987.

[Tod89] M.J. Todd. Probabilistic models for linear programming. Tech. Rep. 836, School of Oper. Res. and Industr. Eng., Cornell Univ., Ithaca, 1989.

[Val95] P. Valtr. Probability that n random points are in convex position. In I. Bárány and J. Pach, editors, *The László Fejes Tóth Festschrift. Discrete Comput. Geom.*, 13:637–643, 1995.

[Val96] P. Valtr. The probability that n random points in a triangle are in convex position. *Combinatorica*, 16:567–573, 1996.

[Vas92] S. Vassallo. Some problems of geometric probability relative to hyperspheres in the Euclidean space E_n. *Arch. Math.*, 59:95–104, 1992.

[VS92] A.M. Vershik and P.V. Sporyshev. Asymptotic behavior of the number of faces of random polyhedra and the neighborliness problem. *Selecta Math. Soviet.*, 11:181–201, 1992.

[WW93] W. Weil and J.A. Wieacker. Stochastic geometry. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 1391–1438. Elsevier, Amsterdam, 1993.

10 GEOMETRIC DISCREPANCY THEORY AND UNIFORM DISTRIBUTION

J. Ralph Alexander, József Beck, and William W.L. Chen

INTRODUCTION

A sequence $s_1, s_2, \ldots$ in $\mathbf{U} = [0,1)$ is said to be *uniformly distributed* if, in the limit, the number of s_j falling in any given subinterval is proportional to its length. Equivalently, $s_1, s_2, \ldots$ is uniformly distributed if the sequence of equiweighted atomic probability measures $\mu_N(s_j) = 1/N$, supported by the initial N-segments $s_1, s_2, \ldots, s_N$, converges weakly to Lebesgue measure on $\mathbf{U}$. This notion immediately generalizes to any topological space with a corresponding probability measure on the Borel sets.

Uniform distribution, as an area of study, originated from the remarkable paper of Weyl [Wey16], in which he established the fundamental result known nowadays as the Weyl criterion (see [Cas57, KN74]). This reduces a problem on uniform distribution to a study of related exponential sums, and provides a deeper understanding of certain aspects of Diophantine approximation, especially basic results such as Kronecker's density theorem. Indeed, careful analysis of the exponential sums that arise often leads to Erdős-Turán type upper bounds, which in turn lead to quantitative statements concerning uniform distribution.

Today, the concept of uniform distribution has important applications in a number of branches of mathematics such as number theory (especially Diophantine approximation), combinatorics, ergodic theory, discrete geometry, statistics, numerical analysis, etc. In this chapter, we focus on the geometric aspects of the theory.

10.1 UNIFORM DISTRIBUTION OF SEQUENCES

GLOSSARY

Uniformly distributed: Given a sequence $(s_n)_{n\in\mathbb{N}}$, with $s_n \in \mathbf{U} = [0,1)$, let $Z_N([a,b)) = |\{j \le N \mid s_j \in [a,b)\}|$. The sequence is uniformly distributed if, for every $0 \le a < b \le 1$, $\lim_{N\to\infty} \frac{Z_N([a,b))}{N} = b - a$.

Fractional part: The fractional part $\{x\}$ of a real number x is $x - \lfloor x \rfloor$.

Kronecker sequence: A sequence of points of the form $(\{N\alpha_1\}, \ldots . \{N\alpha_k\})_{N\in\mathbb{N}}$ in $\mathbf{U}^k$, where $1, \alpha_1, \ldots, \alpha_k \in \mathbb{R}$ are linearly independent over $\mathbb{Q}$.

Discrepancy, or ***irregularity of distribution:*** The discrepancy of a sequence

$(s_n)_{n\in\mathbb{N}}$, with $s_n \in \mathbf{U} = [0,1)$, in a subinterval $[a,b)$ of $\mathbf{U}$, is

$$\Delta_N([a,b)) = |Z_N([a,b)) - N(b-a)|.$$

More generally, the discrepancy of a sequence $(s_n)_{n\in\mathbb{N}}$, with $s_n \in \mathcal{S}$, a topological probability space, in a measurable subset $A \subset \mathcal{S}$, is $\Delta_N(A) = |Z_N(A) - N\mu(A)|$, where $Z_N(A) = |\{j \le N \mid s_j \in A\}|$.

Aligned rectangle, aligned triangle: A rectangle (resp. triangle) in $\mathbb{R}^2$ two sides of which are parallel to the coordinate axes.

Hausdorff dimension: A set S in a metric space has Hausdorff dimension m, where $0 \le m \le +\infty$, if

(i) for any $0 < k < m$, $\mu_k(S) > 0$;

(ii) for any $m < k < +\infty$, $\mu_k(S) < +\infty$.

Here, μ_k is the k-dimensional ***Hausdorff measure***, given by

$$\mu_k(S) = 2^{-k}\kappa_k \liminf_{\epsilon\to 0}\left\{\sum_{i=1}^{\infty}(\operatorname{diam} S_i)^k \;\middle|\; S \subset \cup_{i=1}^{\infty} S_i,\ \operatorname{diam} S_i \le \epsilon\right\},$$

where κ_k is the volume of the unit ball in $\mathbb{E}^k$.

Remark. Throughout this chapter, the symbol c will always represent the generic absolute positive constant, depending only on the indicated parameters. The value generally varies from one appearance to the next.

It is not hard to prove that for any irrational number α, the sequence of fractional parts $\{N\alpha\}$ is everywhere dense in $\mathbf{U}$ (here N is the running index). Suppose that the numbers $1, \alpha_1, \ldots, \alpha_k$ are linearly independent over the rationals. Then Kronecker's theorem states that the k-dimensional Kronecker sequence $(\{N\alpha_1\}, \ldots, \{N\alpha_k\})$ is dense in the unit k-cube $\mathbf{U}^k$. It is a simple consequence of the Weyl criterion that any such Kronecker sequence is uniformly distributed in $\mathbf{U}^k$, a much stronger result than the density theorem. For example, letting $k = 1$, we see that $\{N\sqrt{2}\}$ is uniformly distributed in $\mathbf{U}$.

Weyl's work led naturally to the question: How rapidly can a sequence in $\mathbf{U}$ become uniformly distributed as measured by the discrepancy $\Delta_N([a,b))$ of subintervals? Here, $\Delta_N([a,b)) = |Z_N([a,b)) - N(b-a)|$, where $Z_N([a,b))$ counts those $j \le N$ for which s_j lies in $[a,b)$. Thus we see that Δ_N measures the difference between the actual number of s_j in an interval and the expected number. The sequence is uniformly distributed if and only if $\Delta_N(I) = o(N)$ for all subintervals I. The notion of discrepancy immediately extends to any topological probability space, provided there is at hand a suitable collection of measurable sets $\mathcal{J}$ corresponding to the intervals. If A is in $\mathcal{J}$, set $\Delta_N(A) = |Z_N(A) - N\mu(A)| = N|\mu_N(A) - \mu(A)|$.

From the works of Hardy, Littlewood, Ostrowski, and others, it became clear that the smaller the partial quotients in the continued fractions of the irrational number α are, the more uniformly distributed the sequence $\{N\alpha\}$ is. For instance, the partial quotients of quadratic irrationals are characterized by being cyclic, hence bounded. Studying the behavior of $\{N\alpha\}$ for these numbers has proved an excellent indicator of what might be optimal for general sequences in $\mathbf{U}$. Here one has $\Delta_N(I) < c(\alpha)\log N$ for all intervals I and integers $N \ge 2$. Unfortunately, one does not have anything corresponding to continued fractions in higher dimensions, and this is an obstacle to a similar study of Kronecker sequences.

Van der Corput gave an alternative construction of a super uniformly distributed sequence of rationals in $\mathbf{U}$ for which $\Delta_N(I) < c \log N$ for all intervals I and integers $N \geq 2$ (see [KN74, p. 127]). He also asked for the best possible estimate in this direction. In particular, he posed

PROBLEM 10.1.1 *Van der Corput Problem*

Can there exist a sequence for which $\Delta_N(I) < c$ for all N and I?

He conjectured, in a slightly different formulation, that such a sequence could not exist. This conjecture was affirmed by van Aardenne-Ehrenfest, who later showed that for any sequence in $\mathbf{U}$, $\sup_I \Delta_N(I) > c \log\log N / \log\log\log N$ for infinitely many values of N. Her pioneering work gave the first nontrivial lower bound on the discrepancy of general sequences in $\mathbf{U}$. We remark that it is trivial to construct a sequence for which $\sup_I \Delta_N(I) \leq 1$ for infinitely many values of N.

In a classic paper, Roth showed that for any infinite sequence in $\mathbf{U}$, it must be true that $\sup_I \Delta_N(I) > c(\log N)^{1/2}$ for infinitely many N. Finally, in another classic paper, Schmidt used an entirely new method to prove the following result.

THEOREM 10.1.2 *Schmidt*

The inequality $\sup_I \Delta_N(I) > c \log N$ holds for infinitely many N.

For a more detailed discussion of work arising from the van der Corput conjecture, see [BC87, pp. 3–6].

In light of van der Corput's sequence, as well as $\{N\sqrt{2}\}$, Schmidt's result is best possible. The following problem, which has been described as "excruciatingly difficult," is a major remaining open question from the classical theory.

PROBLEM 10.1.3

Extend Schmidt's result to a best possible estimate of the discrepancy for sequences in $\mathbf{U}^k$ for $k > 1$.

For a given sequence, the results above do not imply the existence of a fixed interval I in $\mathbf{U}$ for which $\sup_N \Delta_N(I) = \infty$. Let I_α denote the interval $[0, \alpha)$, where $0 < \alpha \leq 1$. Schmidt showed that for any fixed sequence in $\mathbf{U}$ there are only countably many values of α for which $\Delta_N(I_\alpha)$ is bounded. The best result in this direction is due to Halász.

THEOREM 10.1.4 *Halász*

For any fixed sequence in $\mathbf{U}$, let A denote the set of values of α for which $\Delta_N(I_\alpha) = o(\log N)$. Then A has Hausdorff dimension 0.

For a more detailed discussion of work arising from this question, see [BC87, pp. 10–11].

The fundamental works of Roth and Schmidt opened the door to the study of discrepancy in higher dimensions, and there were surprises. In his classic paper, Roth transformed the heart of van der Corput's problem to a question concerning the unit square $\mathbf{U}^2$. In this new formulation, Schmidt's "$\log N$ theorem" implies that if N points are placed in $\mathbf{U}^2$, there is always an aligned rectangle $I = [\gamma_1, \alpha_1) \times [\gamma_2, \alpha_2)$ having discrepancy exceeding $c \log N$. Roth also showed that it was possible to place N points in the square $\mathbf{U}^2$ so that the discrepancy of no aligned rectangle exceeds $c \log N$. One way is to choose $p_j = ((j-1)/N, \{j\sqrt{2}\})$ for $j \leq N$. Thus, the function $c \log N$ describes the *minimax discrepancy* for aligned rectangles. However,

Schmidt showed that there is always an aligned right triangle (the part of an aligned rectangle above, or below, a diagonal) with discrepancy exceeding $cN^{1/4-\epsilon}$! Later work has shown that $cN^{1/4}$ exactly describes the minimax discrepancy of aligned right triangles. This paradoxical behavior is not isolated.

Generally, if one studies a collection $\mathcal{J}$ of "nice" sets such as disks, aligned boxes, rotated cubes, etc., in $\mathbf{U}^k$ or some other convex region, it turns out that the minimax discrepancy is either bounded above by $c(\log N)^r$ or bounded below by cN^s, with nothing halfway. In $\mathbf{U}^k$, typically $s = (k-1)/2k$. Thus, there tends to be a logarithmic version of the Vapnik-Chervonenkis principle in operation (see Chapter 31 of this Handbook for a related discussion). Later, we shall see how certain geometric properties place $\mathcal{J}$ in one or the other of these two classes.

10.2 THE GENERAL FREE PLACEMENT PROBLEM FOR *N* POINTS

One can ask for bounds on the discrepancy of N variable points $\mathcal{P} = \{p_1, p_2, \ldots, p_N\}$ that are freely placed in a domain $\mathbf{K}$ in Euclidean t-space $\mathbb{E}^t$. By contrast, when one considers the discrepancy of a *sequence* in $\mathbf{K}$, the initial n-segment of $p_1, \ldots, p_n$, $\ldots, p_N$ remains fixed for $n \leq N$ as new points appear with increasing N. For a given $\mathbf{K}$, as the unit interval $\mathbf{U}$ demonstrates, estimates for these two problems are quite different as functions of N. The freely placed points in $\mathbf{U}$ need never have discrepancy exceeding 1.

With Roth's reformulation (discussed in Section 10.1), the classical problem is easier to state and, more importantly, it generalizes in a natural manner to a wide class of problems. The bulk of geometric discrepancy problems are now posed as free placement problems. In practically all situations, the domain $\mathbf{K}$ has a very simple description as a cube, disk, sphere, etc., and standard notation is used in the specific situations.

PROBABILITY MEASURES AND DISCREPANCY

In a free placement problem there are two probability measures in play. First, there is the atomic measure μ^+ that assigns weight $1/N$ to each p_j. Second, there is a probability measure μ^- on the Borel sets of $\mathbf{K}$. The measure μ^- is generally the restriction of a natural uniform measure, such as scaled Lebesgue measure. An example would be given by $\mu^- = \sigma/4\pi$ on the unit sphere $\mathbf{S}^2$, where σ is the usual surface measure. It is convenient to define the signed measure $\mu = \mu^+ - \mu^-$ (in the previous section μ^- was denoted by μ). The discrepancy of a Borel set A is, as before, given by $\Delta(A) = |Z(A) - N\mu^-(A)| = N|\mu(A)|$.

The function Δ is always restricted to a very special collection $\mathcal{J}$ of sets, and the challenge lies in obtaining estimates concerning the restricted Δ. It is the central importance of the collection $\mathcal{J}$ that gives the study of discrepancy its distinct character. In a given problem it is sometimes possible to reduce the size of $\mathcal{J}$. Taking the unit interval $\mathbf{U}$ as an example, letting $\mathcal{J}$ be the collection of intervals $[\gamma, \alpha)$ seems to be the obvious choice. But a moment's reflection shows that only intervals of the form $I_\alpha = [0, \alpha)$ need be considered for estimates of discrepancy. At most a factor of 2 is introduced in any estimate of bounds.

THE MINIMAX DISCREPANCY

In most interesting problems $\mathcal{J}$ itself carries a measure ν in the sense of integral geometry, and this adds much more structure. While there is artistic latitude in the choice of ν, more often than not there is a natural measure on $\mathcal{J}$. In the example of $\mathbf{U}$, by identifying $I_\alpha = [0, \alpha)$ with its right endpoint, it is clear that Lebesgue measure on $\mathbf{U}$ is the natural choice for ν.

Given that the measure ν exists, for $0 < W < \infty$ define

$$\|\Delta(\mathcal{P}, \mathcal{J})\|_W = \left(\int_{\mathcal{J}} (\Delta(A))^W d\nu \right)^{1/W} \qquad \text{and} \qquad \|\Delta(\mathcal{P}, \mathcal{J})\|_\infty = \sup_{\mathcal{J}} \Delta(A),$$

and for $0 < W \leq \infty$ define

$$D(\mathbf{K}, \mathcal{J}, W, N) = \inf_{\mathcal{P}} \left\{ \|\Delta(\mathcal{P}, \mathcal{J})\|_W \mid \#\mathcal{P} = N \right\}.$$

The determination of the "minimax" $D(\mathbf{K}, \mathcal{J}, \infty, N)$ is generally the most important as well as the most difficult problem in the study. It should be noted that the function $D(\mathbf{K}, \mathcal{J}, \infty, N)$ is defined even if the measure ν is not. These various functions $D(\mathbf{K}, \mathcal{J}, W, N)$ measure how well the continuous distribution μ^- can be approximated by N freely placed atoms.

The inequality

$$\nu(\mathcal{J})^{-1/W} \|\Delta(\mathcal{P}, \mathcal{J})\|_W \leq \|\Delta(\mathcal{P}, \mathcal{J})\|_\infty \tag{10.2.1}$$

provides a general approach for obtaining a lower bound for $D(\mathbf{K}, \mathcal{J}, \infty, N)$. The choice $W = 2$ has been especially fruitful, but good estimates of $D(\mathbf{K}, \mathcal{J}, W, N)$ for any W are of independent interest.

An upper bound on $D(\mathbf{K}, \mathcal{J}, \infty, N)$ generally is obtained by showing the existence of a favorable example. This may be done either by a direct construction, often extremely difficult to verify, or by a probabilistic argument showing such an example does exist without giving it explicitly. These comments would apply as well to upper bounds for any $D(\mathbf{K}, \mathcal{J}, W, N)$.

10.3 ALIGNED RECTANGLES IN THE UNIT SQUARE

The unit square $\mathbf{U}^2 = [0,1) \times [0,1)$ is by far the most thoroughly studied 2-dimensional object. The main reason for this is Roth's reformulation of the van der Corput problem. Many of the interesting questions that arose have been answered, and we give a summary of the highlights.

For $\mathbf{U}^2$ one wishes to study the discrepancy of rectangles of the type $I = [\gamma_1, \alpha_1) \times [\gamma_2, \alpha_2)$. It is a trivial observation that only those I for which $\gamma_1 = \gamma_2 = 0$ need be considered, and this restricted family, denoted by $\mathcal{B}^2$, is the choice for $\mathcal{J}$. By considering this smaller collection one introduces at most a factor of 4 on bounds. There is a natural measure ν on $\mathcal{B}^2$, which may be identified with Lebesgue measure on $\mathbf{U}^2$ via the upper right corner points (α_1, α_2). In the same spirit, let $\mathcal{B}^1$ denote the previously introduced collection of intervals $I_\alpha = [0, \alpha)$ in $\mathbf{U}$.

THEOREM 10.3.1 *Roth's Equivalence* [BC87, pp. 6–7]

Let f be a positive increasing function tending to infinity. Then the following two statements are equivalent:

(i) *There is an absolute positive constant c_1 such that for any finite sequence $s_1, s_2, \ldots, s_N$ in $\mathbf{U}$, there always exists a positive integer $n \leq N$ such that $\|\Delta(\mathcal{P}_n, \mathcal{B}^1)\|_\infty > c_1 f(N)$. Here, $\mathcal{P}_n$ is the initial n-segment.*

(ii) *There is an absolute positive constant c_2 such that for all positive integers N, $D(\mathbf{U}^2, \mathcal{B}^2, \infty, N) > c_2 f(N)$.*

The equivalence shows that the central question of bounds for the van der Corput problem can be replaced by an elegant problem concerning the free placement of N points in the unit square $\mathbf{U}^2$. The mapping $s_j \to ((j-1)/N, s_j)$ plays a role in the proof of this equivalence. If one takes as $\mathcal{P}_N$ the image in $\mathbf{U}^2$ under the mapping of the initial N-segment of the van der Corput sequence, the following upper bound theorem may be proved.

THEOREM 10.3.2 *Lerch* [BC87, Theorem 4, $K = 2$]

For $N \geq 2$,

$$D(\mathbf{U}^2, \mathcal{B}^2, \infty, N) < c \log N. \tag{10.3.1}$$

The corresponding lower bound is established by the important "$\log N$ theorem" of Schmidt.

THEOREM 10.3.3 *Schmidt* [BC87, Theorem 3B]

One has

$$D(\mathbf{U}^2, \mathcal{B}^2, \infty, N) > c \log N. \tag{10.3.2}$$

By an explicit lattice construction, Davenport gave the best possible upper bound estimate for $W = 2$. His analysis shows that if the irrational number α has continued fractions with bounded partial quotients, then the $N = 2M$ points in $\mathbf{U}^2$ given by

$$p_j^{\pm} = ((j-1)/M, \{\pm j\alpha\}), \qquad j \leq M,$$

can be taken as $\mathcal{P}$ in proving the following theorem. Other proofs have been given by Vilenkin, Halton and Zaremba, and Roth. For more details, see [BC87, p. 5].

THEOREM 10.3.4 *Davenport* [BC87, Theorem 2A]

For $N \geq 2$,

$$D(\mathbf{U}^2, \mathcal{B}^2, 2, N) < c(\log N)^{1/2}. \tag{10.3.3}$$

This complements the following lower bound obtained by Roth in his classic paper.

THEOREM 10.3.5 *Roth* [BC87, Theorem 1A, $K = 2$]

One has

$$D(\mathbf{U}^2, \mathcal{B}^2, 2, N) > c(\log N)^{1/2}. \tag{10.3.4}$$

For $W = 1$, an upper bound $D(\mathbf{U}^2, \mathcal{B}^2, 1, N) < c(\log N)^{1/2}$ follows at once from Davenport's bound (10.3.3) by the monotonicity of $D(\mathbf{U}^2, \mathcal{B}^2, W, N)$ as a function of W. The corresponding lower bound was obtained by Halász more recently.

THEOREM 10.3.6 *Halász* [BC87, Theorem 1C, $K = 2$]
One has

$$D(\mathbf{U}^2, \mathcal{B}^2, 1, N) > c(\log N)^{1/2}. \tag{10.3.5}$$

Halász (see [BC87, Theorem 3C]) deduced that there is always an aligned square of discrepancy larger than $c \log N$. Of course, the square generally will not be a member of the special collection $\mathcal{B}^2$. Ruzsa [Ruz93] has given a clever elementary proof that the existence of such a square follows directly from inequality (10.3.2) above.

The ideas developed in the study of discrepancy can be applied to approximations of integrals. We briefly mention one example, a theorem of Chen restricted to 2 dimensions. A function ψ is termed ***M-simple*** if $\psi(x) = \sum_{j=1}^{M} m_j \chi_{B_j}(x)$, where χ_{B_j} is the characteristic function of the aligned rectangle B_j. In this theorem, the lower bounds are nontrivial because of the logarithmic factors coming from discrepancy theory on $\mathbf{U}^2$.

THEOREM 10.3.7 *Chen* [BC87, Theorems 5A, 5C]
Let the function f be defined on $\mathbf{U}^2$ by $f(x) = C + \int_{B(x)} g(y)dy$ where C is a constant, g is nonzero on a set of positive measure in $\mathbf{U}^2$, and $B(x_1, x_2) = [0, x_1) \times [0, x_2)$. Then, for any M-simple function ψ,

$$\begin{aligned} &\|f - \psi\|_W > c(f, W) M^{-1} (\log M)^{1/2}, \qquad 1 \le W < \infty; \\ &\|f - \psi\|_\infty > c(f) M^{-1} \log M. \end{aligned} \tag{10.3.6}$$

10.4 ALIGNED BOXES IN A UNIT k-CUBE

The van der Corput problem led to the study of $D(\mathbf{U}^2, \mathcal{B}^2, W, N)$, which in turn led to the study of $D(\mathbf{U}^k, \mathcal{B}^k, W, N)$ for general positive integers k and positive real W. Here, $\mathcal{B}^k$ denotes the collection of boxes $I = [0, \alpha_1) \times \ldots \times [0, \alpha_k)$, and the measure ν is identified with Lebesgue measure on $\mathbf{U}^k$ via the corner points $(\alpha_1, \ldots, \alpha_k)$.

The principle of Roth's equivalence extends so that the discrepancy problem for sequences in $\mathbf{U}^k$ reformulates as a free placement problem in $\mathbf{U}^{k+1}$, so that we discuss only the latter version. Inequalities (10.3.1) – (10.3.5) give the exact order of magnitude of $D(\mathbf{U}^2, \mathcal{B}^2, W, N)$ for the most natural values of W, namely $1 \le W \le 2$ and $W = \infty$, with the latter being top prize. While much is known, knowledge of $D(\mathbf{U}^k, \mathcal{B}^k, W, N)$ is incomplete, especially for $W = 1$ and $W = \infty$. It should be remarked that if k and N are fixed, then $D(\mathbf{U}^k, \mathcal{B}^k, W, N)$ is a nondecreasing function of W for $0 < W \le \infty$.

As was indicated earlier, upper bound methods generally fall into two classes, explicit constructions and probabilistic existence arguments. In practice, careful constructions are made prior to a probabilistic averaging process. Chen's proof of the following upper bound theorem involved extensive combinatorial and number-theoretic constructions as well as probabilistic considerations.

THEOREM 10.4.1 *Chen* [BC87, Theorem 2D]
For W satisfying $0 < W < \infty$, and integers $k \ge 2$ and $N \ge 2$,

$$D(\mathbf{U}^k, \mathcal{B}^k, W, N) < c(W, k)(\log N)^{(k-1)/2}. \tag{10.4.1}$$

A second proof was given by Chen (see [BC87, Section 3.5]). Earlier, Roth (see [BC87, Theorem 2C]) treated the case $W = 2$. The inequality (10.4.1) highlights one of the truly baffling aspects of the theory, namely the apparent jump discontinuity in the asymptotic behavior of $D(\mathbf{U}^k, \mathcal{B}^k, W, N)$ at $W = \infty$. This discontinuity is most dramatically established for $k = 2$, but is known to occur for $k = 3$ (see (10.4.5) below).

Explicit multidimensional sequences greatly generalizing the van der Corput sequence also have been used to obtain upper bounds for $D(\mathbf{U}^k, \mathcal{B}^k, \infty, N)$. Halton constructed explicit point sets in $\mathbf{U}^k$ in order to prove the next theorem. Faure (see [BC87, Section 3.2]) gave a different proof of the same result. If $k = 2$ is a guide, Halton's result may in fact be the best possible.

THEOREM 10.4.2 *Halton* [BC87, Theorem 4]
For integers $k \geq 2$ and $N \geq 2$,

$$D(\mathbf{U}^k, \mathcal{B}^k, \infty, N) < c(k)(\log N)^{k-1}. \tag{10.4.2}$$

In order to prove (10.3.3), Davenport used properties of special lattices; but only very recently has there been further success with lattices in higher dimensions. Skriganov has announced some most interesting results, which imply the following theorem. Given a region, a lattice is termed ***admissible*** if the region contains no member of the lattice except possibly the origin (see [Cas59]). Examples for the following theorem are given by lattices arising from algebraic integers in totally real algebraic number fields.

THEOREM 10.4.3 *Skriganov* [Skr94]
Suppose Γ is a fixed k-dimensional lattice admissible for the region $|x_1 x_2 \ldots x_k| < 1$.

(i) *Halton's upper bound inequality* (10.4.2) *holds if the N points are obtained by intersecting $\mathbf{U}^k$ with $t\Gamma$, where $t > 0$ is a suitably chosen real scalar.*

(ii) *With the same choice of t as in part* (i)*, there exists $x \in \mathbb{E}^k$ such that Chen's upper bound inequality* (10.4.1) *holds if the N points are obtained by intersecting $\mathbf{U}^k$ with $t\Gamma + x$.*

Moving to lower bound estimates, the following theorem of Schmidt is complemented by Chen's result (10.4.1). For $W \geq 2$ this lower bound is due to Roth, since D is monotone in W.

THEOREM 10.4.4 *Schmidt* [BC87, Theorem 1B]
For $W > 1$ and integers $k \geq 2$,

$$D(\mathbf{U}^k, \mathcal{B}^k, W, N) > c(W, k)(\log N)^{(k-1)/2}. \tag{10.4.3}$$

Concerning $W = 1$, there is the result of Halász, which is probably not optimal. It is reasonably conjectured that $(k-1)/2$ is the correct exponent for all $W < \infty$.

THEOREM 10.4.5 *Halász* [BC87, Theorem 1C]
For integers $k \geq 2$,

$$D(\mathbf{U}^k, \mathcal{B}^k, 1, N) > c(k)(\log N)^{1/2}. \tag{10.4.4}$$

The next lower bound estimate belongs to Beck. Although probably not best possible, it firmly establishes a discontinuity in asymptotic behavior at $W = \infty$ for $k = 3$. For $k > 2$, it is the first step beyond Roth's lower bound estimate for $D(\mathbf{U}^k, \mathcal{B}^k, \infty, N)$ in over 30 years.

THEOREM 10.4.6 *Beck* [Bec89]
For integers $N > 20$,

$$D(\mathbf{U}^3, \mathcal{B}^3, \infty, N) > c(\log N)(\log\log N)^{1/8-\epsilon}. \tag{10.4.5}$$

For $k > 3$, Roth's lower bound $D(\mathbf{U}^k, \mathcal{B}^k, \infty, N) > c(k)(\log N)^{(k-1)/2}$ remains unimproved. Can the factor $1/2$ be removed from the exponent? This is the "great open problem." However, Beck has refined Roth's estimate in a geometric direction.

THEOREM 10.4.7 *Beck* [BC87, Theorem 19A]
Let $\mathcal{J}$ be the collection of aligned cubes contained in $\mathbf{U}^k$. Then

$$D(\mathbf{U}^k, \mathcal{J}, \infty, N) > c(k)(\log N)^{(k-1)/2}. \tag{10.4.6}$$

Actually, Beck's method shows $D(\mathbf{U}^k, \mathcal{J}, 2, N) > c(k)(\log N)^{(k-1)/2}$, with respect to a natural measure ν on sets of aligned cubes. This improves Roth's inequality $D(\mathbf{U}^k, \mathcal{B}^k, 2, N) > c(k)(\log N)^{(k-1)/2}$. So far, it has not been possible to extend Ruzsa's ideas to higher dimensions in order to show that the previous theorem follows directly from Roth's estimate. However, very recently, Drmota [Drm96] has announced a new proof that $D(\mathbf{U}^k, \mathcal{J}, 2, N) > c(k)D(\mathbf{U}^k, \mathcal{B}^k, 2, N)$, and this does imply (10.4.6).

10.5 MOTION-INVARIANT PROBLEMS

In this section and the next three, we discuss collections $\mathcal{J}$ of convex sets having the property that any set in $\mathcal{J}$ may be moved by a direct (orientation preserving) motion of $\mathbb{E}^k$ and yet remain in $\mathcal{J}$. Motion-invariant problems were first extensively studied by Schmidt, and many of his estimates, obtained by a difficult technique using integral equations, were close to best possible. The book [BC87] contains an account of Schmidt's methods. But more recently, the Fourier transform method of Beck has achieved results that in general surpass those obtained by Schmidt. For a broad class of problems, Beck's Fourier method gives nearly best possible estimates for $D(\mathbf{K}, \mathcal{J}, 2, N)$.

The pleasant surprise is that if $\mathcal{J}$ is motion-invariant, then the bounds on $D(\mathbf{K}, \mathcal{J}, \infty, N)$ turn out to be very close to those for $D(\mathbf{K}, \mathcal{J}, 2, N)$. This is shown by a probabilistic upper bound method, which generally pins $D(\mathbf{K}, \mathcal{J}, \infty, N)$ between bounds differing at most by a factor of $c(k)(\log N)^{1/2}$.

The simplest motion-invariant example is given by letting $\mathcal{J}$ be the collection of all directly congruent copies of a given convex set A. In this situation, $\mathcal{J}$ carries a natural measure ν, which may be identified with Haar measure on the motion group on $\mathbb{E}^k$. A broader choice would be to let $\mathcal{J}$ be all sets in $\mathbb{E}^k$ directly similar to A. Again, there is a natural measure ν on $\mathcal{J}$. However, for the results stated in the next two sections, the various measures ν on the choices for $\mathcal{J}$ will not be

discussed in great detail. In most situations, such measures do play an active role in the proofs through inequality (10.2.1) with $W = 2$. A complete exposition of integration in the context of integral geometry, Haar measure, etc., may be found in the book by Santaló [San76].

For any domain $\mathbf{K}$ in $\mathbb{E}^t$ and each collection $\mathcal{J}$, it is helpful to define three auxiliary collections:

Definition:

(i) $\mathcal{J}_{tor}$ consists of those subsets of $\mathbf{K}$ obtained by reducing elements of $\mathcal{J}$ modulo $\mathbb{Z}^k$. To avoid messiness, let us always suppose that $\mathcal{J}$ has been restricted so that this reduction is 1–1 on each member of $\mathcal{J}$. For example, one might consider only those members of $\mathcal{J}$ having diameter less than 1.

(ii) $\mathcal{J}_c$ consists of those subsets of $\mathbf{K}$ that are members of $\mathcal{J}$.

(iii) $\mathcal{J}_i$ consists of those subsets of $\mathbf{K}$ obtained by intersecting $\mathbf{K}$ with members of $\mathcal{J}$.

Note that $\mathcal{J}_c$ and $\mathcal{J}_i$ are well defined for any domain $\mathbf{K}$. However, $\mathcal{J}_{tor}$ essentially applies only to $\mathbf{U}^k$. Viewed as a flat torus, $\mathbf{U}^k$ is the proper domain for Kronecker sequences and Weyl's exponential sums. There are several general inequalities for discrepancy results involving $\mathcal{J}_{tor}$, $\mathcal{J}_c$, and $\mathcal{J}_i$. For example, $D(\mathbf{U}^k, \mathcal{J}_c, \infty, N) \leq D(\mathbf{U}^k, \mathcal{J}_{tor}, \infty, N)$ because $\mathcal{J}_c$ is contained in $\mathcal{J}_{tor}$. Also, if the members of $\mathcal{J}$ have diameters less than 1, then we have $D(\mathbf{U}^k, \mathcal{J}_{tor}, \infty, N) \leq 2^k D(\mathbf{U}^k, \mathcal{J}_i, \infty, N)$, since any set in $\mathcal{J}_{tor}$ is the union of at most 2^k sets in $\mathcal{J}_i$.

10.6 SIMILAR OBJECTS CONTAINED IN THE UNIT k-CUBE

GLOSSARY

If A is a compact convex set in $\mathbb{E}^k$, let $d(A)$ denote the diameter of A, $r(A)$ denote the radius of the largest k-ball contained in A, and $\sigma(\partial A)$ denote the surface content of ∂A. The collection $\mathcal{J}$ is said to be ***ds-generated*** by A if $\mathcal{J}$ consists of all directly similar images of A having diameters not exceeding $d(A)$.

We state two pivotal theorems of Beck. As usual, if $\mathcal{S}$ is a discrete set, $Z(B)$ denotes the cardinality of $B \cap \mathcal{S}$.

THEOREM 10.6.1 *Beck* [BC87, Theorem 17A]

Let $\mathcal{S}$ be an arbitrary infinite discrete set in $\mathbb{E}^k$, A be a compact convex set with $r(A) \geq 1$, and $\mathcal{J}$ be ds-generated by A. Then there is a set B in $\mathcal{J}$ such that

$$|Z(B) - \operatorname{vol} B| > c(k)(\sigma(\partial A))^{1/2}. \tag{10.6.1}$$

COROLLARY 10.6.2 *Beck* [BC87, Corollary 17B]

Let A be a compact convex body in $\mathbb{E}^k$ with $r(A) \geq N^{-1/k}$, and let $\mathcal{J}$ be ds-generated by A. Then

$$D(\mathbf{U}^k, \mathcal{J}_{tor}, \infty, N) > c(A)N^{(k-1)/2k}. \tag{10.6.2}$$

The deduction of Corollary 10.6.2 from Theorem 10.6.1 involves a simple rescaling argument. Another important aspect of Beck's work is the introduction of upper bound methods based on probabilistic considerations. The following result shows that Theorem 10.6.1 is very nearly best possible.

THEOREM 10.6.3 *Beck* [BC87, Theorem 18A]

Let A be a compact convex body in $\mathbb{E}^k$ with $r(A) \geq 1$, and let $\mathcal{J}$ be ds-generated by A. Then there exists an infinite discrete set $\mathcal{S}_0$ such that for every set B in $\mathcal{J}$,

$$|Z(B) - \operatorname{vol} B| < c(k)(\sigma(\partial A))^{1/2}(\log \sigma(\partial A))^{1/2}. \tag{10.6.3}$$

COROLLARY 10.6.4 *Beck* [BC87, Corollary 18C]

Let A be a compact convex body in $\mathbb{E}^k$, and $\mathcal{J}$ be ds-generated by A. Then

$$D(\mathbf{U}^k, \mathcal{J}_{tor}, \infty, N) < c(A)N^{(k-1)/2k}(\log N)^{1/2}. \tag{10.6.4}$$

Beck (see [BC87, pp. 129–130]) deduced several related corollaries from Theorem 10.6.3. The example sets $\mathcal{P}_N$ for Corollary 10.6.4 can be taken as the initial segments of a certain fixed sequence whose choice definitely depends on A. If $d(A) = \lambda$ and A is either a disk (solid sphere) or a cube, then the right side of (10.6.2) takes the form $c(k)(\lambda^k N)^{(k-1)/2k}$. Montgomery has obtained a similar lower bound for cubes and disks.

The problem of estimating discrepancy for $\mathcal{J}_c$ is even more challenging because of "boundary effects." We state, as an example, a theorem for disks. The right inequality follows from (10.6.4).

THEOREM 10.6.5 *Beck* [BC87, Theorem 16A]

Let $\mathcal{J}$ be ds-generated by a k-disk. Then

$$c_1(k, \epsilon)N^{(k-1)/2k-\epsilon} < D(\mathbf{U}^k, \mathcal{J}_c, \infty, N) < c_2(k)N^{(k-1)/2k}(\log N)^{1/2}. \tag{10.6.5}$$

Because all the lower bounds above come from $\mathbf{L}^2$ estimates, these various results (10.6.1)–(10.6.5) allow us to make the general statement that for W in the range $2 \leq W \leq \infty$, the magnitude of $D(\mathbf{U}^k, \mathcal{J}, W, N)$ is controlled by $N^{(k-1)/2k}$. Thus there is no extreme discontinuity in asymptotic behavior at $W = \infty$. However, recent work by Beck and Chen proves that there is a discontinuity at some $1 \leq W \leq 2$, and the following results indicate that $W = 1$ is a likely candidate.

THEOREM 10.6.6 *Beck, Chen* [BC93b]

Let $\mathcal{J}$ be ds-generated by a convex polygon A with $d(A) < 1$. Then

$$\begin{aligned} &D(\mathbf{U}^2, \mathcal{J}_{tor}, W, N) < c(A, W)N^{(W-1)/2W}, \qquad 1 < W \leq 2; \\ &D(\mathbf{U}^2, \mathcal{J}_{tor}, 1, N) < c(A)(\log N)^2. \end{aligned} \tag{10.6.6}$$

The next theorem shows that powers of N other than $N^{(k-1)/2k}$ may appear for $2 \leq W \leq \infty$. It deals with what has been termed the ***isotropic discrepancy*** of $\mathbf{U}^k$.

THEOREM 10.6.7 *Schmidt* [BC87, Theorem 15]

Let $\mathcal{J}$ be the collection of all convex sets in $\mathbb{E}^k$. Then

$$D(\mathbf{U}^k, \mathcal{J}_i, \infty, N) > c(k)N^{(k-1)/(k+1)}. \tag{10.6.7}$$

The function $N^{(k-1)/(k+1)}$ dominates $N^{(k-1)/2k}$, so that this largest possible choice for $\mathcal{J}$ does in fact yield a larger discrepancy. Beck has shown that inequality (10.6.7), excepting a possible logarithmic factor, is best possible for $k = 2$.

The following result of Larcher [Lar91] shows that for certain rotation-invariant $\mathcal{J}$ the discrepancy of Kronecker sequences (defined in Section 10.1) will not behave as $cN^{(k-1)/2k}$, but as the square of this quantity.

THEOREM 10.6.8 *Larcher*

Let the sequence of point sets $\mathcal{P}_N$ be the initial segments of a Kronecker sequence in $\mathbf{U}^k$, and let $\mathcal{J}$ be ds-generated by a cube of edge length $\lambda < 1$. Then, for each N,

$$\|\Delta(\mathcal{P}_N, \mathcal{J}_i)\|_\infty > c(k)\lambda^{k-1}N^{(k-1)/k}. \tag{10.6.8}$$

Furthermore, the exponent $(k-1)/k$ cannot be increased.

10.7 CONGRUENT OBJECTS IN THE UNIT k-CUBE

GLOSSARY

If $\mathcal{J}$ consists of all directly congruent copies of a convex set A, we say that A ***dm-generates*** $\mathcal{J}$. Simple examples are given by the collection of all k-disks of a fixed radius r or by the collection of all k-cubes of a fixed edge length λ.

Given a convex set A, there is some evidence for the conjecture that the discrepancy for the dm-generated collection will be essentially as large as that for the ds-generated collection. However, this is generally very difficult to establish, even in very specific situations. There are the following results in this direction. The upper bound inequalities all come from Corollary 10.6.4 above.

THEOREM 10.7.1 *Beck* [BC87, Theorem 22A]

Let $\mathcal{J}$ be dm-generated by a square of edge length λ. Then

$$c_1(\lambda)N^{1/8} < D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N) < c_2(\lambda)N^{1/4}(\log N)^{1/2}. \tag{10.7.1}$$

It is felt that $N^{1/4}$ gives the proper lower bound, and for $\mathcal{J}_i$ this is definitely true. The lower bound in the next result follows at once from the work of Alexander [Ale91] described in Section 10.8.

THEOREM 10.7.2 *Alexander, Beck*

Let $\mathcal{J}$ be dm-generated by a k-cube of edge length λ. Then

$$c_1(\lambda, k)N^{(k-1)/2k} < D(\mathbf{U}^k, \mathcal{J}_i, \infty, N) < c_2(\lambda, k)N^{(k-1)/2k}(\log N)^{1/2}. \tag{10.7.2}$$

A similar result probably holds for k-disks, but this has been established only for $k = 2$.

THEOREM 10.7.3 *Beck* [BC87, Theorem 22B]
Let $\mathcal{J}$ be dm-generated by a 2-disk of radius r. Then

$$c_1(r)N^{1/4} < D(\mathbf{U}^2, \mathcal{J}_i, \infty, N) < c_2(r)N^{1/4}(\log N)^{1/2}. \qquad (10.7.3)$$

In closing this section it should be reported that Montgomery [Mon89] has independently developed a lower bound method which, as does Beck's method, uses techniques from harmonic analysis. Montgomery's method, especially in dimension 2, obtains for a number of special classes $\mathcal{J}$ estimates comparable to those obtained by Beck's method. In particular, Montgomery has considered $\mathcal{J}$ that are ds-generated by a region whose boundary is a piecewise smooth simple closed curve.

10.8 HALFSPACES AND RELATED OBJECTS

GLOSSARY

Segment: Given a compact subset $\mathbf{K}$ and a closed halfspace H in $\mathbb{E}^k$, $K \cap H$ is called a segment of $\mathbf{K}$.

Slab: The region between two parallel hyperplanes.

Spherical slice: The intersection of two open hemispheres on a sphere.

Let H be a closed halfspace in $\mathbb{E}^k$. Then the collection $\mathcal{H}^k$ of all closed halfspaces is dm-generated by H, and if we associate H with the oriented hyperplane ∂H, there is a well known invariant measure ν on $\mathcal{H}^k$. Further information concerning this and related measures may be found in Chapter 12 of Santaló [San76]. For a compact domain $\mathbf{K}$ in $\mathbb{E}^k$, it is clear that only the collection $\mathcal{H}_i^k$, the *segments* of $\mathbf{K}$, are proper for study, since $\mathcal{H}_c^k$ is empty and $\mathcal{H}_{tor}^k$ is unsuitable.

In this section, it is necessary for the domain $\mathbf{K}$ to be somewhat more general; hence we make only the following broad assumptions:

(i) $\mathbf{K}$ lies on the boundary of a fixed convex set $\mathbf{M}$ in $\mathbb{E}^{k+1}$;

(ii) $\sigma(\mathbf{K}) = 1$, where σ is the usual k-measure on $\partial\mathbf{M}$.

Since $\mathbb{E}^k$ is the boundary of a convex body in $\mathbb{E}^{k+1}$, any set in $\mathbb{E}^k$ of unit Lebesgue k-measure satisfies these assumptions. The normalization of assumption (ii) is for convenience, and, by rescaling, the inequalities of this section may be applied to any uniform probability measure on a domain $\mathbf{K}$ in $\mathbb{E}^{k+1}$. Such rescaling only affects dimensional constants; for standard domains, such as the unit k-sphere $\mathbf{S}^k$ and the unit k-disk $\mathbf{D}^k$, this will be done without comment.

Although in applications $\mathbf{K}$ will have a simple geometric description, the next theorem treats the general situation and obtains the essentially exact magnitude of $D(\mathbf{K}, \mathcal{H}_i^{k+1}, 2, N)$. If $\mathbf{K}$ lies in $\mathbb{E}^k$, then $\mathcal{H}^{k+1}$ may be replaced by $\mathcal{H}^k$. If ν is properly normalized, this change invokes no rescaling.

THEOREM 10.8.1 *Alexander* [Ale91]
Let $\mathcal{K}$ be the collection of all $\mathbf{K}$ satisfying assumptions (i) *and* (ii) *above. Then*

$$c_1(k)N^{(k-1)/2k} < \inf_{K \in \mathcal{K}} D(\mathbf{K}, \mathcal{H}_i^{k+1}, 2, N) < c_2(\mathbf{M})N^{(k-1)/2k}. \qquad (10.8.1)$$

The upper bound of (10.8.1) can be proved by an indirect probabilistic method introduced by Alexander [Ale72] for $\mathbf{K} = \mathbf{S}^2$, but the method of Beck and Chen [BC90] also may be applied for standard choices of $\mathbf{K}$ such as $\mathbf{U}^k$ and $\mathbf{D}^k$. When $\mathbf{M} = \mathbf{K} = \mathbf{S}^k$, the segments are the spherical caps. For this important special case the upper bound is due to Stolarsky [Sto73], while the lower bound is due to Beck (see [BC87, Theorem 24B]).

Since the ν-measure of the halfspaces that separate $\mathbf{M}$ is less than $c(k)d(\mathbf{M})$, inequality (10.2.1) may be applied to obtain a lower bound for $D(\mathbf{K}, \mathcal{H}_i^{k+1}, \infty, N)$. The upper bound in the following theorem should be taken in the context of actual applications such as $\mathbf{M}$ being a k-sphere $\mathbf{S}^k$, a compact convex body in $\mathbb{E}^k$, or more generally, a compact convex hypersurface in $\mathbb{E}^{k+1}$.

THEOREM 10.8.2 *Alexander, Beck*

Let $\mathcal{K}$ be the collection of $\mathbf{K}$ satisfying assumptions (i) *and* (ii) *above. Furthermore, suppose that $\mathbf{M}$ is of finite diameter. Then*

$$c_3(k)(d(\mathbf{M}))^{-1/2}N^{(k-1)/2k} < \inf_{K\in\mathcal{K}} D(\mathbf{K}, \mathcal{H}_i^{k+1}, \infty, N) < c_4(\mathbf{M})N^{(k-1)/2k}(\log N)^{1/2}. \tag{10.8.2}$$

For $\mathbf{M} = \mathbf{K} = \mathbf{S}^k$, inequalities (10.8.2) are due to Beck, improving a slightly weaker lower bound by Schmidt [Sch69]. Consideration of $\mathbf{K} = \mathbf{U}^2$ makes it obvious that there exists an aligned right triangle with discrepancy at least $cN^{1/4}$, as stated in Section 10.1. For the case $\mathbf{M} = \mathbf{K} = \mathbf{D}^2$, a unit 2-disk (Roth's *disk-segment problem*), Beck (see [BC87, Theorem 23A]) obtained inequalities (10.8.2), excepting a factor $(\log N)^{-7/2}$ in the lower bound. Later, Alexander [Ale90] improved the lower bound, and Matoušek [Mat95] obtained essentially the same upper bound. Matoušek's work on $\mathbf{D}^2$ makes it seem likely that Beck's factor $(\log N)^{1/2}$ in his general upper bound theorem might be removable in many specific situations, but this is very challenging.

THEOREM 10.8.3 *Alexander, Matoušek*

For Roth's disk-segment problem,

$$c_1N^{1/4} < D(\mathbf{D}^2, \mathcal{H}_i^2, \infty, N) < c_2N^{1/4}. \tag{10.8.3}$$

Alexander's lower bound method, by the nature of the convolutions employed, gives information on the discrepancy of slabs. This is especially apparent in the recent work of Chazelle, Matoušek, and Sharir, who have developed a more direct and geometrically transparent version of Alexander's method. The following theorem on the discrepancy of thin slabs is a corollary to their technique. It is clear that if a slab has discrepancy Δ, then one of the two bounding halfspaces has discrepancy at least $\Delta/2$.

THEOREM 10.8.4 *Chazelle, Matoušek, Sharir* [CMS94]

Let N points lie in the unit cube $\mathbf{U}^k$. Then there exists a slab $\mathbf{T}$ of width $c_1(k)N^{-1/k}$ such that $\Delta(\mathbf{T}) > c_2(k)N^{(k-1)/2k}$.

Alexander [Ale94] has investigated the effect of the dimension k on the discrepancy of halfspaces, and obtained somewhat complicated inequalities that imply the following result.

THEOREM 10.8.5 *Alexander*

For the lower bounds in inequalities (10.8.1) *and* (10.8.2) *above, there is an absolute positive constant* c *such that one may choose* $c_1(k) > ck^{-3/4}$ *and* $c_3(k) > ck^{-1}$.

Schmidt [Sch69] studied the discrepancy of spherical slices (the intersection of two open hemispheres) on $\mathbf{S}^k$. Associating a hemisphere with its pole, Schmidt identified ν with the normalized product measure on $\mathbf{S}^k \times \mathbf{S}^k$. Blümlinger [Blü91] demonstrated a surprising relationship between halfspace (spherical cap) and slice discrepancy for $\mathbf{S}^k$. However, his definition for ν in terms of Haar measure on $SO(k+1)$ differed somewhat from Schmidt's.

THEOREM 10.8.6 *Blümlinger*

Let $\mathcal{S}^k$ *be the collection of slices of* $\mathbf{S}^k$. *Then*

$$c(k)D(\mathbf{S}^k, \mathcal{H}_i^{k+1}, 2, N) < D(\mathbf{S}^k, \mathcal{S}^k, 2, N). \tag{10.8.4}$$

For the next result, the left inequality follows from inequalities (10.2.1), (10.8.1), and (10.8.4). Blümlinger uses a version of Beck's probabilistic method to establish the right inequality.

THEOREM 10.8.7 *Blümlinger*

For slice discrepancy on $\mathbf{S}^k$,

$$c_1(k)N^{(k-1)/2k} < D(\mathbf{S}^k, \mathcal{S}^k, \infty, N) < c_2(k)N^{(k-1)/2k}(\log N)^{1/2}. \tag{10.8.5}$$

Grabner [Gra91] has given an Erdős-Turán type upper bound on spherical cap discrepancy in terms of spherical harmonics. This adds to the considerable body of results extending inequalities for exponential sums to other sets of orthonormal functions, and thereby extends the Weyl theory.

All of the results so far in this section treat $2 \leq W \leq \infty$. For W in the range $0 < W < 2$ there is mystery, but we do have the following result, related to inequality (10.6.6), showing that a dramatic change in asymptotic behavior occurs in the range $1 \leq W \leq 2$. For $\mathbf{U}^2$, Beck and Chen show that regular grid points will work for the upper bound example for $W = 1$, and they are able to modify their method to apply to any bounded convex domain in $\mathbb{E}^2$.

THEOREM 10.8.8 *Beck, Chen* [BC93a]

Let $\mathbf{K}$ *be a bounded convex domain in* $\mathbb{E}^2$. *Then*

$$\begin{aligned} &D(\mathbf{K}, \mathcal{H}_i^2, W, N) < c(\mathbf{K}, W)N^{(W-1)/2W}, \qquad 1 < W \leq 2; \\ &D(\mathbf{K}, \mathcal{H}_i^2, 1, N) < c(\mathbf{K})(\log N)^2. \end{aligned} \tag{10.8.6}$$

10.9 BOUNDARIES OF GENERATORS FOR HOMOTHETICALLY INVARIANT J

We have already noted several factors that play a role in determining whether $D(\mathbf{K}, \mathcal{J}, W, N)$ behaves like N^r as opposed to $(\log N)^s$. Beck's work shows that if $\mathcal{J}$ is dm-generated, $D(\mathbf{K}, \mathcal{J}, \infty, N)$ behaves as N^r. However, the work of Beck and Chen clearly shows that if W is sufficiently small, then even for motion-invariant $\mathcal{J}$,

it may be that $D(\mathbf{K}, \mathcal{J}, W, N)$ is bounded above by $(\log N)^s$. Beck has extensively studied $D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N)$ under the assumption that $\mathcal{J}$ is homothetically invariant, and in this section we record some of the results obtained. It turns out that the boundary shape of a generator is the critical element in determining to which, if either, class $\mathcal{J}$ belongs. Remarkably, for the "typical" homothetically invariant class $\mathcal{J}$, $D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N)$ oscillates infinitely often to be larger than $N^{1/4-\epsilon}$ and smaller than $(\log N)^{4+\epsilon}$.

GLOSSARY

The convex set A ***h-generates*** $\mathcal{J}$ if $\mathcal{J}$ consists of all homothetic images B of A with $d(B) \leq d(A)$.

Blaschke-Hausdorff metric: The metric on the space CONV(2) of all compact convex sets in $\mathbb{E}^2$ in which the distance between two sets is the minimum distance from any point of one set to the other.

A set is of ***first category*** if it is a countable union of nowhere dense sets.

If one considers the two examples of $\mathcal{J}$ being h-generated by an aligned square and by a disk, previously stated results make it very likely that shape strongly affects discrepancy for homothetically invariant $\mathcal{J}$. The first two theorems quantify this phenomenon for two very standard boundary shapes, first polygons, then smooth closed curves.

THEOREM 10.9.1 *Beck* [BC87, Corollary 20D]

Let $\mathcal{J}$ be h-generated by a convex polygon A. Then, for any $\epsilon > 0$,

$$D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N) = o((\log N)^{4+\epsilon}). \tag{10.9.1}$$

Beck and Chen [BC89] have given a less complicated argument that obtains $o((\log N)^{5+\epsilon})$ on the right side of (10.9.1).

THEOREM 10.9.2 *Beck* [BC87, Corollary 19F]

Let A be a compact convex set in $\mathbb{E}^2$ with a twice continuously differentiable boundary curve having strictly positive curvature. If A h-generates $\mathcal{J}$, then for $N \geq 2$,

$$D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N) > c(A) N^{1/4} (\log N)^{-1/2}. \tag{10.9.2}$$

Recently, for sufficiently smooth positively curved bodies, Drmota [Drm93] has extended (10.9.2) into higher dimensions and also removed the logarithmic factor. Thus, he obtains a lower bound of the form $c(A)N^{(k-1)/2k}$, along with the standard upper bound obtained by Beck's probabilistic method.

Let CONV(2) denote the usual locally compact space of all compact convex sets in $\mathbb{E}^2$ endowed with the Blaschke-Hausdorff metric. There is the following surprising result, which quantifies the oscillatory behavior mentioned above. The final theorem of this section will say more about the rationale of such estimates.

THEOREM 10.9.3 *Beck* [BC87, Theorem 21]

Let $\epsilon > 0$ be given. For all A in CONV(2)*, excepting a set of first category, if $\mathcal{J}$ is h-generated by A, then each of the following inequalities is satisfied infinitely often:*

$$D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N) < (\log N)^{4+\epsilon} \tag{10.9.3}$$

$$D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N) > N^{1/4}(\log N)^{-(1+\epsilon)/2}. \tag{10.9.4}$$

The next theorem gives the best lower bound estimate known if it is assumed only that the generator A has nonempty interior, certainly a minimal hypothesis.

THEOREM 10.9.4 *Beck* [BC87, Corollary 19G]
If $\mathcal{J}$ is h-generated by a compact convex set A having positive area, then

$$D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N) > c(A)(\log N)^{1/2}. \tag{10.9.5}$$

Possibly the right side should be $c(A)\log N$, which would be best possible as the example of aligned squares demonstrates. Lastly, we discuss the important theorem underlying most of these results about h-generated $\mathcal{J}$. Let A be a member of CONV(2) with nonempty interior, and for each integer $l \geq 3$ let A_l be an inscribed l-gon of maximal area. The Nth ***approximability number*** $\xi_N(A)$ is defined as the smallest integer l such that the area of $A \setminus A_l$ is less than l^2/N.

THEOREM 10.9.5 *Beck* [BC87, Corollary 19H,Theorem 20C]
Let A be a member of CONV(2) *with nonempty interior. Then if $\mathcal{J}$ is h-generated by A, we have*

$$c_1(A)(\xi_N(A))^{1/2}(\log N)^{-1/4} < D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N) < c_2(A,\epsilon)\xi_N(A)(\log N)^{4+\epsilon}. \tag{10.9.6}$$

The proof of the preceding fundamental theorem, which is in fact the join of two major theorems, is long, but the import is clear; namely, that for h-generated $\mathcal{J}$, if one understands $\xi_N(A)$, then one essentially understands $D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N)$. If $\xi_N(A)$ remains nearly constant for long intervals, then A acts like a polygon and D will drift below $(\log N)^{4+2\epsilon}$. If, at some stage, ∂A behaves as if it consists of circular arcs, then $\xi_N(A)$ will begin to grow as $cN^{1/2}$.

For still more information concerning the material in this section, along with the proofs, see [BC87, Chapter 7]. Very recently, Károlyi [Kár95a, Kár95b] has extended the idea of approximability number to higher dimensions and obtained upper bounds analogous to those in (10.9.6).

10.10 D(*K,J,2,N)* IN THE LIGHT OF DISTANCE GEOMETRY

Although knowledge of $D(\mathbf{K}, \mathcal{J}, \infty, N)$ is our highest aim, in the great majority of problems this is achieved by first obtaining bounds on $D(\mathbf{K}, \mathcal{J}, 2, N)$. In this section, we briefly show how this function fits nicely into the theory of metric spaces of negative type. In our situation, the distance between points will be given by a Crofton formula with respect to the measure ν on $\mathcal{J}$. This approach evolved from a paper written in 1971 by Alexander and Stolarsky investigating extremal problems in distance geometry, and has been developed in a number of subsequent papers by both authors studying special cases. However, we reverse history and leap immediately to a formulation suitable for our present purposes. We avoid mention of certain technical assumptions concerning $\mathcal{J}$ and ν which cause no difficulty in practice.

Assume that $\mathbf{K}$ is a compact convex set in $\mathbb{E}^k$ and that $\mathcal{J} = \mathcal{J}_c$. This latter

assumption causes no loss of generality since one can always just redefine $\mathcal{J}$. Let ν, as usual, be a measure on $\mathcal{J}$, with the further assumption that $\nu(\mathcal{J}) < \infty$.

Definition:

If p and q are points in $\mathbf{K}$, the set A in $\mathcal{J}$ is said to ***separate*** p and q if A contains exactly one of these two points. The ***distance function*** ρ on $\mathbf{K}$ is defined by the Crofton formula $\rho(p,q) = (1/2)\,\nu\{J \mid J \text{ separates } p \text{ and } q\}$, and if μ is any signed measure on $\mathbf{K}$ having finite positive and negative parts, one defines the functional $\mathbf{I}(\mu)$ by

$$\mathbf{I}(\mu) = \iint \rho(p,q)d\mu(p)d\mu(q).$$

With these definitions one obtains the following representation for $\mathbf{I}(\mu)$.

THEOREM 10.10.1 *Alexander*

One has

$$\mathbf{I}(\mu) = \int_{\mathcal{J}} \mu(A)\mu(\mathbf{K} \setminus A)d\nu(A). \tag{10.10.1}$$

For μ satisfying the condition of total mass zero, $\int_K d\mu = 0$, the integrand in (10.10.1) becomes $-(\mu(A))^2$. The signed measures $\mu = \mu^+ - \mu^-$ that we are considering, with μ^- being a uniform probability measure on $\mathbf{K}$ and μ^+ consisting of N atoms of equal weight $1/N$, certainly have total mass zero. Here one has $\Delta(A) = N\mu(A)$. Hence there is the following corollary.

COROLLARY 10.10.2

For the signed measures μ presently considered, if $\mathcal{P}$ denotes the N points supporting μ^+, then

$$-N^2\mathbf{I}(\mu) = \int_{\mathcal{J}} (\Delta(A))^2 d\nu(A) = (\|\Delta(\mathcal{P},\mathcal{J})\|_2)^2. \tag{10.10.2}$$

Thus if one studies the metric ρ, it may be possible to prove that $-\mathbf{I}(\mu) > f(N)$, whence it follows that $(D(\mathbf{K},\mathcal{J},2,N))^2 > N^2 f(N)$. If $\mathcal{J}$ consists of the halfspaces of $\mathbb{E}^k$, then ρ is the Euclidean metric. In this important special case, Alexander [Ale91] was able to make good estimates. Chazelle, Matoušek, and Sharir [CMS94] and A.D. Rogers [Rog94] contributed still more techniques for treating the halfspace problem.

If μ_1 and μ_2 are any two signed measures of total mass 1 on $\mathbf{K}$, then one can define the ***relative discrepancy*** $\Delta(A) = N(\mu_1(A) - \mu_2(A))$. The first equality of (10.10.2) still holds if $\mu = \mu_1 - \mu_2$. A signed measure μ_0 of total mass 1 is termed ***optimal*** if it solves the integral equation $\int_K \rho(x,y)d\mu(y) = \lambda$ for some positive number λ. If an optimal measure μ_0 exists, then $\mathbf{I}(\mu_0) = \lambda$ maximizes $\mathbf{I}$ on the class of all signed Borel measures of total mass 1 on $\mathbf{K}$. In the presence of an optimal measure, one has the following very pretty identity.

THEOREM 10.10.3 *Generalized Stolarsky Identity*

Suppose that the measure μ_0 is optimal on $\mathbf{K}$, and that μ is any signed measure of total mass 1 on $\mathbf{K}$. If Δ is the relative discrepancy with respect to μ_0 and μ, then

$$N^2\mathbf{I}(\mu) + \int_{\mathcal{J}} (\Delta(A))^2 d\nu(A) = N^2\mathbf{I}(\mu_0). \tag{10.10.3}$$

The first important example of this formula is due to Stolarsky [Sto73] where he treated the sphere $\mathbf{S}^k$, taking as μ the uniform atomic measure supported by N variable points. For $\mathbf{S}^k$ it is clear that the uniform probability measure μ_0 is optimal. His integrals involving the spherical caps are equivalent, up to a scale factor, to integrals with respect to the measure on the halfspaces of $\mathbb{E}^k$ for which ρ is the Euclidean metric. Stolarsky's tying of a geometric extremal problem to Schmidt's work on the discrepancy of spherical caps was a major step forward in the study of discrepancy and of distance geometry.

Very little has been done to investigate the deeper nature of the individual metrics ρ determined by classes $\mathcal{J}$ other than halfspaces. They are all metrics of ***negative type***, which essentially means that $\mathbf{I}(\mu) \leq 0$ if μ has total mass 0. There is a certain amount of general theory, begun by Schoenberg and developed by a number of others, but it does not apply directly to the problem of estimating discrepancy.

10.11 UNIFORM PLACEMENT OF POINTS ON SPHERES

As demonstrated by Stolarsky, formula (10.10.3) shows that if one places N points on $\mathbf{S}^k$ so that the sum of all distances is maximized, then $D(\mathbf{S}^k, \mathcal{H}_i^k, 2, N)$ is achieved by this arrangement. Berman and Hanes [BH77] have given a pretty algorithm that searches for optimal configurations. For $k = 2$, while the exact configurations are not known for $N \geq 5$, this algorithm appears to be successful for $N \leq 50$. For such an N surprisingly few rival configurations will be found. Lubotsky, Phillips, and Sarnak [LPS86] have given an algorithm, based on iterations of a specially chosen element in $SO(3)$, which can be used to place many thousands of reasonably well distributed points on $\mathbf{S}^2$. Difficult analysis shows that these points are well placed, but not optimally placed, relative to $\mathcal{H}_i^2$. On the other hand, it is shown that these points are essentially optimally placed with respect to a nongeometric operator discrepancy. Data concerning applications to numerical integration are also included in the paper. Very recently, Rakhmanov, Saff, and Zhou [RSZ94] have studied the problem of placing points uniformly on a sphere relative to optimizing certain functionals, and they state a number of interesting conjectures.

In yet another theoretical direction, the existence of very well distributed point sets on $\mathbf{S}^k$ allows the sphere, after difficult analysis, to be closely approximated by equi-edged zonotopes (sums of line segments). The recent papers of Wagner [Wag93] and of Bourgain and Lindenstrauss [BL93] treat this problem.

10.12 COMBINATORIAL DISCREPANCY

GLOSSARY

A **2-*coloring*** of $\mathbf{X}$ is a mapping $\chi : \mathbf{X} \to \{-1, 1\}$. For each such χ there is a natural integer-valued set function μ_χ on the finite subsets of $\mathbf{X}$ defined by

$\mu_\chi(A) = \sum_{x\in A}\chi(x)$, and if $\mathcal{J}$ is a given family of finite subsets of $\mathbf{X}$ we define

$$D(\mathbf{X},\mathcal{J}) = \min_{\chi}\max_{A\in\mathcal{J}}|\mu_\chi(A)|.$$

Degree: If $\mathcal{J}$ is a collection of subsets of a finite set X, $\deg\mathcal{J} = \max\{|\mathcal{J}(x)| \mid x \in \mathbf{X}\}$, where $\mathcal{J}(x)$ is the subcollection consisting of those members of $\mathcal{J}$ that contain x.

The collection $\mathcal{J}$ ***shatters*** a set $S \subset X$ if, for any given subset $B \subset S$, there exists A in $\mathcal{J}$ such that $B = A \cap S$. The ***VC-dimension*** of $\mathcal{J}$ is defined by $\dim_{vc}\mathcal{J} = \max\{|S| \mid S \subset \mathbf{X},\ \mathcal{J}\text{ shatters }S\}$. For $m \le |\mathbf{X}|$, the ***primal shatter function*** $\pi_{\mathcal{J}}$ is defined by

$$\pi_{\mathcal{J}}(m) = \max_{\substack{Y\subset\mathbf{X}\\|Y|\le m}}|\{Y\cap A \mid A\in\mathcal{J}\}|.$$

The ***dual shatter function*** is defined by $\pi^*_{\mathcal{J}}(m) = \pi_{\mathcal{J}^*}(m)$, where $\mathbf{X}^* = \mathcal{J}$, and $\mathcal{J}^* = \{\mathcal{J}(x) \mid x \in \mathbf{X}\}$.

Techniques in combinatorial discrepancy theory have proved very powerful in this geometric setting. Here one 2-colors a discrete set and studies the discrepancy of a special class $\mathcal{J}$ of subsets as measured by $|\#\text{red} - \#\text{blue}|$. If one 2-colors the first N positive integers, then the beautiful "1/4 theorem" of Roth says that there will always be an arithmetic progression having discrepancy at least $cN^{1/4}$. This result should be compared to van der Waerden's theorem, which says that there is a long monochromatic progression, whose discrepancy obviously will be its length. However, it is known that this length need not be more than $\log N$, and the minimax might be as small as $\log\log\ldots\log N$ (here the number of iterated logarithms may be arbitrarily large). Moreover, general results concerning combinatorial discrepancy, for example, those that use the Vapnik-Chervonenkis dimension, are very useful in computational geometry.

Combinatorial discrepancy theory involves discrepancy estimates arising from 2-colorings of a set $\mathbf{X}$. Upper bound estimates of combinatorial discrepancy have proved to be very helpful in obtaining upper bound estimates of geometric discrepancy. In this final section we briefly discuss various properties of the collection $\mathcal{J}$ that lead to useful upper bound estimates of combinatorial discrepancy.

The simplest property of the collection $\mathcal{J}$ is its cardinality $|\mathcal{J}|$. Here, Spencer obtained a fine result.

THEOREM 10.12.1 *Spencer* [AS93]
Let $\mathbf{X}$ *be a finite set. If* $|\mathcal{J}| \ge |\mathbf{X}|$, *then*

$$D(\mathbf{X},\mathcal{J}) \le c\left(|\mathbf{X}|\log\left(1+\frac{|\mathcal{J}|}{|\mathbf{X}|}\right)\right)^{1/2}. \tag{10.12.1}$$

Applications and extensions of the following theorem may be found in [BC87, Chapter 8].

THEOREM 10.12.2 *Beck, Fiala* [BC87, Lemma 8.5.]
Let $\mathbf{X}$ *be a finite set. Then*

$$D(\mathbf{X},\mathcal{J}) \le 2\deg\mathcal{J} - 1. \tag{10.12.2}$$

Since $\pi_{\mathcal{J}}(m) = 2^m$ if and only if $\dim_{vc} \mathcal{J} \geq m$, the function $\pi_{\mathcal{J}}$ contains much more information than does VC-dimension alone. If $\dim_{vc} \mathcal{J} = d$, then $\pi_{\mathcal{J}}(m)$ is polynomially bounded by cm^d. However, in many geometric situations this bound on the shatter function can be improved, leading to better discrepancy bounds. Detailed discussions may be found in the papers by Haussler and Welzl [HW87] and by Chazelle and Welzl [CW89].

Dual objects are defined in the usual manner (see Glossary). We state several recent results.

THEOREM 10.12.3 *Matoušek, Welzl, Wernisch* [MWW93]
Suppose that $(\mathbf{X}, \mathcal{J})$ *is a finite set system with* $|\mathbf{X}| = n$. *If* $\pi_{\mathcal{J}}(m) \leq c_1 m^d$ *for* $m \leq n$, *then*

$$\begin{aligned} D(\mathbf{X}, \mathcal{J}) &\leq c_2 n^{(d-1)/2d} (\log n)^{1+1/2d}, \qquad d > 1, \\ D(\mathbf{X}, \mathcal{J}) &\leq c_3 (\log n)^{5/2}, \qquad d = 1. \end{aligned} \tag{10.12.3}$$

If $\pi^*_{\mathcal{J}}(m) \leq c_4 m^d$ *for* $m \leq |\mathcal{J}|$, *then*

$$\begin{aligned} D(\mathbf{X}, \mathcal{J}) &\leq c_5 n^{(d-1)/2d} \log n, \qquad d > 1, \\ D(\mathbf{X}, \mathcal{J}) &\leq c_6 (\log n)^{3/2}, \qquad d = 1. \end{aligned} \tag{10.12.4}$$

More recently, Matoušek [Mat95] has shown that the factor $(\log n)^{1+1/2d}$ may be dropped from inequality (10.12.3) for $d > 1$, and has applied this result to half-spaces with great effect (see inequality (10.8.3)). One part of Matoušek's argument depends on combinatorial results of Haussler [Hau95].

10.13 SOURCES AND RELATED MATERIAL

FURTHER READING

The principal surveys on discrepancy theory are [BC87], [KN74], and [Sch77]. Auxiliary texts relating to this chapter include [AS93], [Cas57], [Cas59], and [San76].

RELATED CHAPTERS

Chapter 1: Finite point configurations
Chapter 2: Packing and covering
Chapter 8: Euclidean Ramsey theory
Chapter 9: Discrete aspects of stochastic geometry
Chapter 31: Range searching
Chapter 42: Computer graphics

REFERENCES

[Ale72] J.R. Alexander. On the sum of distances between n points on a sphere. *Acta Math. Hungar.*, 23:443–448, 1972.

[Ale90] J.R. Alexander. Geometric methods in the study of irregularities of distribution. *Combinatorica*, 10:115–136, 1990.

[Ale91] J.R. Alexander. Principles of a new method in the study of irregularities of distribution. *Invent. Math.*, 103:279–296, 1991.

[Ale94] J.R. Alexander. The effect of dimension on certain geometric problems of irregularities of distribution. *Pacific J. Math.*, 165:1–15, 1994.

[AS93] N. Alon and J. Spencer. *The Probabilistic Method.* Wiley, New York, 1993.

[Bec89] J. Beck. A two dimensional van Aardenne-Ehrenfest theorem in irregularities of distribution. *Compositio Math.*, 72:269–339, 1989.

[BC87] J. Beck and W.W.L. Chen. *Irregularities of Distribution.* Volume 89 of *Cambridge Tracts in Math.*, Cambridge University Press, 1987.

[BC89] J. Beck and W.W.L. Chen. Irregularities of point distribution relative to convex polygons. In G. Halász and V.T. Sós, editors, *Irregularities of Partitions*, volume 8 of *Algorithms Combin.*, pages 1–22. Springer-Verlag, Berlin, 1989.

[BC90] J. Beck and W.W.L. Chen. Note on irregularities of distribution II. *Proc. London Math. Soc.*, 61:251–272, 1990.

[BC93a] J. Beck and W.W.L. Chen. Irregularities of point distribution relative to half planes I. *Mathematika*, 40:102–126, 1993.

[BC93b] J. Beck and W.W.L. Chen. Irregularities of point distribution relative to convex polygons II. *Mathematika*, 40:127–136, 1993.

[BH77] J. Berman and K. Hanes. Optimizing the arrangement of points on the unit sphere. *Math. Comp.*, 31:1006–1008, 1977.

[Blü91] M. Blümlinger. Slice discrepancy and irregularities of distribution on spheres. *Mathematika*, 38:105–116, 1991.

[BL93] J. Bourgain and J. Lindenstrauss. Approximating the ball by a Minkowski sum of segments with equal length. *Discrete Comput. Geom.*, 9:131–144, 1993.

[Cas57] J.W.S. Cassels. *An Introduction to Diophantine Approximation.* Volume 45 of *Cambridge Tracts in Math.*, Cambridge University Press, 1957.

[Cas59] J.W.S. Cassels. *An Introduction to the Geometry of Numbers.* Volume 99 of *Mathematischen Wissenschaften*, Springer-Verlag, Berlin, 1959.

[CMS94] B. Chazelle, J. Matoušek, and M. Sharir. An elementary approach to lower bounds in geometric discrepancy. In I. Bárány and J. Pach, editors, *The László Fejes Tóth Festschrift, Discrete Comput. Geom.*, 13:363–381, 1995.

[CW89] B. Chazelle and E. Welzl. Quasi-optimal range searching in spaces of finite VC-dimension. *Discrete Comput. Geom.*, 4:467–489, 1989.

[Drm93] M. Drmota. Irregularities of distribution and convex sets. *Grazer Math. Ber.*, 318:9–16, 1993.

[Drm96] M. Drmota. Irregularities of distribution with respect to polytopes. *Mathematika*, 43:108–119, 1996.

[Gra91] P.J. Grabner. Erdős-Turán type discrepancy bounds. *Monatsh. Math.*, 111:127–135, 1991.

[Hau95] D. Haussler. Sphere packing numbers for subsets of the Boolean n-cube with bounded Vapnik-Chervonenkis dimension. *J. Combin. Theory Ser. A*, 69:217–232, 1995.

[HW87] D. Haussler and E. Welzl. ϵ-nets and simplex range queries. *Discrete Comput. Geom.*, 2:127–151, 1987.

[Kár95a] G. Károlyi. Geometric discrepancy theorems in higher dimensions. *Studia Sci. Math. Hungar.*, 30:59–94, 1995.

[Kár95b] G. Károlyi. Irregularities of point distributions with respect to homothetic convex bodies. *Monatsh. Math.*, 120:247–279, 1995.

[KN74] L. Kuipers and H. Niederreiter. *Uniform Distribution of Sequences.* Wiley, New York, 1974.

[Lar91] G. Larcher. On the cube discrepancy of Kronecker sequences. *Arch. Math. (Basel)*, 57:362–369, 1991.

[LPS86] A. Lubotsky, R. Phillips, and P. Sarnak. Hecke operators and distributing points on a sphere. *Comm. Pure Appl. Math.*, 39:149–186, 1986.

[Mat95] J. Matoušek. Tight upper bounds for the discrepancy of half-spaces. In I. Bárány and J. Pach, editors, *The László Fejes Tóth Festschrift, Discrete Comput. Geom.*, 13:593–601, 1995.

[MWW93] J. Matoušek, E. Welzl, and L. Wernisch. Discrepancy and approximations for bounded VC-dimension. *Combinatorica*, 13:455–467, 1993.

[Mon89] H.L. Montgomery. Irregularities of distribution by means of power sums. In *Congress of Number Theory (Zarautz)*, pages 11–27. Universidad del País Vasco, Bilbao, 1989.

[RSZ94] E.A. Rakhmanov, E.B. Saff, and Y.M. Zhou. Minimal discrete energy on the sphere. *Math. Res. Lett.*, 1:647–662, 1994.

[Rog94] A.D. Rogers. A functional from geometry with applications to discrepancy estimates and the Radon transform. *Trans. Amer. Math. Soc.*, 341:275–313, 1994.

[Ruz93] I.Z. Ruzsa. The discrepancy of rectangles and squares. *Grazer Math. Ber.*, 318:135–140, 1993.

[San76] L.A. Santaló. *Integral Geometry and Geometric Probability.* Volume 1 of *Encyclopedia of Mathematics*, Addison-Wesley, Reading, 1976.

[Sch69] W.M. Schmidt. Irregularities of distribution III. *Pacific J. Math.*, 29:225–234, 1969.

[Sch77] W.M. Schmidt. *Irregularities of Distribution.* Volume 56 of *Lecture Notes on Mathematics and Physics*, Tata, Bombay, 1977.

[Skr94] M.M. Skriganov. Constructions of uniform distributions in terms of geometry of numbers. *St. Petersburg Math. J. (Algebra i. Analiz)*, 6:200–230, 1994.

[Sto73] K.B. Stolarsky. Sums of distances between points on a sphere II. *Proc. Amer. Math. Soc.*, 41:575–582, 1973.

[Wag93] G. Wagner. On a new method for constructing good point sets on spheres. *Discrete Comput. Geom.*, 9:111–129, 1993.

[Wey16] H. Weyl. Über die Gleichverteilung von Zahlen mod Eins. *Math. Ann.*, 77:313–352, 1916.

11 TOPOLOGICAL METHODS

Rade T. Živaljević

INTRODUCTION

A problem is solved or some other goal achieved by "topological methods" if in our arguments we appeal to the "form," the "shape," the "global" rather than "local" structure of the object or configuration space associated with the phenomenon we are interested in. This configuration space is typically a manifold or a simplicial complex. The global properties of the configuration space are usually expressed in terms of its homology groups, which capture the idea of the higher (dis)connectivity of a geometric object. Topological methods have been used very successfully in discrete and computational geometry, especially in the problems of dissections of masses (ham sandwich-type theorems) and partitions of point sets (Tverberg-type theorems). The methods that have proven to be the most effective are based on various generalizations of the Borsuk-Ulam theorem. As a guiding principle that may give the reader a general idea of when and where it is reasonable to expect the success of topological methods, we formulate the following thesis.

Thesis: *Any global effect that depends on the object as a whole and that cannot be localized is of a homological nature, and should be amenable to topological methods.*

HOW IS TOPOLOGY APPLIED IN GEOMETRIC AND COMBINATORIAL PROBLEMS?

Step 1: Rephrase the problem in topological terms.

The problem should give us a clue how to define a "natural" *configuration space* X and how to interpret the question as a question about *the coincidence of continuous maps on* X or the *nonempty intersection of subsets of* X.

Step 2: Use a standard topological technique to solve the rephrased problem.

The topological technique that is most frequently used in discrete geometric problems is based on the technique of *intersecting homology classes* and on *generalized Borsuk-Ulam theorems.*

11.1 THE INTERSECTION OF HOMOLOGY CLASSES: AN EXAMPLE

We are frequently led to the problem of checking whether two subspaces A and B of a manifold X have nonempty intersection or, if $A \cap B \neq \emptyset$ is known, what else can be said about $A \cap B$. The question can be rephrased as a question about the intersection of homology classes in A and B.

GLOSSARY

Manifold: A d-manifold is a topological space locally homeomorphic to d-dimensional Euclidean space; curves and surfaces are manifolds of dimensions 1 and 2, respectively.

Homology classes, cohomology algebra, cup product: Algebraic objects and operations associated with topological spaces that measure "the number of holes in the space," "the higher disconnectivity of the space," "the torsion of the space," etc. (See any book on algebraic topology, for example [Mun84].)

Poincaré duality, intersection of homology classes: The fundamental duality between homology and cohomology of manifolds; the intersection product of homology classes is the operation dual to the cup product of cohomology classes.

Configuration space: A topological space that parameterizes all configurations of a given set of geometric objects.

11.1.1 A WATCH WITH TWO EQUAL HANDS

Here is a problem that serves as an introductory example of how topology comes into play and proves useful in geometric and combinatorial problems.

EXAMPLE 11.1.1

A watch was manufactured with a defect so that both hands (minute and hour) are identical. Otherwise, the watch works well and the question is whether you can always decide the right time in spite of the defect. One easily finds out that more often than not this is indeed possible. Actually, there are only a finite number of ambiguous positions, i.e., positions when it is not possible to determine the exact time. The problem is to find the exact number of these positions.

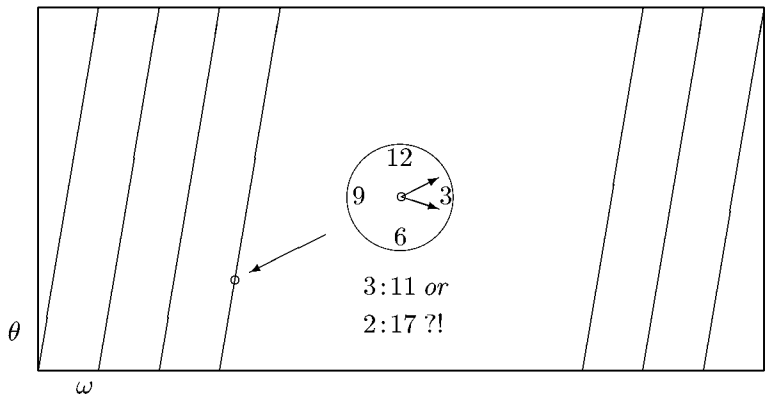


FIGURE 11.1.1
The configuration space of the two hands.

First of all we observe that every position of a hand is determined by an angle $\omega \in [0, 2\pi]$, so that the configuration space of all possible positions of a hand is

homeomorphic to the unit circle S^1. Two independent hands have the 2-dimensional torus $T^2 \cong S^1 \times S^1$ as their configuration space, i.e., the space representing all allowed states or positions of the system. A convenient model of a torus is the plane $\mathbb{R}^2$ with coordinates taken modulo 2π, i.e., $T^2 \cong \mathbb{R}^2/\mathbb{Z}^2$. A related model is just a square or a rectangle, see Figure 11.1.1, which produces the torus if opposite sides are glued together. If θ corresponds to the minute hand and ω is the coordinate of the hour hand, then the fact that the first hand is twelve times faster is recorded by the equation $\theta = 12\omega$. This equation describes a curve Γ_1 on the torus T^2, which is just a circle winding 12 times in the direction of the θ axis while it winds only once in the direction of ω axis. The curve Γ_1 is represented in our picture as the union of 12 line segments, seven of them indicated in Figure 11.1.1. If the hands change places then the corresponding curve Γ_2 has equation $\omega = 12\theta$. Note that the ambiguous positions are exactly the intersection points of these two curves (except those that belong to the diagonal $\Delta := \{(\theta, \omega) \,|\, \theta = \omega\}$, when it is still possible to tell the exact time without knowing which hand is for hours and which for minutes). The reader can now easily find the number of these intersection points and compute that there are 143 of them in the intersection $\Gamma_1 \cap \Gamma_2$, and 11 in the intersection $\Gamma_1 \cap \Gamma_2 \cap \Delta$, which shows that there are all together 132 ambiguous positions.

Let us note that the problem reduced to counting points or 0-dimensional manifolds in the intersection of two circles, viewed as 1-dimensional submanifolds of the 2-dimensional manifold T^2. More generally, one may be in a situation to ask how many points there are in the intersection of two or more submanifolds of a higher-dimensional ambient manifold. Topology gives us a versatile tool for computing this and much more in terms of the so-called *intersection product* $\alpha \frown \beta$ *of homology classes* α *and* β in a manifold M. This intersection product is, via Poincaré duality, equivalent to the "cup" product, and has the usual properties [Mun84]. In our example, knowing that $a \frown b = -\, b \frown a$ for all 1-dimensional classes, and in particular that $a \frown a = 0$ if $\dim a = 1$, we have

$$[\Gamma_1] \frown [\Gamma_2] = ([\theta] + 12\,[\omega]) \frown ([\omega] + 12\,[\theta]) =$$

$$= [\theta] \frown [\omega] + 12\,[\omega] \frown [\omega] + 12\,[\theta] \frown [\theta] + 144\,[\omega] \frown [\theta] = 143\,[\omega] \frown [\theta],$$

and, taking the orientation into account, we conclude that the number of intersection points is 143.

11.2 THE HAM SANDWICH THEOREM AND ITS GENERALIZATIONS

An (open) ham sandwich is a collection of three measurable sets in $\mathbb{R}^3$, representing a slice of bread, a slice of ham, and a slice of cheese. It turns out that there always exists a plane simultaneously halving all three measurable sets or, in other words, that a ham sandwich can be cut fairly into two pieces by a single straight cut. Suppose, on the other hand, that you want to split an irregularly shaped slice of pizza with a hungry friend who is supposed to divide the pizza into two pieces by a straight cut of a knife, but who can cut anywhere he likes. You are allowed to mark your piece in advance by specifying a single point that will lie in your piece. Then,

if you are very careful about marking your piece, you can count on at least one third of the pizza. These two results are instances of *the ham sandwich theorem* and *the center point theorem* which, together with their relatives and related topological statements, are shown in Figure 11.2.1.

GLOSSARY

Measure: An abstract function μ defined on a class of sets that has all the formal properties (additivity, positivity) of the usual *volume* or *area* functions.

Measurable set: Any set in the domain of the function μ.

Mass distribution and density function: A density function is an integrable function $f : \mathbb{R}^d \to [0, +\infty)$ representing the density of a "mass distribution" (measure) on $\mathbb{R}^d$. The measure μ arising this way is defined by $\mu(A) := \int_A f\, dx$.

Halving hyperplane: A hyperplane that simultaneously bisects a family of measurable sets.

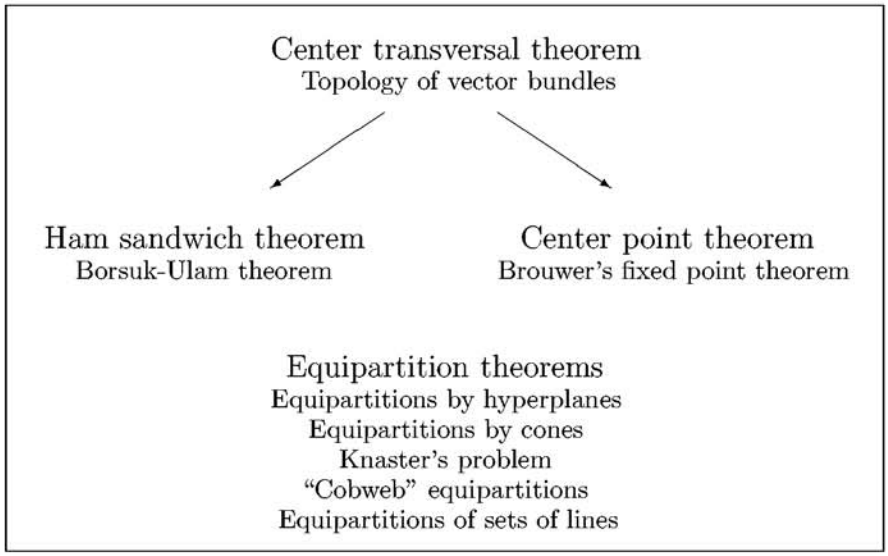


FIGURE 11.2.1
The ham sandwich theorem and its relatives.

11.2.1 THE HAM SANDWICH THEOREM

Given a collection of d measurable sets (mass distributions, finite sets) in $\mathbb{R}^d$, the problem is to simultaneously bisect all of them by a single hyperplane. Often a measurable set is a geometric object $A \subset \mathbb{R}^d$, say a polytope, whose measure is simply its volume vol A. More generally, a measurable set A is an arbitrary subset of $\mathbb{R}^d$ if it is clear from the context what we mean by its "measure" $\mu(A)$. Typically, A is a Lebesgue measurable set and $\mu(A) = m(A)$ its Lebesgue measure which, in the usual cases, reduces to the measure vol described above. More generally, if $f : \mathbb{R}^d \to \mathbb{R}^+$ is an integrable density function, then $\mu(A) := \int_A f\, dm = \int_{\mathbb{R}^d} f\phi_A\, dm$

is the measure or the mass distribution associated with the function f, where ϕ_A is the characteristic function of A (1 on A, 0 otherwise). An important special case arises if $f = \phi_B$ for a Lebesgue measurable set B, where $\mu(A) = m(A \cap B)$. Finally, if $S \subset \mathbb{R}^d$ is a finite set, then $\mu(A) := |A \cap S|$ is the so-called *counting measure* induced by the set S. In the usual language of measure theory, μ is an arbitrary positive, σ-additive function, defined on a family $\mathcal{F}$ of subsets of $\mathbb{R}^d$ that is large enough to include all halfspaces and other sets we are interested in. The reader should, in principle, not have any difficulty reformulating any of the following results for whatever special class of measures she may be interested in.

THEOREM 11.2.1 *Ham Sandwich Theorem*

Let $\mu_1, \mu_2, \ldots, \mu_d$ be a collection of measures (mass distributions, measurable sets, finite sets) in the sense above. Then there exists a hyperplane H such that for all $i = 1, \ldots, d$, $\mu_i(H^+) \geq 1/2\,\mu_i(\mathbb{R}^d)$ and $\mu_i(H^-) \geq 1/2\,\mu_i(\mathbb{R}^d)$, where H^+ and H^- are the closed halfspaces associated with the hyperplane H.

In the special case where $\mu(H) = 0$, i.e., where the hyperplane itself has measure zero, H is called a ***halving hyperplane*** since $\mu_i(H^+) = \mu_i(H^-) = 1/2\,\mu_i(\mathbb{R}^d)$ for all i. A halving hyperplane H is also called a "ham sandwich cut," for the reasons noted above.

TOPOLOGICAL BACKGROUND

The topological result that lies behind the ham sandwich theorem is the Borsuk-Ulam theorem.

THEOREM 11.2.2 *Borsuk-Ulam Theorem*

For every continous map $f : S^d \to \mathbb{R}^d$ from a d-dimensional sphere into d-dimensional Euclidean space, there exists a point $x \in S^d$ such that $f(x) = f(-x)$.

An important special case of the Borsuk-Ulam theorem arises if f is an "odd" map, i.e., if $f(-x) = -f(x)$ for all $x \in S^d$. The conclusion is that a continuous odd map must have a zero on the sphere, i.e., $f(x) = 0$ for some $x \in S^d$. (Conversely, the general Borsuk-Ulam theorem follows from the special case if the latter is applied to the map $\phi : S^d \to \mathbb{R}^d$ given by $\phi(x) = f(x) - f(-x)$.)

One can deduce the ham sandwich theorem from this by choosing an odd function $f : S^{d-1} \to \mathbb{R}^{d-1}$ so that a zero $x \in S^{d-1}$ of f is automatically a vector orthogonal to a halving hyperplane.

There is a different topological approach to the ham sandwich theorem closer to the initial example about a watch with two indistinguishable hands. Here we mention only that the role of the torus T^2 is played by a manifold representing all hyperplanes in $\mathbb{R}^d$, while the curves Γ_1 and Γ_2 are replaced by suitable submanifolds, one for each of the measures μ_i, $i = 1, \ldots, d$.

APPLICATIONS AND RELATED INFORMATION

Let $S_1, \ldots, S_d$ be a collection of finite sets, called "colors," in $\mathbb{R}^d$. Assume that the size of each of these sets is n and that the points are all in general position. Then,

according to Akiyama and Alon (see [Bár93]), the ham sandwich theorem implies that there exists a partition of $S := \bigcup_{i=1}^{d} S_i$ into n nonempty, pairwise disjoint sets $D_1, \ldots, D_n$ that are multicolored in the sense that $|D_i \cap S_j| = 1$ for all i and j, such that the simplices conv $D_1, \ldots,$ conv D_n are pairwise disjoint.

11.2.2 THE CENTER POINT THEOREM

THEOREM 11.2.3 *Center Point Theorem*

Let $A \subset \mathbb{R}^d$ be a Lebesgue measurable subset of $\mathbb{R}^d$ or, more generally, one of the measures μ described prior to Theorem 11.2.1. *Then there exists a point $x \in \mathbb{R}^d$ so that for every closed halfspace $P \subset \mathbb{R}^d$, if $x \in P$ then*

$$\text{vol}(P \cap A) \geq \frac{\text{vol}(A)}{d+1}.$$

If formulated for a more general measure μ, the result guarantees that $\mu(P) \geq \mu(\mathbb{R}^d)/(d+1)$ for every closed halfspace $P \ni x$.

TOPOLOGICAL BACKGROUND

If the Borsuk-Ulam theorem is responsible for the ham sandwich theorem, then R. Rado's center point theorem can be seen as a consequence of another well-known topological result, Brouwer's fixed point theorem.

THEOREM 11.2.4 *Brouwer's Fixed Point Theorem*

Every continuous map $f : B \to B$, mapping a ball (or, more generally, a closed convex set) $B \subset \mathbb{R}^d$ into itself, has a fixed point $x = f(x)$.

Very often it is more convenient to use Kakutani's theorem, which is a generalization of Brouwer's theorem to "multivalued functions" $f : B \to B$.

The center point theorem is deduced from Brouwer's theorem roughly as follows. Let $x \in B$, where B is a "large" ball containing the set A. If x is not a center point, then there is a vector $e \in S^{d-1}$ pointing in a direction in which x can be moved to make it closer to being one. In this way we define a function $x \mapsto -f(x)$, and a fixed point, i.e., a point that doesn't need to be moved, is a center point.

Recall that the center point theorem was originally deduced from Helly's convexity theorem, which also has several topological relatives.

APPLICATIONS AND RELATED INFORMATION

According to Miller and Thurston (see [Ede94]), the center point theorem and the Koebe theorem can be used to prove the existence of a small separator for a planar graph, a result proved originally (by Lipton and Tarjan) by different methods.

The center point theorem implies Helly's convexity theorem, and so it can be viewed as a measure-theoretic equivalent of Helly's theorem.

11.2.3 CENTER TRANSVERSAL THEOREM

THEOREM 11.2.5 *Center Transversal Theorem*

Let $A_0, A_1, \ldots, A_k$, $0 \leq k \leq d-1$, be a collection of Lebesgue measurable sets in $\mathbb{R}^d$ or, more generally, let $\mu_0, \mu_1, \ldots, \mu_k$ be a sequence of measures. Then there exists a k-dimensional affine subspace $D \subseteq \mathbb{R}^d$ such that for every closed halfspace $H(v,\alpha) := \{x \in \mathbb{R}^d \mid \langle x, v\rangle \leq \alpha\}$ and every $i \in \{0, 1, \ldots, k\}$,

$$D \subseteq H(v,\alpha) \Longrightarrow m(A_i \cap H(v,\alpha)) \geq \frac{m(A_i)}{d-k+1}.$$

If formulated for a sequence $\mu_0, \ldots, \mu_k$ of more general measures, the result guarantees that $\mu_i(H(v,\alpha)) \geq \mu_i(\mathbb{R}^d)/(d-k+1)$ for all i and all $H(v,\alpha) \supseteq D$.

TOPOLOGICAL BACKGROUND

The center transversal theorem contains the ham sandwich and the center point theorems as two boundary cases. The topological principle that is at the root of this result ought to be strong enough for this purpose. This principle is very similar in spirit to the well known "hairy ball theorem" and comes down to the fact that the cohomology class $(w_k)^{n-k}$ is nonzero, where $w_k \in H^k(G_k(\mathbb{R}^n); \mathbb{Z}_2)$ is the k^{th} Stiefel-Whitney characteristic cohomology class of the canonical k-dimensional vector bundle over the Grassmann manifold $G_k(\mathbb{R}^n)$ of all linear, k-dimensional subspaces of $\mathbb{R}^n$.

APPLICATIONS AND RELATED INFORMATION

The following Helly-type transversal theorem, due to Dol'nikov (see [Eck93]), is a consequence of the same topological principle that is at the root of the center transversal theorem. Moreover, the center transversal theorem is related to Dol'nikov's result in the same way that the center point theorem is related to Helly's theorem.

THEOREM 11.2.6

Let $\mathcal{K}_0, \ldots, \mathcal{K}_k$ be families of compact convex sets. Suppose that for every i, and for each k-dimensional subspace $V \subset \mathbb{R}^d$, there exists a translate V_i of V intersecting every set in $\mathcal{K}_i$. Then there exists a common k-dimensional transversal of the family $\mathcal{K} := \bigcup_{i=0}^{k} \mathcal{K}_i$, i.e., there exists an affine k-dimensional subspace of $\mathbb{R}^d$ intersecting all the sets in $\mathcal{K}$.

11.2.4 EQUIPARTITION OF MASSES BY HYPERPLANES

A measurable set $A \subset \mathbb{R}^3$ can be partitioned by three hyperplanes into 8 pieces of equal measure. This is an instance of the general problem of characterizing all triples (d, j, k) such that for any j mass distributions (measurable sets) in $\mathbb{R}^d$, there exist k hyperplanes, $k \leq d$, such that each of the 2^k "orthants" contains the fraction $1/2^k$ of each of the masses. Such a triple (d, j, k) will be called admissible. For example, the ham sandwich theorem implies that $(d, d, 1)$ is admissible. It is

known (E. Ramos, [Ram96]) that $d \geq j(2^k - 1)/k$ is a necessary condition and $d \geq j2^{k-1}$ a sufficient one for a triple (d, j, k) to be admissible. Ramos's method yields many interesting results in lower dimensions, including the admissibility of the triples $(9, 3, 3)$, $(9, 5, 2)$, and $(5, 1, 4)$. The most interesting special case that still seems to be out of reach is the triple $(4, 1, 4)$. The key idea in these proofs is to use, for this purpose, a specially designed, generalized form of the Borsuk-Ulam theorem for continuous, "even-odd" maps of the form $f : S^{d-1} \times \ldots \times S^{d-1} \to \mathbb{R}^l$.

11.2.5 EQUIPARTITION BY CONVEX CONES

The center transversal theorem is a common generalization of the ham sandwich theorem and the center point theorem. There is another general statement extending the ham sandwich theorem which, as a special boundary case, includes the following result.

PROPOSITION 11.2.7

Let μ be the measure arising from a continuous mass distribution given by an integrable (density) function $f : \mathbb{R}^d \to \mathbb{R}^+$. Let $\Delta := \text{conv}(\{a_i\}_{i=0}^d)$ be a fixed, regular d-dimensional simplex centered at the origin and let $\mathcal{D}(\Delta) = \{D_i\}_{i=0}^d$ be the dissection of $\mathbb{R}^d$ into closed, convex cones $D_i := \text{cone}(\text{conv}(\{a_j\}_{j \neq i}))$. Then there exists $x \in \mathbb{R}^d$ such that

$$\mu(x + D_0) = \ldots = \mu(x + D_d) = \frac{\mu(\mathbb{R}^d)}{d+1}.$$

REMARK 11.2.8

This result is also an offspring of the Kakutani fixed point theorem [VZ92].

Let $\Delta := \text{conv}(\{a_i\}_{i=0}^m)$ be a regular simplex of dimension $m \leq d$ and let $P := \text{aff}\ \Delta$ be its affine hull. Then there arises a dissection $\mathcal{D}(\Delta) = \{D_i\}_{i=0}^m$ of $\mathbb{R}^d$ into $m+1$ wedge-like cones, where $D_i := P^\perp \oplus \text{cone}(\text{conv}(\{a_j\}_{j \neq i}))$.

CONJECTURE 11.2.9

Let $\mu_0, \ldots, \mu_k$ be a family of continuous mass distributions (measures), $0 \leq k \leq d-1$, defined on $\mathbb{R}^d$. Then there exists a $(d-k)$-dimensional regular simplex Δ and the corresponding dissection, $\mathcal{D}(\Delta)$, such that for some $x \in \mathbb{R}^d$, and for all i, j,

$$\mu_i(x + D_j) \geq \frac{\mu_i(\mathbb{R}^d)}{d-k+1}.$$

This conjecture is denoted in [VZ92] by $B(d, k)$. Proposition 11.2.8 implies $B(d, 0)$ and the ham sandwich theorem is $B(d, d-1)$. The conjecture is also confirmed in the case $B(d, d-2)$ for all d. Moreover, a natural topological statement $C(d, k)$ can be formulated that implies $B(d, k)$ and that is closely related to the analogous statement needed for the center transversal theorem. It is not known whether this statement holds in general. Its importance is illustrated by the surprising fact that $C(d, k)$ implies the following form of the well-known Knaster conjecture.

CONJECTURE 11.2.10 *Knaster's Conjecture*

Let $f : S^{n-1} \to \mathbb{R}^k$ be a continuous map and let us assume that $\{a_0, \ldots, a_{n-k}\} \subset S^{n-1}$ are the vertices of a regular simplex Δ centered at the origin. Then there

exists a rotation $o \in O(n)$ *of the ambient space such that* $f(o(a_i)) = f(o(a_j))$ *for all* i, j.

11.2.6 OTHER EQUIPARTITIONS

There are other types of partitions of mass distributions. A "cobweb partition theorem" of Schulman (see [Mat94]) guarantees an equipartition of a plane mass distribution into 8 pieces by an arrangement of lines resembling a cobweb.

A result of Paterson (see [Mat94]) is an interesting example of a ham sandwich-type theorem that deals with partitions of lines rather than of points. It says that for every set of lines in $\mathbb{R}^3$, there exist 3 mutually perpendicular planes such that the interior of each of the resulting octants is intersected by no more than half of the lines.

11.3 TVERBERG-TYPE THEOREMS AND THEIR APPLICATIONS

A collection of seven points in the plane can be partitioned into three nonempty, disjoint subsets so that the corresponding convex hulls have a nonempty intersection. If we add two more points and color all the points with three colors so that each color is equally represented, then there exists a partition of this set of nine colored points into three multicolored three-point sets so that the corresponding multicolored triangles have a nonempty intersection. Something similar is possible in 3-space, but this time we need five points of each color in order to guarantee a partition of this kind. In short, given a constellation of five blue, five red, and five yellow stars in space, it is always possible to form three vertex-disjoint multicolored triangles with nonempty intersection. These are the simplest, nontrivial cases of Tverberg-type theorems which, together with their consequences and most important applications, are shown in Figure 11.3.1.

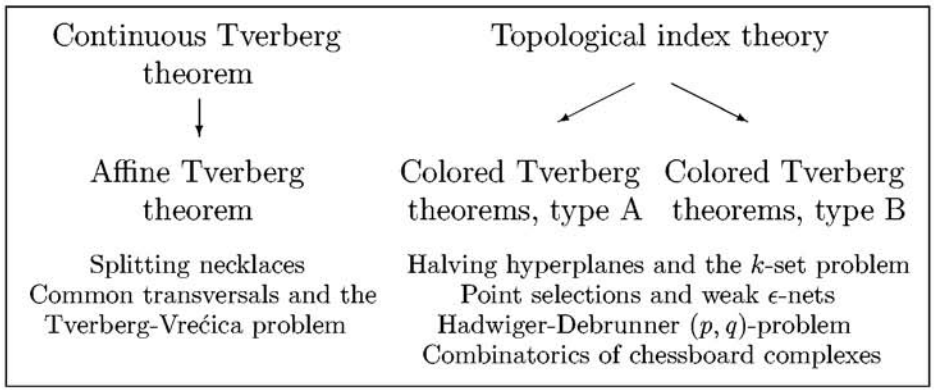


FIGURE 11.3.1
Tverberg-type theorems.

GLOSSARY

Tverberg-type problem: A problem in which a finite set $A \subset \mathbb{R}^d$ is to be partitioned into nonempty, disjoint pieces $A_1, \ldots, A_p$, possibly subject to some additional constraints, so that the corresponding convex hulls $\{\text{conv}(A_i)\}_{i=1}^p$ intersect.

Colors: A set of $k+1$ colors is a collection $\mathcal{C} = \{C_0, \ldots, C_k\}$ of disjoint subsets of $\mathbb{R}^d$, $d \geq k$. A set $B \subset \mathbb{R}^d$ is ***multicolored*** if it contains a point from each of the sets C_i; in this case conv B is called a ***rainbow simplex*** (possibly degenerate).

Type A and ***type B:*** Colored Tverberg problems are of type A or type B depending on whether $k = d$ or $k < d$ (resp.), where $k+1$ is the number of colors.

Tverberg numbers $T(r,d)$, $T(r,k,d)$: $T(r,k,d)$ is the minimal size of each of the colors C_i, $i = 0, \ldots, k$, that guarantees that there always exist r intersecting rainbow simplices. $T(r,d) := T(r,d,d)$.

11.3.1 MONOCHROMATIC TVERBERG THEOREMS

THEOREM 11.3.1 *Affine Tverberg Theorem*

Every set $K = \{a_j\}_{j=0}^{(q-1)(d+1)} \subset \mathbb{R}^d$ *with* $(d+1)(q-1)+1$ *elements can be partitioned into* q *nonempty, disjoint subsets* $K_1, \ldots, K_q$ *so that the corresponding convex hulls have nonempty intersection:*

$$\bigcap_{i=1}^{q} \text{conv}\,(K_i) \neq \emptyset\,.$$

(The special case $q = 2$ is Radon's theorem: see Chapter 4.)

THEOREM 11.3.2 *Continuous Tverberg Theorem*

Let Δ^m *be an* m*-dimensional simplex and assume that* q *is a prime integer. Then for every continuous map* $f : \Delta^{(q-1)(d+1)} \to \mathbb{R}^d$ *there exist vertex-disjoint faces* $\Delta^{t_1}, \ldots, \Delta^{t_q} \subset \Delta^{(q-1)(d+1)}$ *such that* $\bigcap_{i=1}^{q} f(\Delta^{t_i}) \neq \emptyset$.

APPLICATIONS AND RELATED INFORMATION

The affine Tverberg theorem was proved by Helge Tverberg in 1966. The continuous Tverberg theorem, proved by Bárány, Shlosman, and Szücs, reduces to the affine version if f is an affine (simplicial) map. The relevant references for these two theorems are [Bár93], [Bjö95], [Eck93], and [Mat94].

The following "necklace-splitting theorem" is a very nice application of the continuous Tverberg theorem.

THEOREM 11.3.3

Assume that an open necklace has ka_i *beads of color* i, $1 \leq i \leq t$, $k \geq 2$. *Then it is possible to cut this necklace at* $t(k-1)$ *places and assemble the resulting intervals into* k *collections, each containing exactly* a_i *beads of color* i.

An interesting connection has emerged recently between the ham sandwich and Tverberg-type problems. An example of this is the so-called Tverberg-Vrećica con-

jecture, which incorporates both the center transversal theorem, Theorem 11.2.6, and the (affine) Tverberg theorem in a single general statement.

CONJECTURE 11.3.4

Assume that $0 \le k \le d-1$ *and let* $S_0, S_1, \ldots, S_k$ *be a collection of finite sets in* $\mathbb{R}^d$ *of given cardinalities* $|S_i| = (r_i - 1)(d-k+1)+1$, $i = 0, 1, \ldots, k$. *Then* S_i *can be split into* r_i *nonempty sets,* $S_i^1, \ldots, S_i^{r_i}$, *so that for some* k*-dimensional affine subspace* $D \subset \mathbb{R}^d$, $D \cap \operatorname{conv}(S_i^j) \neq \emptyset$ *for all* i *and* j, $0 \le i \le k$, $1 \le j \le r_i$.

11.3.2 COLORED TVERBERG THEOREMS

Let $T(r,k,d)$ be the minimal number t so that for every collection of colors $\mathcal{C} = \{C_0, \ldots, C_k\}$ with the property $|C_i| \ge t$ for all $i = 0, \ldots, k$, there exist r multicolored sets $A_i = \{a_j^i\}_{j=0}^k$, $i = 1, \ldots, r$, that are pairwise disjoint but where the corresponding rainbow simplices $\sigma_i := \operatorname{conv} A_i$ have a nonempty intersection, $\bigcap_{i=1}^r \sigma_i \neq \emptyset$.

The colored Tverberg problem is to establish the existence of, and then to evaluate or estimate, the integer $T = T(r,k,d)$. The cases $k = d$ and $k < d$ are related, but there is also an essential difference. In the case $k = d$, provided t is large enough, the number of intersecting rainbow simplices can be arbitrarily large. In the case $k < d$, for dimension reasons, one cannot expect more than $r \le d/(d-k)$ intersecting k-dimensional rainbow simplices. This is the reason why colored Tverberg theorems are classified as type A or type B, depending on whether $k = d$ or $k < d$.

In the type A case, where $T(r,d,d)$ is abbreviated simply as $T(r,d)$, it is easy to see that a lower bound for this function is r. It is conjectured that this lower bound is attained:

CONJECTURE 11.3.5 *(Type A)*

$T(r,d) = r$ *for all* r *and* d.

REMARK 11.3.6

This conjecture has been confirmed for $r = 2$ *and for* $d \le 2$ [Bár93].

The following theorem of Živaljević and Vrećica (see [Bár93] and [Mat94]) provides the best bounds known in the general case. It implies that $T(r,d) \le 4r - 3$ for all r and d.

THEOREM 11.3.7 *(Type A)*

For every integer r *and every collection of* $d+1$ *disjoint sets* $C_0, C_1, \ldots, C_d$ *in* $\mathbb{R}^d$, *called colors, each of cardinality at least* $4r-3$, *there exist* r *disjoint, multicolored subsets* $S_i \subseteq \bigcup_{i=0}^d C_i$ *such that*

$$\bigcap_{i=1}^{r} \operatorname{conv} S_i \neq \emptyset.$$

If r *is a prime number then it suffices to assume that the size of each of the colors is at least* $2r-1$. *In other words,* $T(r,d) \le 2r-1$ *if* r *is a prime and* $T(r,d) \le 4r-3$ *in the general case.*

In the Type B case, let us assume that $r \le d/(d-k)$, which is a necessary condition for a colored Tverberg theorem of Type B.

CONJECTURE 11.3.8 *(Type B)*

$T(r,k,d) = 2r-1$.

REMARK 11.3.9

There exist examples that show that $T(r,k,d) \geq 2r-1$.

The following theorem [VZ94] confirms the conjecture above for the case of a prime number r.

THEOREM 11.3.10 *(Type B)*

Let $C_0,\ldots,C_k$ *be a collection of* $k+1$ *disjoint finite sets (colors) in* $\mathbb{R}^d$. *Let* r *be a prime integer such that* $r \leq d/(d-k)$ *and let* $|C_i| = t \geq 2r-1$. *Then there exist* r *multicolored* k*-dimensional simplices* S_i, $i = 1,\ldots,r$, *that are pairwise vertex-disjoint such that*

$$\bigcap_{i=1}^{r} \text{conv } S_i \neq \emptyset.$$

The usual price for using topological (equivariant) methods is the extra assumption that r is a prime number. On the other hand, the results obtained by these methods hold in greater generality and include nonlinear versions of Theorems 11.3.7 and 11.3.10. In short, the results above hold for more general convexity structures in the sense of the following definition.

Definition:

Let $S \subset \mathbb{R}^d$ be a finite set of points. A ***convexity structure*** on S is a continuous map $f : \Delta \to \mathbb{R}^d$, where Δ is a k-dimensional simplex, $k = |S|-1$, and where f maps the vertices of Δ onto the set S. The ***convex hull*** of a set $T \subset S$ is, by definition,

$$\text{conv}_f(T) := \bigcup \{f(\text{conv}(f^{-1}(P)) \mid P \subset T, |P| \leq d+1\}.$$

We note that the usual convexity structure is obtained if the map f is assumed to be affine. Informally speaking, the convex hull of two points a and b is a curve connecting a and b, and the convex hull of three points is a curved triangle with vertices at a, b, and c so that the side determined by vertices a and b is the curved convex hull of these two points, and so on.

EXAMPLE 11.3.11

The simplest instance of Theorem 11.3.10 *is the case* $d = 2$, $k = 1$, *and* $r = 2$. *Then, in the nonlinear version of this theorem, we recognize the well-known fact that the complete bipartite graph* $K_{3,3}$ *is not planar. This is one of the earliest results in topology, already known to Euler, who formulated it as a problem about three houses and three wells.*

11.3.3 APPLICATIONS OF COLORED TVERBERG THEOREMS

Theorem 11.3.7 provides a general bound of the form $T(d+1,d) \leq 4d+1$, which has opened the possibility of proving many interesting results in discrete and computational geometry.

HALVING HYPERPLANES AND THE k-SET PROBLEM

The number $h_d(n)$ of halving hyperplanes of a set of size n in $\mathbb{R}^d$, i.e., the number of essentially distinct placements of a hyperplane that split the set in half, according to Bárány, Füredi, and Lovász (see [Bár93]), satisfies

$$h_d(n) = O(n^{d-\epsilon_d}), \quad \text{where} \quad \epsilon_d = T(d+1,d)^{-(d+1)}.$$

POINT SELECTIONS AND WEAK ϵ-NETS

The equivalence of the following statements was established in [ABFK92] before Theorem 11.3.7 was proved, and illustrates the significance of this result.

- Weak colored Tverberg theorem: $T(d+1,d)$ is finite.
- Point selection theorem: There exists a constant $s = s_d$, whose value depends on the bound for $T(d+1,d)$, such that any family $\mathcal{H}$ of $(d+1)$-element subsets of a set $X \subset \mathbb{R}^d$, of size $|\mathcal{H}| = p\binom{|X|}{d+1}$, contains a pierceable subfamily $\mathcal{H}'$ such that $|\mathcal{H}'| \gg p^s\binom{|X|}{d+1}$. ($\mathcal{H}'$ is ***pierceable*** if $\bigcap_{S\in\mathcal{H}'} \operatorname{conv} S \neq \emptyset$. $A \gg_d B$ if $A \geq c_1(d)B + c_2(d)$, where $c_1(d) > 0$ and $c_2(d)$ are constants depending only on the dimension d.)
- Weak ϵ-net theorem: For any $X \subset \mathbb{R}^d$ there exists a weak ϵ-net F with $|F| \ll_d \epsilon^{(d+1)(1-1/s)}$, where $s = s_d$ is as above. (See Chapter 34 for the notion of ϵ-net; a *weak* ϵ-net is similar, except that it need not be part of X.)
- Hitting set theorem: For every $\eta > 0$ and every $X \subset \mathbb{R}^d$ there exists a set $E \subset \mathbb{R}^d$ that misses at most $\eta\binom{|X|}{d+1}$ simplices of X and has size $|E| \ll_d \eta^{1-s_d}$, where s_d is as above.

OTHER RELATED RESULTS

The weak ϵ-net theorem was among the results used in the Alon-Kleitman proof of the Hadwiger-Debrunner (p,q)-conjecture (see Chapter 4).

A topological configuration space that arises in proofs of Theorems 11.3.7 and 11.3.10 is the so-called ***chessboard complex***, $\Delta_{r,t}$, which owes its name to the fact that it can be described as the complex of all nontaking rook placements on an $r \times t$ chessboard. This is an interesting combinatorial object, which arises independently as the coset complex of the symmetric group, as the complex of partial matchings in a complete bipartite graph, and as the complex of all partial injective functions.

11.3.4 TOOLS FROM EQUIVARIANT TOPOLOGY

The method of equivariant maps is a versatile tool for proving results in geometry and combinatorics and, in particular, lies behind the proofs of many results of Tverberg type.

GLOSSARY

G-space X, G-action: A group G acts on a space X if each element of G is a continuous transformation of X and multiplication in G corresponds to

composition of transformations. Formally, a G-action α is a continuous map $\alpha : G \times X \to X$ such that $\alpha(g, \alpha(h, x)) = \alpha(gh, x)$. Then X is called a G-space and $\alpha(g, x)$ is often abbreviated as $g \cdot x$ or gx.

Free G-action: An action is free if $g \cdot x = x$ for some $x \in X$ implies $g = e$, where e is the unit element in G.

G-equivariant map: A map $f : X \to Y$ of two G-spaces X and Y is equivariant if for all $g \in G$ and $x \in X$, $f(g \cdot x) = g \cdot f(x)$.

Borsuk-Ulam type theorem: Any theorem establishing the nonexistence of a G-equivariant map between two G-spaces X and Y.

n-connected space: A path-connected and simply connected space with trivial homology in dimensions $1, 2, \ldots, n$. A path-connected space X is simply connected if every closed loop $\omega : S^1 \to X$ can be continuously deformed to a point.

The following generalization of the Borsuk-Ulam theorem is the key result used in proofs of many Tverberg-type statements.

THEOREM 11.3.12

Suppose X and Y are simplicial (more generally, CW) complexes equipped with the free action of a finite group G, and that X is m-connected, where $m = \dim Y$. Then there does not exist a G-equivariant map $f : X \to Y$.

EXAMPLE 11.3.13

If $X = S^n$ and $Y = S^{n-1}$ are spheres, then the involution $x \mapsto -x$ is an action of the group $\mathbb{Z}_2$, and a $\mathbb{Z}_2$-equivariant map $f : S^n \to S^{n-1}$ is just an odd map: $f(-x) = -f(x)$. In this case, Theorem 11.3.12 specializes to the second form of the Borsuk-Ulam theorem given in Section 11.2 (following Theorem 11.2.2).

Theorem 11.3.12 is strong enough for the majority of applications. Nevertheless, in some cases more sophisticated tools are needed. *A topological index theory* is a complexity theory for G-spaces that allows us to conclude that there does not exist a G-equivariant map $f : X \to Y$ if the G-space Y is of *larger complexity* than the G-space X. The complexity, or the *equivariant index* $\mathrm{Ind}_G(X)$ of X, is usually a positive integer. However, in some applications it is more natural, and sometimes essential, to use more sophisticated partially ordered sets of G-degrees of complexity. A fairly self-contained description of an elementary index theory of this kind, with an emphasis on applications in combinatorics, is given in [Živ96].

11.4 SOURCES AND RELATED MATERIAL

FURTHER READING

The reader will find additional information about applications of topological methods in discrete geometry and combinatorics, as well as a more comprehensive bibliography, in review papers [Alo88], [Bár93], [Bjö95], [Eck93], [Mat94], [Ste85], and [Živ96]. Reviews [Bár93] and [Mat94] overlap partially with our exposition and,

along with the ham sandwich and Tverberg type theorems, discuss other questions such as the van Kampen-Flores type results, the Kneser conjecture, etc. The chapter in "Handbook of Combinatorics" written by A. Björner, [Bjö95], offers a broader perspective on topological applications in combinatorics. This review illuminates other topological methods and techniques and provides numerous examples of interesting applications, including applications of the "Hard Lefschetz Thereom," the fixed points in posets, etc. The book [Mun84] serves as a standard textbook of algebraic topology. The review [Živ96] can be viewed as an expanded version of our Section 11.3.4, and serves as a fairly self-contained account and a "user-friendly" guide for applications of equivariant topological methods in combinatorics.

RELATED CHAPTERS

Chapter 1: Finite point configurations
Chapter 4: Helly-type theorems and geometric transversals

REFERENCES

[Alo88] N. Alon. Some recent combinatorial applications of Borsuk-type theorems. In M.M. Deza, P. Frankl, and D.G. Rosenberg, editors, *Algebraic, Extremal, and Metric Combinatorics*, pages 1–12. Cambridge University Press, 1988.

[ABFK92] N. Alon, I. Bárány, Z. Füredi, and D. Kleitman. Point selections and weak ϵ-nets for convex hulls. *Combin. Probab. Comput.*, 1:189-200, 1992.

[Bár93] I. Bárány. Geometric and combinatorial applications of Borsuk's theorem. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, Volume 10 of *Algorithms Combin.*, pages 235–249. Springer-Verlag, Berlin, 1993.

[Bjö95] A. Björner. Topological methods. In R. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, pages 1819–1872. North-Holland, Amsterdam, 1995.

[Eck93] J. Eckhoff. Helly, Radon, and Carathéodory type theorems. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 389–448. North-Holland, Amsterdam, 1993.

[Ede94] H. Edelsbrunner. *Lectures in Geometry and Algorithms.* Lecture notes. Urbana, 1994.

[Mat94] J. Matoušek. *Topological Methods in Combinatorics and Geometry.* Lecture notes. Prague, 1994.

[Mun84] J.R. Munkres. *Elements of Algebraic Topology.* Addison-Wesley, Menlo-Park, 1984.

[Ram96] E. Ramos. Equipartitions of mass distributions by hyperplanes. *Discrete Comput. Geom.*, 15:147–167, 1996.

[Ste85] H. Steinlein. Borsuk's antipodal theorem and its generalizations and applications: a survey. In *Topological Methods in Nonlinear Analysis*, volume 95 of *Sém. Math. Sup.*, pages 166–235. Presses de l'Université de Montréal, 1985.

[VZ92] S. Vrećica and R. Živaljević. The ham sandwich theorem revisited. *Israel J. Math.*, 78:21–32, 1992.

[VZ94] S. Vrećica and R. Živaljević. New cases of the colored Tverberg theorem. In H. Barcelo and G. Kalai, editors, *Jerusalem Combinatorics '93*, pages 325–334. Volume 178 of *Contemp. Math.*, Amer. Math. Soc., Providence, 1994.

[Živ96] R. Živaljević. *User's Guide to Equivariant Methods in Combinatorics.* Volume 59(73) of *Publ. Inst. Math. (Beograd) (N.S.)*, Inst. Math., Belgrade, 1996.

12 POLYOMINOES

David A. Klarner

INTRODUCTION

A *polyomino* is a finite, connected subgraph of the square-grid graph consisting of infinitely many unit cells matched edge-to-edge, with pairs of adjacent cells forming edges of the graph. Polyominoes have a long history, going back to the start of the 20th century, but they were popularized in the present era initially by Solomon Golomb, then by Martin Gardner in his *Scientific American* columns "Mathematical Games." They now constitute one of the most popular subjects in mathematical recreations, and have found interest among mathematicians, physicists, biologists, and computer scientists as well.

12.1 BASIC CONCEPTS

GLOSSARY

Cell: A unit square in the Cartesian plane with its sides parallel to the coordinate axes and with its center at an integer point (u, v). This cell is denoted $[u, v]$ and identified with the corresponding member of $\mathbb{Z}^2$.

Adjacent cells: Two cells, $[u, v]$ and $[r, s]$, with $|u - r| + |v - s| = 1$.

Square-grid graph: The graph with vertex set $\mathbb{Z}^2$ and an edge for each pair of adjacent cells.

Polyomino: A finite set S of cells such that the induced subgraph of the square-grid graph with vertex set S is connected. A polyomino with exactly n cells is called an ***n-omino***. Polyominoes are also known as ***animals***.

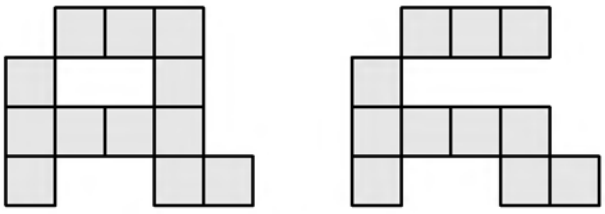


FIGURE 12.1.1
Two sets of cells: the set on the left is a polyomino, the one on the right is not.

12.2 EQUIVALENCE OF POLYOMINOES

Notions of equivalence for polyominoes are defined in terms of groups of affine maps that act on the set $\mathbb{Z}^2$ of cells in the plane.

GLOSSARY

Translation by (r, s)***:*** The mapping from $\mathbb{Z}^2$ to itself that maps $[u, v]$ to $[u + r, v + s]$; it sends any subset $S \subset \mathbb{Z}^2$ to its *translate* $S + (r, s) = \{[u + r, v + s] : [u, v] \in S\}$.

Translation-equivalent: Sets S, S' of cells such that S' is a translate of S.

Fixed polyomino: A translation-equivalence class of polyominoes; $\boldsymbol{t(n)}$ denotes the number of fixed n-ominoes.

Representatives of the six fixed 3-ominoes are shown in Figure 12.2.1.

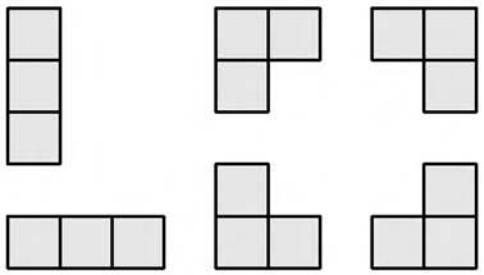


FIGURE 12.2.1
The six fixed 3-ominoes.

Lexicographically minimum cell: The unique member $[u, v]$ of a finite set $S \subset \mathbb{Z}^2$ with $v = \min\{v' : [u', v'] \in S\}$, $u = \min\{u' : [u', v] \in S\}$.

Standard position: The translate $S - (u, v)$ of S, where $[u, v]$ is the lexicographically minimum cell in S.

Rotation-translation group: The group $\mathcal{R}$ of mappings of $\mathbb{Z}^2$ to itself of the form $[u, v] \mapsto [u, v]\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}^k + (r, s)$. (The matrix $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$, which is denoted by R, maps $[u, v]$ to $[v, -u]$ by right multiplication, hence represents a clockwise rotation of 90°.)

Rotationally equivalent: Sets S, S' of cells with $S' = \rho S$ for some $\rho \in \mathcal{R}$.

Chiral polyomino, or ***handed polyomino:*** A rotational-equivalence class of polyominoes; $\boldsymbol{r(n)}$ denotes the number of chiral n-ominoes.

The top row of 5-ominoes in Figure 12.2.2 consists of the set of cells $F = \{[0, -1], [-1, 0], [0, 0], [0, 1], [1, 1]\}$, together with FR, FR^2, and FR^3. All four of these 5-ominoes are rotationally equivalent. The bottom row in Figure 12.2.2 shows these same four 5-ominoes reflected about the x-axis. These four 5-ominoes are rotationally equivalent as well, but none of them is rotationally equivalent to any of the 5-ominoes shown in the top row. Representatives of the seven chiral 4-ominoes are shown in Figure 12.2.3.

Congruence group: The group $\mathcal{S}$ of motions generated by the matrix $M = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ (reflection in the x-axis) and the rotation-translation group $\mathcal{R}$. (A typical element of $\mathcal{S}$ has the form $[u, v] \mapsto [u, v]R^k M^i + (r, s)$, for some $k = 0, 1, 2$, or 3, some $i = 0$ or 1, and some $r, s \in \mathbb{Z}$.)

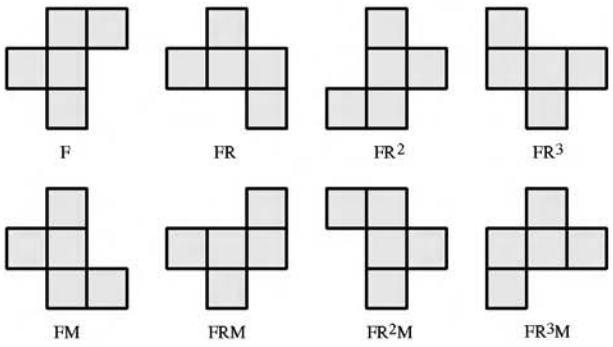


FIGURE 12.2.2
The 5-ominoes in the top row are rotationally equivalent, and so are their reflections in the bottom row, but the two sets are rotationally distinct.

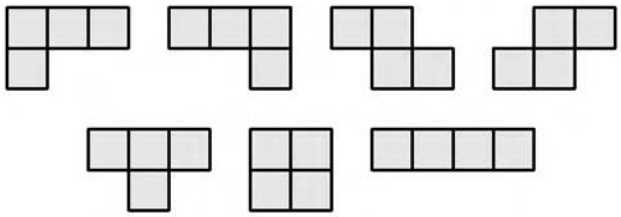


FIGURE 12.2.3
The seven chiral 4-ominoes.

Congruent: Sets S, S' of cells such that $S' = \sigma(S)$ for some $\sigma \in \mathcal{S}$.

Free polyomino: A congruence class of polyominoes; $\boldsymbol{s(n)}$ denotes the number of free n-ominoes.

The twelve free 5-ominoes are shown in Figure 12.2.4.

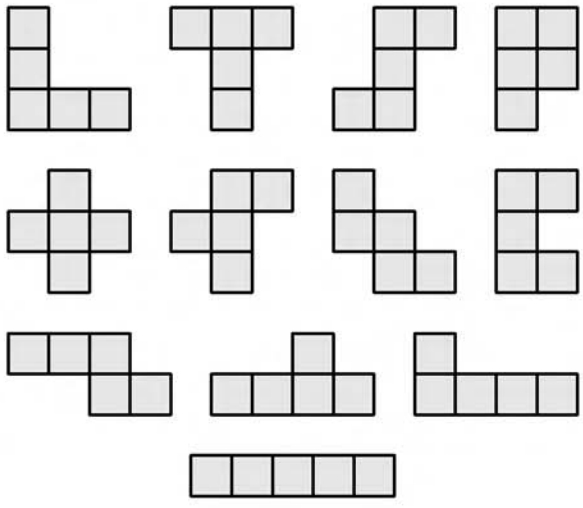


FIGURE 12.2.4
The twelve free 5-ominoes.

Standard position: A finite set $S \subset \mathbb{Z}^2$ is in standard position if and only if $[0,0] \in S$, $0 \leq v$ for all $[u,v] \in S$, and $0 \leq u$ for all $[u,0] \in S$.

THEOREM 12.2.1 *Embedding Theorem*

For each n, let U_n consist of the $n^2 - n + 1$ cells of the form $[u, v]$, where $\begin{cases} 0 \le u \le n, & \text{for } v = 0 \\ |u| + v \le n, & \text{for } v > 0 \end{cases}$. (See Figure 12.2.5 for the case $n = 5$.) *Then every n-omino in standard position is a subset of U_n.*

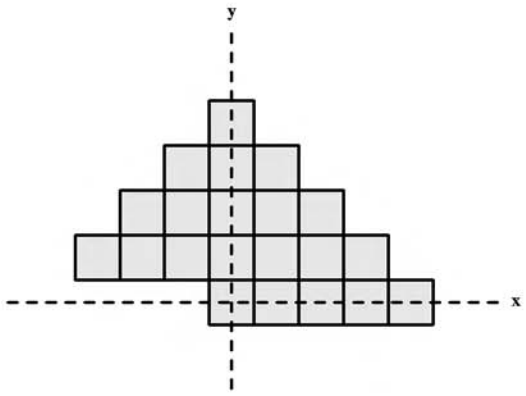


FIGURE 12.2.5
A set of $n^2 - n + 1$ cells that contains every n-omino in standard position.

COROLLARY 12.2.2

The number of fixed n-ominoes is finite for each n.

12.3 HOW MANY *n*-OMINOES ARE THERE?

Table 12.3.1, calculated by Redelmeier [Red81], indicates the values of $t(n)$, $r(n)$, and $s(n)$ for $n = 1, \ldots, 24$.

The values seem to be growing exponentially, and indeed they have exponential bounds. It is easy to see that for each n,

$$\frac{t(n)}{8} \le s(n) \le r(n) \le t(n),$$

and results of Klarner and Rivest [KR73], and of Klarner and Satterfield [KS], using automata theory and building on earlier work of Eden, Klarner, and Read, have shown:

THEOREM 12.3.1

$\lim_{n\to\infty}(t(n))^{1/n} = \theta$ *exists, and* $3.9 < \theta < 4.65$.

ALGORITHMS

Considerable effort has been expended to find a formula for the number of fixed n-ominoes (say), with no success. Redelmeier's algorithm, which produced the entries

TABLE 12.3.1 The number of fixed, chiral, and free n-ominoes for $n \leq 24$.

n	$t(n)$	$r(n)$	$s(n)$
1	1	1	1
2	2	1	1
3	6	2	2
4	19	7	5
5	63	18	12
6	216	60	35
7	760	196	108
8	2725	704	369
9	9910	2500	1285
10	36446	9189	4655
11	135268	33896	17073
12	505861	126759	63600
13	1903890	476270	238591
14	7204874	1802312	901971
15	27394666	6849777	3426576
16	104592937	26152418	13079255
17	400795844	100203194	50107909
18	1540820542	385221143	192622052
19	5940738676	1485200848	742624232
20	22964779660	5741256764	2870671950
21	88983512783	22245940545	11123060678
22	345532572678	86383382827	43191857688
23	1344372335524	336093325058	168047007728
24	5239988770268	1309998125640	654999700403

in Table 12.3.1 (and took over ten months of computer time to run), generates the fixed n-ominoes one by one and counts them. Although the running time is (necessarily) exponential, the algorithm takes only $O(n)$ space. There are other algorithms available [KS], but none has subexponential running time. At present, the computation of $t(n)$ for $n > 30$ seems intractable.

UNSOLVED PROBLEMS

PROBLEM 12.3.2

Can $t(n)$ be computed by a polynomial-time algorithm?

A related problem concerns the constant θ defined above:

PROBLEM 12.3.3

Is there a polynomial algorithm to find, for each n, an approximation θ_n of θ satisfying

$$10^{-n} < |\theta_n - \theta| < 10^{-n+1} \ ?$$

The lower-bound method of [KS1] gives an algorithm for approximating θ from below that has exponential complexity; no such method is known for approximating θ from above.

PROBLEM 12.3.4

Define some decreasing sequence $\beta = (\beta_1, \beta_2, \ldots)$ that tends to θ, and give an algorithm to compute β_n for every n.

It is known that $(t(n))^{1/n} \leq \theta$ for all n, and it seems that the ratios $\tau(n) = t(n+1)/t(n)$ increase for all n. If the latter is true, $\tau(n)$ would approach θ from below. This gives two more unsolved problems:

PROBLEM 12.3.5

Show that $(t(n))^{1/n} < (t(n+1))^{1/(n+1)}$ for all n.

PROBLEM 12.3.6

Show that $\tau(n) < \tau(n+1)$ for all n.

12.4 GENERATING POLYOMINOES

The algorithm we describe to generate all n-ominoes, which is essentially due to Redelmeier [Red81], also provides a way of encoding n-ominoes. Starting with all n-ominoes in standard position, with each cell and each neighboring cell numbered, it constructs all numbered $(n+1)$-ominoes in standard position.

GLOSSARY

Border cell of an n-omino S: A cell $[u, v]$, with $v \geq 0$ or with $v = 0$ and $u \geq 0$, adjacent to some cell of S. The set of all border cells, which is denoted by $B(S)$, can be shown by induction to have no more than $2n$ elements.

The ***lexicographic cell ordering*** $\prec$ on $\mathbb{Z}^2$ is defined by: $[r, s] \prec [u, v]$ if $s < v$, or if $s = v$ and $r < u$.

The algorithm, illustrated in Figure 12.4.1 for $n = 1$, 2, and 3, begins with cell 1 in position $[0, 0]$, with its border cells marked 2 and 3, and then adds these—one at a time—each time numbering *new* border cells in their lexicographic order. Whenever a number used for a border cell is not larger than the largest internal number, it is circled, and the corresponding cell is *not* added at the next stage.

Figure 12.4.2 shows all the 4-ominoes produced in this way, with their border cells marked for the next step of the algorithm.

This process assigns a unique set of positive integers to each n-omino S, also illustrated in Figure 12.4.2. The *set character functions* for these integer sets, in turn, truncated after their final 1's, provide a binary codeword $\chi(S)$ for each n-omino S. For example, the code words for the first three 4-ominoes in Figure 12.4.2 would be 1111, 11101, and 111001.

FIGURE 12.4.1

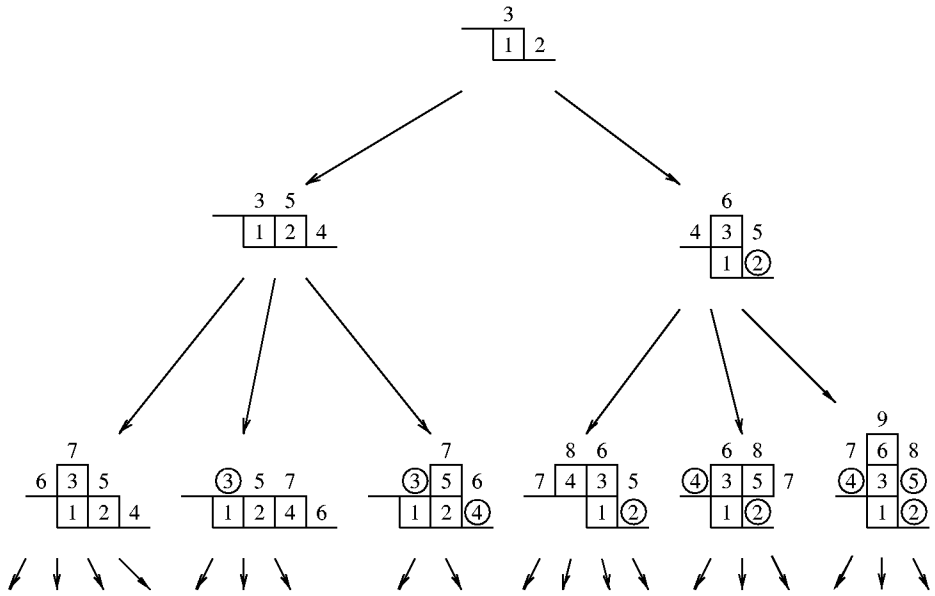
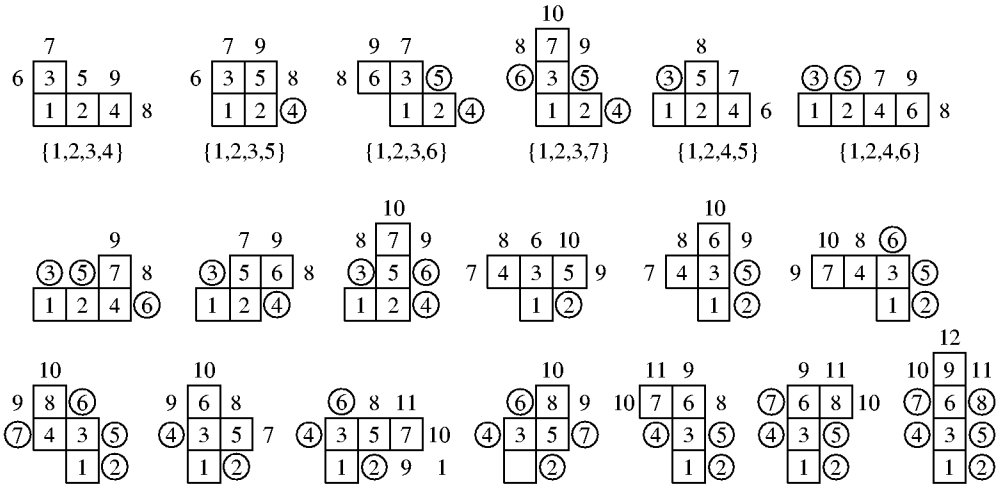


FIGURE 12.4.2



PROBLEM 12.4.1

Which binary strings arise as codewords for n-ominoes?

The following is easy to see:

THEOREM 12.4.2

$t(n+1) = \sum n + |B(S)| - |\chi(S)|$, where the sum extends over all n-ominoes S in standard position, and $|\chi(S)|$ is the number of bits in the codeword of S.

PROBLEM 12.4.3

Is the generating function $T(z) = \sum_{n=1}^{\infty} t(n)z^n$ *a rational function? Is* $T(z)$ *even algebraic?*

12.5 SPECIAL TYPES OF POLYOMINOES

Particular kinds of polyominoes arise in various contexts. We will look at several of the most interesting ones.

GLOSSARY

A ***composition*** of n with k parts is an ordered k-tuple $(p_1, \ldots, p_k)$ of positive integers with $p_1 + \ldots + p_k = n$.

A polyomino is called ***row-convex*** if every (horizontal) row consists of a single strip of cells. It is ***row-column-convex*** if this holds for every column as well.

Simply connected polyomino: A polyomino without holes. (Golumb calls these nonholey polyominoes ***profane***.)

A ***width-k*** polyomino: One each of whose vertical cross sections fits in a $k \times 1$ strip of cells.

A ***directed*** polyomino is defined recursively as follows: Any single cell is a directed polyomino. An $(n+1)$-omino is directed if it can be obtained by adding a new cell immediately above, or to the right of, a cell belonging to some directed n-omino.

COMPOSITIONS AND ROW-CONVEX POLYOMINOES

There is a natural 1-1 correspondence between compositions of n and a certain class of n-ominoes in standard position, as indicated in Figure 12.5.1 for the case $n = 4$.

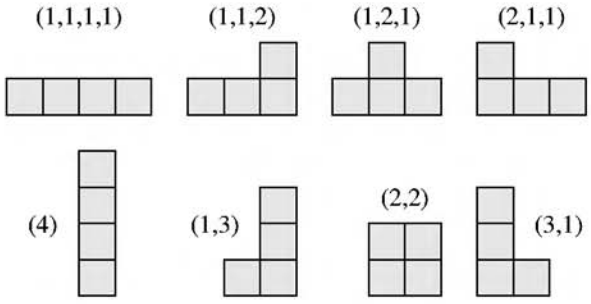


FIGURE 12.5.1
Compositions of 4 *corresponding to certain* 4*-ominoes.*

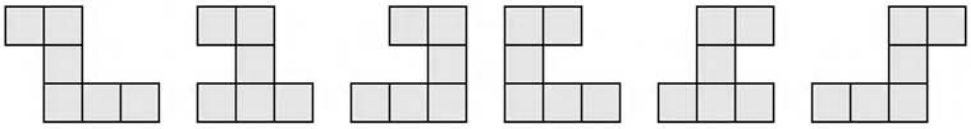
Let us, instead, assign to each composition $(a_1, \ldots, a_k)$ of n an n-omino with a *horizontal* strip of a_i cells in row i. This can be done in many ways, and the results are all the row-convex n-ominoes. Since there are $m + n - 1$ ways to form an $(m+n)$-omino by placing a strip of n cells atop a strip of m cells, it follows that

for each composition $(a_1, \ldots, a_k)$ of n into positive parts, there are

$$(a_1 + a_2 - 1)(a_2 - a_3 - 1) \cdots (a_{k-1} - a_k - 1)$$

n ominoes having a strip of a_i cells in the ith row for each i (see Figure 12.5.2 for an example arising from the composition $6 = 3 + 1 + 2$).

FIGURE 12.5.2
The 6 row-convex 6-ominoes corresponding to the composition $(3, 1, 2)$ *of* 6.



It follows that if $b(n)$ is the number of row-convex n-ominoes, then

$$b(n) = \sum (a_1 + a_2 - 1)(a_2 - a_3 - 1) \cdots (a_{k-1} - a_k - 1),$$

where the sum extends over all compositions $(a_1, \ldots, a_k)$ of n into k parts, for all k. $b(n)$, and the generating function $B(z) = \sum_{n=1}^{\infty} b(n) z^n$, are given by

THEOREM 12.5.1 [Kla67]

$b(n+3) = 5b(n+2) - 7b(n+1) + 4b(n)$, and $B(z) = \dfrac{z(1-z)^3}{1 - 5z + 7z^2 - 4z^3}$.

COROLLARY 12.5.2

$\lim_{n\to\infty}(b(n))^{1/n} = \beta$, *where* β *is the largest real root of* $z^3 - 5z^2 + 7z - 4 = 0$; $3.20 < \beta < 3.21$.

ROW-COLUMN-CONVEX POLYOMINOES

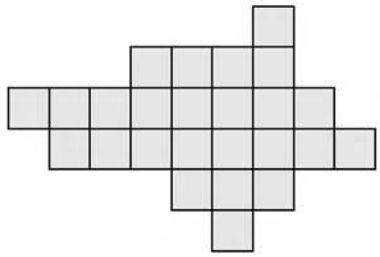


FIGURE 12.5.3
A typical row-column-convex polyomino.

The problem of finding the number, $c(n)$, of row-column-convex polyominoes was first posed by D. Knuth [Knu72]. The existence of a generating function for $c(n)$ with special properties, proved in [KR74], enabled Bender to prove the following asymptotic formula:

THEOREM 12.5.3 [Ben74]

$c(n) \sim cg^n$, *where* $c = 2.67564...$ *and* $g = 2.30914....$

The following problem concerns polyominoes radically different from row-column-convex ones.

PROBLEM 12.5.4

Find the smallest natural number n such that there exists an n-omino with no row or column consisting of just a single strip of cells. (An example of a 21-omino with this property is shown in Figure 12.5.4.)

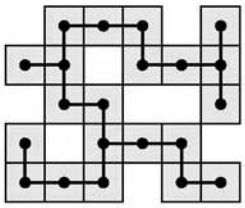


FIGURE 12.5.4
A 21-omino with no row or column a single strip of cells.

SIMPLY CONNECTED POLYOMINOES

Let $t^*(n)$, $s^*(n)$, and $r^*(n)$ denote the numbers of profane fixed, free, and chiral n-ominoes, respectively. Not much is known about their values.

PROBLEM 12.5.5

Compute $t^*(n)$, $s^*(n)$, and $r^*(n)$ for as many values of n as possible.

It is easy to see that $(t^*(n))^{1/n}$, $(s^*(n))^{1/n}$, and $(r^*(n))^{1/n}$ all approach the same limit, θ^*, as $n \to \infty$, and that $\theta^* \leq \theta$ $(= \lim_{n\to\infty}(t(n))^{1/n}$ as defined in Section 12.3).

PROBLEM 12.5.6 [Gol]

Does $\theta^* = \theta$?

We conjecture that the answer is "no".

WIDTH-k POLYOMINOES

A typical width-3 polyomino is shown in Figure 12.5.5.

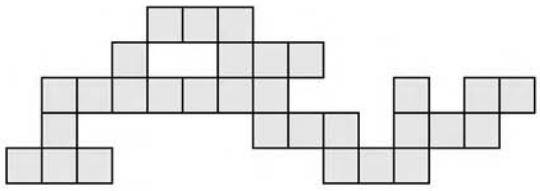


FIGURE 12.5.5
A width-3 polyomino.

THEOREM 12.5.7 [Rea62, KS]

Let $t(n,k)$ be the number of fixed width-k n-ominoes, and $T_k(z) = \sum_{n=1}^{\infty} t(n,k)z^n$. Then $T_k(z) = P_k(z)/Q_k(z)$ for some polynomials $P_k(z), Q_k(z)$ with integer coefficients, no common zeroes, and $Q_k(0) = 1$. Equivalently, the sequence $t(n,k)$, $n = 1, 2, \ldots$, satisfies a linear, homogeneous difference equation with constant coefficients for each fixed k; the order of the equation is roughly 3^k. Furthermore, the sequence $(t(n,k))^{1/n}$ converges to a limit τ_k as $n \to \infty$, and $\lim_{k\to\infty} \tau_k = \theta$ (see Section 12.3).

For example, for the fixed width-2 n-ominoes (shown in Figure 12.5.6 for small n), we have

$$T_2(z) = \frac{z}{1 - 2z - z^2} = z + 2z^2 + 5z^3 + 12z^4 + \ldots,$$

and $t(n+2,2) = 2t(n+1,2) + t(n,2)$ for $n \geq 1$.

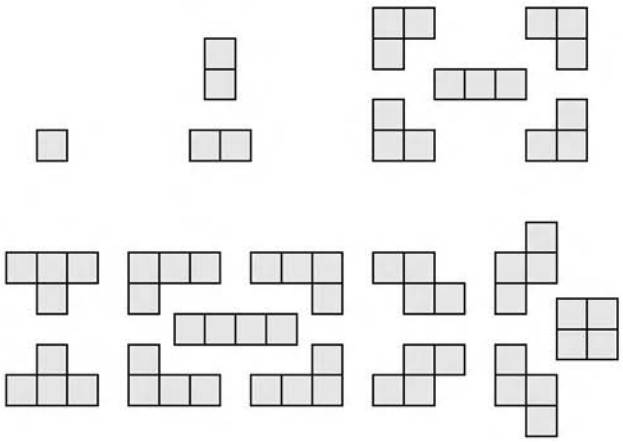


FIGURE 12.5.6
Width-2 n-ominoes for $n = 1, 2, 3, 4$.

DIRECTED POLYOMINOES

A portion of the family tree for directed polyominoes, constructed similarly to the one in Figure 12.4.1, is shown in Figure 12.5.7. As in Section 4, codewords can be defined for directed polyominoes, and converted into binary words. Let $\mathcal{V}$ be the language formed by all of these.

PROBLEM 12.5.8

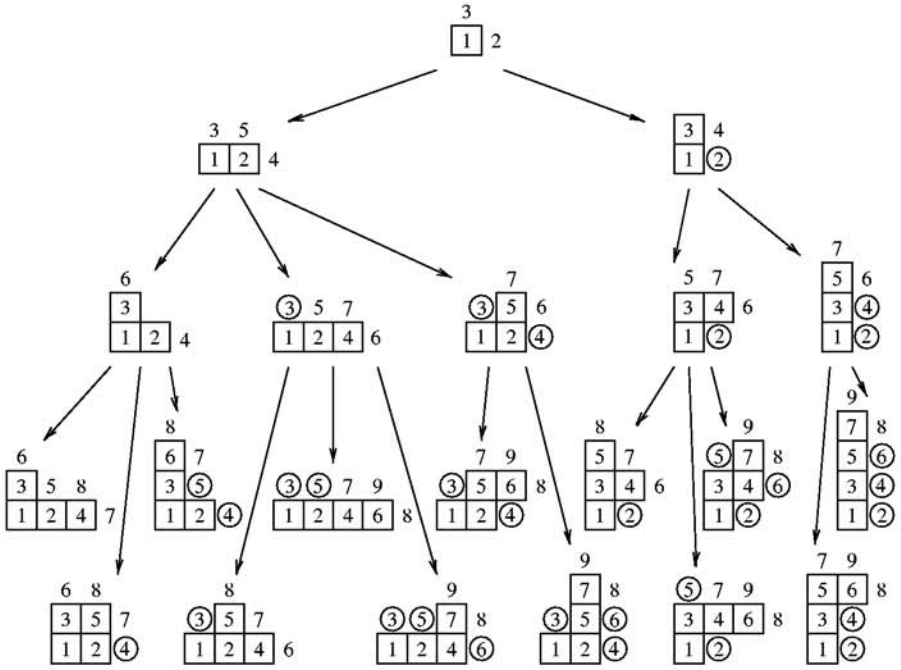
Characterize the words in $\mathcal{V}$. In particular, is $\mathcal{V}$ an unambiguous context-free language?

THEOREM 12.5.9 [Mél94]

If $d(n)$ is the number of directed n-ominoes in standard position, and $D(z) = \sum d(n)z^n$, then

$$D(z) = \frac{1}{2}\left(\sqrt{\frac{1+z}{1-3z}} - 1\right).$$

FIGURE 12.5.7
A family tree for fixed directed polyominoes.



COROLLARY 12.5.10

$$d(n) = \frac{1}{2}\sum_{k=0}^{n}\binom{1/2}{k}\binom{-1/2}{n-k}(-3)^{n-k}$$

and $d(n)$ satisfies the recurrence relation

$$d(n) = 3^{n-1} - \sum_{k=1}^{n-1} d(k)d(n-k).$$

D. Kaiser [Kai95] used the corollary to generate Table 12.5.1.

12.6 TILING WITH POLYOMINOES

We consider the special case of the tiling problem (see Chapter 3) in which the space we wish to tile is a set S of cells in the plane and the tiles are polyominoes. Usually S will be a rectangular set.

GLOSSARY

π-type: If S is a finite set of cells, $\mathcal{C}$ a collection of subsets of S, $\pi = (S_1, \ldots, S_k)$

TABLE 12.5.1 The first 30 values of $d(n)$, the number of directed n-ominoes in standard position.

n	$d(n)$	n	$d(n)$	n	$d(n)$
1	1	11	17303	21	741365049
2	2	12	49721	22	2173243128
3	5	13	143365	23	6377181825
4	13	14	414584	24	17830782252
5	35	15	1201917	25	55062568341
6	96	16	3492117	26	161995031226
7	267	17	10165779	27	476941691177
8	750	18	29643870	28	1405155255055
9	2123	19	86574831	29	4142457992363
10	6046	20	253188111	30	12219350698880

a partition (or cover) of S, and $T \subset S$, the π-type of T is defined as

$$\tau(\pi, T) = (|S_1 \cap T|, \ldots, |S_k \cap T|).$$

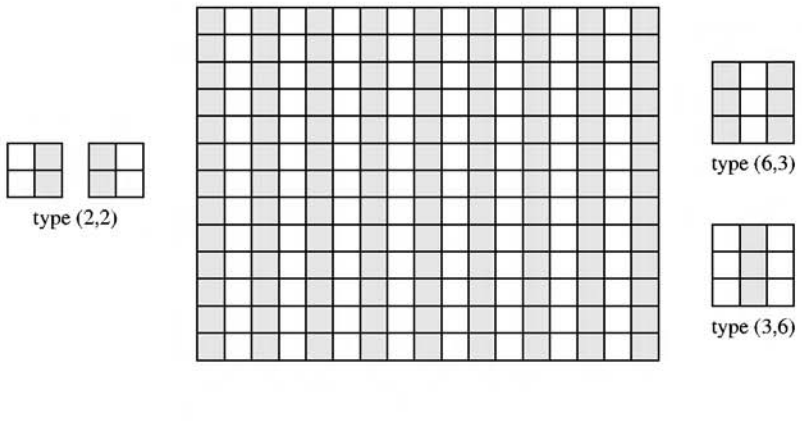
Basis: If every rectangle in a set R can be tiled with translates of rectangles belonging to a finite subset $B \subset R$, and if B is minimal with this property, B is called a basis of R.

THEOREM 12.6.1 [Kla70]

Suppose S is a finite set and $\mathcal{C}$ a collection of subsets of S. Then $\mathcal{C}$ tiles S if and only if, for every partition (or cover) π of S, $\tau(\pi, S)$ is a nonnegative integer combination of the types $\tau(\pi, T)$ where T ranges over $\mathcal{C}$.

For example, one can use this to show that a 13×17 rectangular array of squares cannot be tiled with 2×2 and 3×3 squares: Let π be the partition of the 13 array S into "black" and "white" cells shown in Figure 12.6.1, and $\mathcal{C}$ the set of all 2×2 and 3×3 squares in S.

FIGURE 12.6.1
A coloring of the 13×17 *rectangle.*



Then each 2×2 square in $\mathcal{C}$ has type $(2, 2)$, while the 3×3 squares have types $(6, 3)$ and $(3, 6)$. If a tiling were possible, with x 2×2 squares, and with y_1 and y_2 3×3 squares of types $(6, 3)$ and $(3, 6)$ (respectively), then we would have

$$(9 \cdot 13, 8 \cdot 13) = x(2, 2) + y_1(6, 3) + y_2(3, 6),$$

which gives $13 = 3(y_1 - y_2)$, a contradiction.

THEOREM 12.6.2

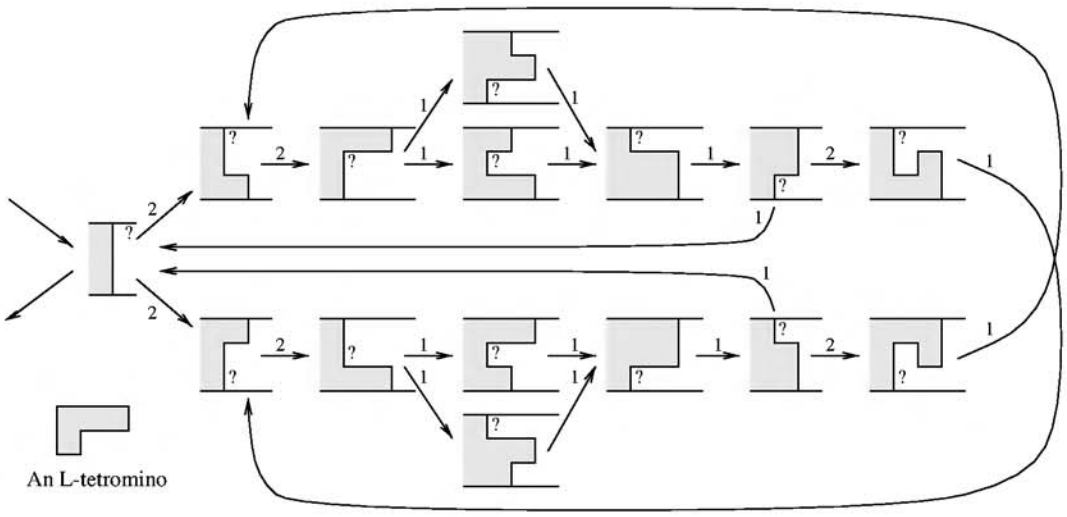
Let $\mathcal{C}$ be a finite union of translation classes of polyominoes, and let w be a fixed positive integer. Then one can construct a finite automaton that generates all $\mathcal{C}$-tilings of $w \times n$ rectangles for all possible values of n.

COROLLARY 12.6.3

If w is fixed and $\mathcal{C}$ is given, then it is possible to decide whether there exists some n for which $\mathcal{C}$ tiles a $w \times n$ rectangle.

For example, if we want to tile a $3 \times n$ rectangle with copies of the L-tetromino shown in Figure 12.6.2 in all eight possible orientations, the automaton of Figure 12.6.2 shows that it is necessary and sufficient for n to be a multiple of 8.

FIGURE 12.6.2
An automaton for tiling a $3 \times n$ rectangle with L-tetrominoes.



THEOREM 12.6.4 [KG69, dBK75]

Let R be an infinite set of oriented rectangles with integer dimensions. Then R has a finite basis.

(This theorem, which was originally conjectured by F. Göbel, extends to higher dimensions as well [dBK75].)

For example, let R be the set of all rectangles that can be tiled with the L-tetromino of Figure 12.6.2, and let $B = \{2 \times 4, 4 \times 2, 3 \times 8, 8 \times 3\} \subset R$. Then the following three facts are related:

(a) R is the set of all $a \times b$ rectangles with $a, b > 1$ and $8|ab$;
(b) B is a basis of R;
(c) Each member of B is tilable with the L-tetromino.

PROBLEM 12.6.5

The smallest rectangle that can be tiled with the Y-pentomino (shown in Figure 12.6.3) *is* 5×10. *Find a basis B for the set R of all rectangles that can be tiled with Y-pentominoes.*

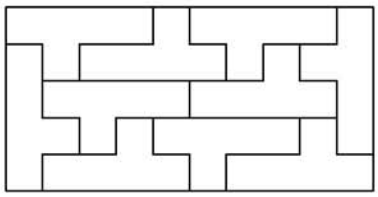


FIGURE 12.6.3
A 5×10 *rectangle tiled with Y-pentominoes.*

12.7 SOURCES AND RELATED MATERIAL

FURTHER READING

An excellent recent survey of the subject, complete with an abundance of references, is [Gol94]. Another notable book on the subject is [Mar91]. Finally, there are a great many articles, puzzles, and problems concerning polyominoes to be found in the magazine *Recreational Mathematics*.

RELATED CHAPTERS

Chapter 3: Tilings

REFERENCES

[Ben74] E.A. Bender. Convex n-ominoes. *Discrete Math*, 8:219–226, 1974.

[dBK75] N.G. de Bruijn and D.A. Klarner. A finite basis theorem for packing boxes with bricks. In *Papers Dedicated to C.J. Bouwkamp*, Philips Research Reports 30:337–343, 1975.

[Gol] S. Golomb. Personal communication.

[Gol94] S. Golomb. *Polyominoes*, 2nd edition. Princeton Univ. Press, Princeton, 1994.

[Kai95] D. Kaiser. Personal communication, 1995.

[Kla67] D.A. Klarner. Cell growth problems. *Canad. J. Math.*, 19:851–863, 1967.

[Kla70] D.A. Klarner. A packing theory. *J. Combin. Theory*, 8:272–278, 1970.

[KG69] D.A. Klarner and F. Göbel. Packing boxes with congruent figures. *Indag. Math.* (Amsterdam), 31:465–472, 1969.

[KR73] D.A. Klarner and R.L. Rivest. A procedure for improving the upper bound for the number of n-ominoes. *Canad. J. Math*, 25:585–602, 1973.

[KR74] D.A. Klarner and R.L. Rivest. Asymptotic bounds for the number of convex n-ominoes. *Discrete Math*, 8:31–40, 1974.

[KS] D.A. Klarner and W. Satterfield. The number of width-k n-ominoes. To appear.

[Knu72] D.E. Knuth. Personal communication, 1972.

[Mar91] G.E. Martin. *Polyominoes. A Guide to Puzzles and Problems in Tiling.* Math. Assoc. Amer., Washington, D.C., 1991.

[Mél94] M. Bousquet-Mélou. Polyominoes and polygons. *Contemp. Math.*, 178:55–70, 1994.

[Rea62] R.C. Read. Contributions to the cell growth problem. *Canad. J. Math.*, 14:1–20, 1962.

[Red81] D.H. Redelmeier. Counting polyominoes: yet another attack. *Discrete Math*, 36:191–203, 1981.

POLYTOPES AND POLYHEDRA

13 BASIC PROPERTIES OF CONVEX POLYTOPES

Martin Henk, Jürgen Richter-Gebert, and Günter M. Ziegler

INTRODUCTION

Convex polytopes are fundamental geometric objects that have been investigated since antiquity. The beauty of their theory is nowadays complemented by their importance for many other mathematical subjects, ranging from integration theory, algebraic topology, and algebraic geometry (toric varieties) to linear and combinatorial optimization.

In this chapter we try to give a short introduction, provide a sketch of "what polytopes look like" and "how they behave," with many explicit examples, and briefly state some main results (where further details are in the subsequent chapters of this Handbook). We concentrate on two main topics:

- Combinatorial properties: faces (vertices, edges, ..., facets) of polytopes and their relations, with special treatments of the classes of "low-dimensional polytopes" and "polytopes with few vertices";
- Geometric properties: volume and surface area, mixed volumes, and quermassintegrals, including explicit formulas for the cases of the regular simplices, cubes, and cross-polytopes.

We refer to Grünbaum [Grü67] for a comprehensive view of polytope theory, and to Ziegler [Zie95] and Schneider [Sch93] for recent treatments of the combinatorial resp. convex geometric aspects of polytope theory.

13.1 COMBINATORIAL STRUCTURE

GLOSSARY

$\mathcal{V}$-polytope: The convex hull of a finite set $X = \{x^1, \ldots, x^n\}$ of points in $\mathbb{R}^d$:

$$P = \mathrm{conv}(X) := \Big\{ \sum_{i=1}^{n} \lambda_i x^i \;\Big|\; \lambda_i \geq 0,\ \sum_{i=1}^{n} \lambda_i = 1 \Big\}.$$

$\mathcal{H}$-polytope: A bounded solution set of a finite system of linear inequalities:

$$P = P(A, b) := \big\{ x \in \mathbb{R}^d \mid a_i^T x \leq b_i \text{ for } 1 \leq i \leq m \big\},$$

where $A \in \mathbb{R}^{m \times d}$ is a real matrix with rows a_i^T, and $b \in \mathbb{R}^m$ is a real vector with entries b_i. Here boundedness means that there is a constant N such that $||x|| \leq N$ holds for all $x \in P$.

Polytope: A subset $P \subseteq \mathbb{R}^d$ that can be presented as a $\mathcal{V}$-polytope or (equivalently, by the main theorem below!) as an $\mathcal{H}$-polytope.

Dimension: The dimension of an arbitrary subset $S \subseteq \mathbb{R}^d$ is defined as the dimension of its affine hull: $\dim(S) := \dim(\mathrm{aff}(S))$.

(Recall that $\mathrm{aff}(S)$, the affine hull of a set S, is $\left\{ \sum_{j=1}^{p} \lambda_j x^j \mid x^1, ..., x^p \in S, \sum_{j=1}^{p} \lambda_j = 1 \right\}$, the smallest affine subspace of $\mathbb{R}^d$ containing S.)

d-polytope: A d-dimensional polytope. In what follows, a subscript in the name of a polytope usually denotes its dimension.

Interior and ***relative interior:*** The interior $\mathrm{int}(P)$ is the set of all points $x \in P$ such that for some $\epsilon > 0$, the ϵ-ball $B_\epsilon(x)$ around x is contained in P.

Similarly, the relative interior $\mathrm{relint}(P)$ is the set of all points $x \in P$ such that for some $\epsilon > 0$, the intersection $B_\epsilon(x) \cap \mathrm{aff}(P)$ is contained in P.

Affine equivalence: For polytopes $P \subseteq \mathbb{R}^d$ and $Q \subseteq \mathbb{R}^e$, an affine map $\pi: \mathbb{R}^d \longrightarrow \mathbb{R}^e$, $x \mapsto Ax + b$ mapping P bijectively to Q. π need not be injective or surjective. However, it has to restrict to a bijective map $\mathrm{aff}(P) \longrightarrow \mathrm{aff}(Q)$. In particular, if P and Q are affinely equivalent, then they have the same dimension.

THEOREM 13.1.1 *Main Theorem of Polytope Theory*

The definitions of $\mathcal{V}$-polytopes and of $\mathcal{H}$-polytopes are equivalent. That is, every $\mathcal{V}$-polytope has a description by a finite system of inequalities, and every $\mathcal{H}$-polytope can be obtained as the convex hull of a finite set of points (its vertices*).*

Geometrically, a $\mathcal{V}$-polytope is the projection of an $(n-1)$-dimensional simplex, while an $\mathcal{H}$-polytope is the bounded intersection of m closed halfspaces [Zie95, Lecture 1]. To see the main theorem at work, consider the following two statements: the first one is easy to see for $\mathcal{V}$-polytopes, but not for $\mathcal{H}$-polytopes, and for the second statement we have the opposite effect.

1. *Projections:* Every image of a polytope P under an affine map $\pi: x \mapsto Ax + b$ is a polytope.
2. *Intersections:* Any intersection of a polytope with an affine subspace is a polytope.

However, the computational step from one of the main theorem's descriptions of polytopes to the other—a "convex hull computation"—is far from trivial. Essentially, there are three types of algorithms available: inductive algorithms (inserting vertices, using a so-called beneath-beyond technique), projection resp. intersection algorithms (known as Fourier-Motzkin elimination resp. double description algorithms), and reverse search methods (as introduced by Avis and Fukuda). For explicit computations one can use public domain codes such as the PORTA code [Chr96] that we use here, which implements an algorithm of the second type; see also Chapters 19 and 52.

In the following definitions of d-simplices, d-cubes, and d-cross-polytopes we give both a $\mathcal{V}$- and an $\mathcal{H}$-presentation in each case. From this one can see that the $\mathcal{H}$-presentation can have exponential "size" in terms of the size of the $\mathcal{V}$-presentation (e.g., for the d-cross-polytopes), and vice versa (for the d-cubes).

Definition: A (regular) d-dimensional ***simplex*** in $\mathbb{R}^d$ is given by

$$T_d := \mathrm{conv}\left\{ e^1, e^2, \ldots, e^d, \frac{1 - \sqrt{d+1}}{d}(e^1 + \ldots + e^d) \right\}$$

$$= \quad \Big\{x \in \mathbb{R}^d \mid \sum_{i=1}^{d} x_i \le 1, \ -(1+\sqrt{d+1}+d)x_k + \sum_{i=1}^{d} x_i \le 1 \text{ for } 1 \le k \le d\Big\},$$

where $e^1, \ldots, e^d$ denotes the coordinate unit vectors in $\mathbb{R}^d$.

The simplices T_d are ***regular polytopes*** (with a symmetry group that is flag-transitive—see Chapter 16): the parameters have been chosen so that all edges of T_d have length $\sqrt{2}$. Furthermore, the origin $0 \in \mathbb{R}^d$ is in the interior of T_d: this is clear from the $\mathcal{H}$-presentation.

However, for the combinatorial theory one considers polytopes that differ only by a change of coordinates (an affine transformation) to be equivalent. Thus, we would refer to any d-polytope that can be presented as the convex hull of $d+1$ points as a *d-simplex*, since any two such polytopes are equivalent with respect to an affine map. Other standard choices include

$$\begin{aligned} \Delta_d &:= \operatorname{conv}\{0, e^1, e^2, \ldots, e^d\} \\ &= \Big\{x \in \mathbb{R}^d \mid \sum_{i=1}^{d} x_i \le 1, \quad x_k \ge 0 \text{ for } 1 \le k \le d\Big\} \end{aligned}$$

and the $(d-1)$-dimensional simplex in $\mathbb{R}^d$ given by

$$\begin{aligned} \Delta'_{d-1} &:= \operatorname{conv}\{e^1, e^2, \ldots, e^d\} \\ &= \Big\{x \in \mathbb{R}^d \mid \sum_{i=1}^{d} x_i = 1, \quad x_k \ge 0 \text{ for } 1 \le k \le d\Big\}. \end{aligned}$$

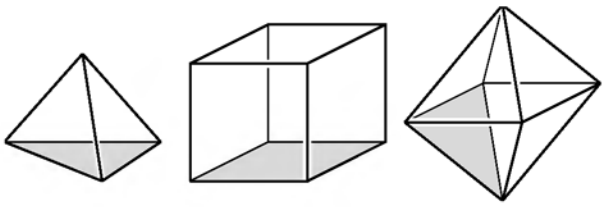


FIGURE 13.1.1
A 3-simplex, a 3-cube, and a 3-dimensional cross-polytope (octahedron).

Definition: A *d-**cube*** (a.k.a. the d-dimensional ***hypercube***) is

$$\begin{aligned} C_d &:= \operatorname{conv}\big\{\alpha_1 e^1 + \alpha_2 e^2 + \ldots + \alpha_d e^d \mid \alpha_1, \ldots, \alpha_d \in \{+1, -1\}\big\} \\ &= \Big\{x \in \mathbb{R}^d \mid -1 \le x_k \le 1 \text{ for } 1 \le k \le d\Big\}, \end{aligned}$$

and a d-dimensional ***cross-polytope*** in $\mathbb{R}^d$ (known as the ***octahedron*** for $d = 3$) is given by

$$C_d^{\Delta} := \operatorname{conv}\{\pm e^1, \pm e^2, \ldots, \pm e^d\} = \Big\{x \in \mathbb{R}^d \mid \sum_{i=1}^{d} |x_i| \le 1\Big\}.$$

Again, there are other natural choices, among them

$$\begin{aligned} [0,1]^d &= \operatorname{conv}\Big\{\sum_{i \in S} e^i \mid S \subseteq \{1, 2, \ldots, d\}\Big\} \\ &= \Big\{x \in \mathbb{R}^d \mid 0 \le x_k \le 1 \text{ for } 1 \le k \le d\Big\}, \end{aligned}$$

the d-dimensional ***unit cube***.

As another example to illustrate concepts and results we will occasionally use the unnamed polytope with six vertices shown in Figure 13.1.2.

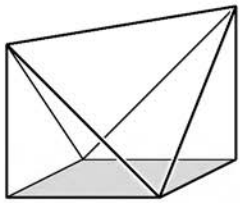


FIGURE 13.1.2
Our unnamed "typical" 3-polytope. It has 6 vertices, 11 edges and 7 facets.

This polytope without a name can be presented as a $\mathcal{V}$-polytope by listing its six vertices. The following coordinates make it into a subpolytope of the 3-cube C_3: the vertex set consists of all but two vertices of C_3. Our list below (on the left) is in the format used as input for the PORTA program, e.g., in a file named `unnamedpoly.poi`. From these data the PORTA program produces a description (on the right) of the polytope as an $\mathcal{H}$-polytope, stored in the file `unnamedpoly.poi.ieq`

```
DIM = 3
CONV_SECTION
( 1)   1  1  1
( 2)  -1 -1  1
( 3)   1  1 -1
( 4)   1 -1 -1
( 5)  -1  1 -1
( 6)  -1 -1 -1
END
```

```
INEQUALITIES_SECTION
(  1)    +x2     <= 1
(  2) +x1        <= 1
(  3) -x1        <= 1
(  4)    -x2     <= 1
(  5)       -x3  <= 1
(  6) -x1+x2+x3  <= 1
(  7) +x1-x2+x3  <= 1
```

Unbounded polyhedra can, via projective transformations, be treated as polytopes with a distinguished facet (see [Zie95, p. 75]). In this respect, we do not lose anything on the combinatorial level if we restrict the following discussion to the setting of full-dimensional convex polytopes: d-polytopes embedded in $\mathbb{R}^d$.

13.1.1 FACES

GLOSSARY

Support function: Given a polytope $P \subseteq \mathbb{R}^d$, the function

$$h(P,\cdot)\colon \mathbb{R}^d \to \mathbb{R}, \qquad h(P,x) := \sup\{\langle x,y\rangle \mid y \in P\},$$

where $\langle x,y\rangle$ denotes the inner product on $\mathbb{R}^d$. (Since P is compact one may replace sup by max.)

For $v \in \mathbb{R}^d \setminus \{0\}$ the hyperplane

$$H(P,v) := \{x \in \mathbb{R}^d \mid \langle x,v\rangle = h(P,v)\}$$

is the ***supporting hyperplane*** of P with ***outer normal vector*** v. Note that $H(P,\mu v) = H(P,v)$ for $\mu \in \mathbb{R}$, $\mu > 0$. For a vector u of the $(d-1)$-dimensional ***unit sphere*** S^{d-1}, $h(P,u)$ is the signed distance of the supporting plane $H(P,u)$ from the origin. (For $v = 0$ we set $H(P,0) := \mathbb{R}^d$, which is not a hyperplane.)

The intersection of P with a supporting hyperplane $H(P,v)$ is called a (nontrivial) ***face***, or more precisely a ***k-face*** if the dimension of $\mathrm{aff}(P \cap H(P,v))$ is k. Each face is itself a polytope.

The set of all k-faces is denoted by $\mathcal{F}_k(P)$ and its cardinality by $f_k(P)$.

f-vector: The vector of face numbers $\boldsymbol{f}(P) = (f_0(P), f_1(P), \ldots, f_{d-1}(P))$ associated with a d-polytope.

The empty set $\emptyset$ and the polytope P itself are considered ***trivial faces*** of P, of dimensions -1 and $\dim(P)$, respectively. All faces other than P are ***proper faces***.

The faces of dimension 0 and 1 are called ***vertices*** and ***edges***, respectively. The $(\dim(P)-1)$-faces of P are called ***facets***.

Facet-vertex incidence matrix: The matrix $M \in \{0,1\}^{f_{d-1}(P) \times f_0(P)}$ that has an entry $M(F,v) = 1$ if the facet F contains the vertex v, and $M(F,v) = 0$ otherwise.

Graded poset: A partially ordered set $(P, \leq)$ with a unique minimal element $\hat{0}$, a unique maximal element $\hat{1}$, and a ***rank function*** $r{:}P \longrightarrow \mathbb{N}_0$ that satisfies
(1) $r(\hat{0}) = 0$, and $p < p'$ implies $r(p) < r(p')$, and
(2) $p < p'$ and $r(p') - r(p) > 1$ implies that there is a $p'' \in P$ with $p < p'' < p'$.

Lattice L: A partially ordered set $(P, \leq)$ in which every pair of elements $p, p' \in P$ has a unique maximal lower bound, called the ***meet*** $p \wedge p'$, and a unique minimal upper bound, called the ***join*** $p \vee p'$.

Atom, coatom: If L is a graded lattice, the minimal elements of $L \setminus \{\hat{0}\}$ (i.e., the elements of rank 1) are the atoms of L. Similarly, the maximal elements of $L \setminus \{\hat{1}\}$ (i.e., the elements of rank $r(\hat{1})-1$) are the coatoms of L. A graded lattice is ***atomic*** if every element is a join of a set of atoms, and it is ***coatomic*** if every element is a meet of a set of coatoms.

Face lattice L(P): The set of all faces of P, partially ordered by inclusion.

Combinatorially isomorphic: Polytopes whose face lattices are isomorphic as abstract (unlabeled) partially ordered sets/lattices.

Equivalently, P and P' are combinatorially equivalent if their facet-vertex incidence matrices differ only by column and row permutations.

Combinatorial type: An equivalence class of polytopes under combinatorial equivalence.

THEOREM 13.1.2 *Face Lattices of Polytopes*

The face lattices of convex polytopes are finite, graded, atomic, and coatomic lattices. The meet operation $G \wedge H$ is given by intersection, while the join $G \vee H$ is the intersection of all facets that contain both G and H. The rank function on $L(P)$ is given by $r(G) = \dim(G) + 1$.

The minimal nonempty faces of a polytope are its vertices: they correspond to atoms of the lattice $L(P)$. Every face is the join of its vertices, hence $L(P)$ is atomic. Similarly, the maximal proper faces of a polytope are its facets: they

correspond to the coatoms of $L(P)$. Every face is the intersection of the facets it is contained in, hence face lattices of polytopes are coatomic.

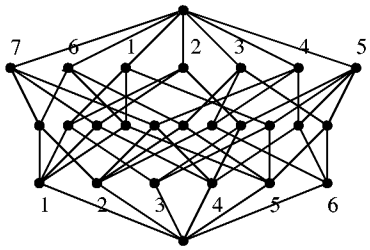


FIGURE 13.1.3
The face lattice of our unnamed 3-polytope. The 7 coatoms and the 6 atoms have been labeled to correspond to the labels in the facet-vertex incidence matrix. Thus, the downwards-path from the coatom "4" to the atom "2" represents the fact that the facet numbered (4) *contains the vertex* (2).

The face lattice is a complete encoding of the combinatorial structure of a polytope. However, in general the encoding by a facet-vertex incidence matrix is more efficient. The following matrix—also provided by PORTA—represents our unnamed 3-polytope:

$$M = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

How do we decide whether a set of vertices $\{v^1, \ldots, v^k\}$ is (the vertex set of) a face of P? This is the case if and only if no other vertex v^0 is contained in all the facets that contain $\{v^1, \ldots, v^k\}$. This criterion makes it possible, for example, to derive the edges of a polytope P from a facet-vertex matrix.

For low-dimensional polytopes, the criterion can be simplified: if $d \leq 4$, then two vertices are connected by an edge if and only if there are at least $d-1$ different facets that contain them both. However, the same is not true any longer for 5-dimensional polytopes, where vertices may be nonadjacent despite being contained in many common facets. (The best way to see this is by using polarity; see below.)

13.1.2 POLARITY

GLOSSARY

Polarity: If $P \subseteq \mathbb{R}^d$ is a d-polytope with the origin in its interior, then the ***polar*** of P is the d-polytope

$$P^{\Delta} := \{y \in \mathbb{R}^d \mid \langle y, x \rangle \leq 1 \text{ for all } x \in P\}.$$

Stellar subdivision: The stellar subdivision of a polytope P in a face F is the

polytope $\text{conv}(P \cup x^F)$, where x^F is a point of the form $y^F - \epsilon(y^P - y^F)$, where y^P is in the interior of P, y^F is in the relative interior of F, and ϵ is small enough.

Vertex figure P/v: If v is a vertex of P, then $P/v := P \cap H$ is the polytope obtained by intersecting P with a hyperplane H that has v on one side and all the other vertices of P on the other side.

Cutting off a vertex: The polytope $P \cap H^-$ obtained by intersecting P with a closed halfspace H^- that does not contain the vertex v, but contains all other vertices of P in its interior. (In this situation, $P \cap H^+$ is a pyramid over the vertex figure P/v.)

Quotient of P: A polytope obtained from P by taking vertex figures (possibly) several times.

Simplicial polytope: A polytope all of whose facets (equivalently, proper faces) are simplices.

Simple polytope: A polytope all of whose vertex figures (equivalently, proper quotients) are simplices.

Polarity is a fundamental construction in the theory of polytopes. One always has $P^{\Delta\Delta} = P$, under the assumption that P has the origin in its interior. This condition can always be obtained after a change of coordinates. In particular, we speak of (combinatorial) polarity between d-polytopes Q and R that are combinatorially isomorphic to P and P^Δ, respectively.

Any $\mathcal{V}$-presentation of P yields an $\mathcal{H}$-presentation of P^Δ, and conversely, via

$$P = \text{conv}\{v^1, \ldots, v^n\} \quad \Longleftrightarrow \quad P^\Delta = \{x \in \mathbb{R}^d \mid \langle v^i, x\rangle \leq 1 \text{ for } 1 \leq i \leq n\}.$$

There are basic relations between polytopes and polytopal constructions under polarity. For example, the fact that the d-cross-polytopes C_d^Δ are the polars of the d-cubes C_d is built into our notation. More generally, the polars of simple polytopes are simplicial, and conversely. This can be deduced from the fact that the facets F of a polytope P correspond to the vertex figures P^Δ/v of its polar P^Δ. In fact, F and P^Δ/v are combinatorially polar in this situation. More generally, one has a correspondence between faces and quotients under polarity.

At a combinatorial level, all this can be derived from the fact that the face lattices $L(P)$ and $L(P^\Delta)$ are anti-isomorphic: $L(P^\Delta)$ may be obtained from $L(P)$ by reversing the order relations. Thus, lower intervals in $L(P)$, corresponding to faces of P, translate under polarity into upper intervals of $L(P^\Delta)$, corresponding to quotients of P^Δ.

13.1.3 BASIC CONSTRUCTIONS

GLOSSARY

For the following constructions, let
$P \subseteq \mathbb{R}^d$ be a d-dimensional polytope with n vertices and m facets, and
$P' \subseteq \mathbb{R}^{d'}$ a d'-dimensional polytope with n' vertices and m' facets.

Scalar multiple: For $\lambda \in \mathbb{R}$, the scalar multiple λP is defined by $\lambda P := \{\lambda x \mid x \in P\}$. P and λP are combinatorially (in fact, affinely) isomorphic for all $\lambda \neq 0$. In particular, $(-1)P = -P = \{-p \mid p \in P\}$, and $(+1)P = P$.

Minkowski sum: $P + P' := \{p + p' \mid p \in P,\, p' \in P'\}$.

It is also useful to define the difference as $P - P' = P + (-P')$. The polytopes $P + \lambda P'$ are combinatorially isomorphic for all $\lambda > 0$, and similarly for $\lambda < 0$.

If $P' = \{p'\}$ is one single point, then $P - \{p'\}$ is the image of P under the translation that takes p' to the origin.

Product: The $(d+d')$-dimensional polytope $P \times P' := \{(p, p') \in \mathbb{R}^{d+d'} \mid p \in P,\, p' \in P'\}$. $P \times P'$ has $n \cdot n'$ vertices and $m + m'$ facets.

Join: The convex hull $P * P'$ of $P \cup P'$, after embedding P and P' in a space where their affine hulls are skew. For example,
$P * P' := \mathrm{conv}(\{(p, 0, 0) \in \mathbb{R}^{d+d'+1} \mid p \in P\} \cup \{(0, p', 1) \in \mathbb{R}^{d+d'+1} \mid p' \in P'\})$.
$P*P'$ has dimension $d+d'+1$ and $n+n'$ vertices. Its k-faces are the joins of i-faces of P and $(k-i-1)$-faces of P', hence $f_k(P * P') = \sum_{i=-1}^{k} f_i(P) f_{k-i-1}(P')$.

Free sum: The free sum is the $(d+d')$-dimensional polytope
$P \oplus P' := \mathrm{conv}(\{(p, 0) \in \mathbb{R}^{d+d'} \mid p \in P\} \cup \{(0, p') \in \mathbb{R}^{d+d'} \mid p' \in P'\})$.

Thus the free sum $P \oplus P'$ is a projection of the join $P * P'$. If both P and P' have the origin in their interiors—this is the "usual" situation for creating free sums, then $P \oplus P'$ has $n + n'$ vertices and $m \cdot m'$ facets.

Pyramid: The join $\mathrm{pyr}(P) := P * \{0\}$ of P with a point (a 0-dimensional polytope $P' = \{0\} \subseteq \mathbb{R}^0$). The pyramid $\mathrm{pyr}(P)$ has $n + 1$ vertices and $m + 1$ facets.

Prism: The product $\mathrm{prism}(P) := P \times I$, where I denotes the real interval $I = [-1, +1] \subseteq \mathbb{R}$.

Bipyramid: If P has the origin in its interior, then the bipyramid over P is the $(d+1)$-dimensional polytope constructed as the free sum $\mathrm{bipyr}(P) := P \oplus I$.

Lawrence extension: If $p \in \mathbb{R}^d$ is a point outside P, then the free sum $(P - \{p\}) \oplus [1, 2]$ is a *Lawrence extension of P at p.* (For $p \in P$ this is just a pyramid.)

Of course, the many constructions listed in the glossary above are not independent of each other. For instance, some of these constructions are related by polarity: for polytopes P and P' with the origin in their interiors, the product and the free sum constructions are related by polarity,

$$P \times P' \;=\; (P^{\Delta} \oplus P'^{\Delta})^{\Delta},$$

and this specializes to polarity relations among the pyramid, bipyramid, and prism constructions,

$$\mathrm{pyr}(P) \;=\; (\mathrm{pyr}(P^{\Delta}))^{\Delta} \qquad \text{and} \qquad \mathrm{prism}(P) \;=\; (\mathrm{bipyr}(P^{\Delta}))^{\Delta}.$$

Similarly, "cutting off a vertex" is polar to "stellar subdivision in a facet."

It is interesting to study—and this has not been done systematically—how the basic polytope operations generate complicated convex polytopes from simpler ones. For example, starting from a one-dimensional polytope $I = C_1 = [-1, +1] \subset \mathbb{R}$, the direct product construction generates the cubes C_d, while free sums generate the cross-polytopes C_d^{Δ}.

Even more complicated centrally symmetric polytopes, the *Hanner polytopes*, are obtained from copies of the interval I by using products and free sums. They are

interesting since they achieve with equality the conjectured bound that all centrally symmetric d-polytopes have at least 3^d nonempty faces (Kalai [Kal89]).

Every polytope can be viewed as a region of a hyperplane arrangement: for this, take as $\mathcal{A}_P$ the set of all hyperplanes of the form aff(F), where F is a facet of P. For additional points, such as the points outside the polytope used for Lawrence extensions, or those used for stellar subdivisions, it is often important only in which region, or in which lower-dimensional region, of the arrangement $\mathcal{A}_P$ they lie.

The Lawrence extension, by the way, may seem like quite a harmless little construction. However, it has the amazing property that it can encode the structure of a point *outside* a d-polytope into the boundary structure of a $(d+1)$-polytope. This accounts for a large part of the "special" 4- and 5-polytopes in the literature, such as the 4-polytopes for which a facet, or even a 2-face, cannot be prescribed in shape [Ric96].

13.1.4 MORE EXAMPLES

There are many interesting classes of polytopes arising from diverse areas of mathematics (as well as physics, optimization, crystallography, etc.). Some of these are discussed below. You will find many more classes of examples discussed in other chapters of this Handbook. For example, regular and semiregular polytopes are discussed in Chapter 16, while polytopes that arise as Voronoi cells of lattices appear in Chapters 3, 7, and 51.

GLOSSARY

Graph of a polytope: The graph $G(P) = (V(P), E(P))$ with vertex set $V(P) = \mathcal{F}_0(P)$ and edge set $E(P) = \{\{v^1, v^2\} \subseteq \binom{V}{2} \mid \text{conv}\{v^1, v^2\} \in \mathcal{F}_1(P)\}$.

Zonotope: Any polytope Z that can be represented as the image of an n-dimensional cube C_n under an affine map; equivalently, any polytope that can be written as a Minkowski sum of n line segments (1-dimensional polytopes). The smallest n such that Z is an image of C_n is the ***number of zones*** of Z.

Moment curve: The curve γ in $\mathbb{R}^d$ defined by $\gamma : \mathbb{R} \longrightarrow \mathbb{R}^d, t \longmapsto (t, t^2, \ldots, t^d)^T$.

Cyclic polytope: The convex hull of a finite set of points on a moment curve, or any polytope combinatorially equivalent to it.

k-neighborly polytope: A polytope such that each subset of at most k vertices forms the vertex set of a face. Thus every polytope is 1-neighborly, and a polytope is 2-neighborly if and only if its graph is complete.

Neighborly polytope: A d-dimensional polytope that is $\lfloor d/2 \rfloor$-neighborly.

(0,1)-polytope: A polytope all of whose vertex coordinates are 0 or 1, that is, whose vertex set is a subset of the vertex set $\{0,1\}^d$ of the unit cube.

ZONOTOPES

Zonotopes appear in quite different guises. They can equivalently be defined as the Minkowski sums of finite sets of line segments (1-dimensional polytopes), as the

affine projections of d-cubes, or as polytopes all of whose faces (equivalently, all 2-faces) exhibit central symmetry. Thus a 2-dimensional polytope is a zonotope if and only if it is centrally symmetric.

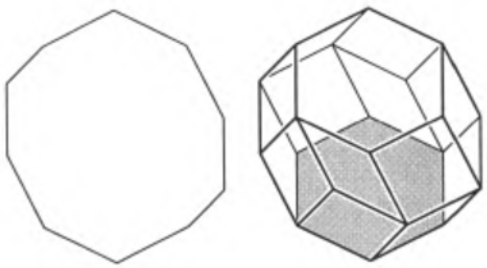


FIGURE 13.1.4
A 2-dimensional and a 3-dimensional zonotope, each with 5 zones. (The 2-dimensional one is a projection of the 3-dimensional one; note that every projection of a zonotope is a zonotope.)

Among the most prominent zonotopes are the permutohedra: The ***permutohedron*** Π_{d-1} is constructed by taking the convex hull of all d-vectors whose coordinates are $\{1, 2, \ldots, d\}$, in any order. The permutohedron Π_{d-1} is a $(d-1)$-dimensional polytope (contained in the hyperplane $\{x \in \mathbb{R}^d \mid \sum_{i=1}^{d} x_i = d(d+1)/2\}$) with $d!$ vertices and $2^d - 2$ facets.

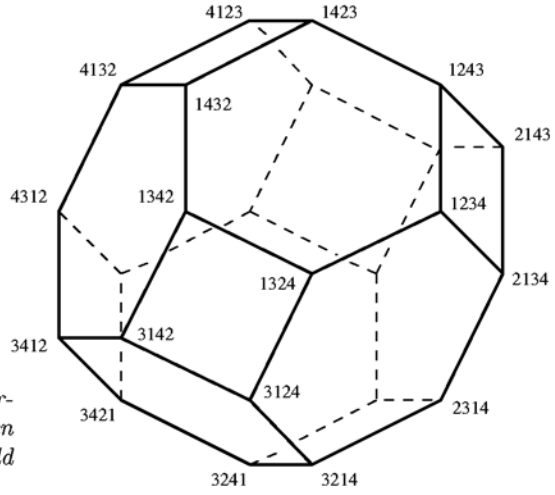


FIGURE 13.1.5
The 3-dimensional permutohedron Π_3. (The vertices are labeled by the permutations that, when applied to the coordinate vector in $\mathbb{R}^4$, yield $(1, 2, 3, 4)^T$.)

One unusual feature of permutohedra is that they are simple zonotopes: these are rare in general, and the (unsolved) problem of classifying them is equivalent to the problem of classifying all simplicial arrangements of hyperplanes (see Section 6.3.3).

Zonotopes are important because their theory is equivalent to the theories of vector configurations (realizable oriented matroids) and of hyperplane arrangements. In fact, the system of line segments that generates a zonotope can be considered as a vector configuration, and the hyperplanes that are orthogonal to the line segments provide the associated hyperplane arrangement. We refer to [BLS+93, Section 2.2] and [Zie95, Lecture 7].

Finally, we mention in passing a surprising bijective correspondence between the

tilings of a zonotope with smaller zonotopes and oriented matroid liftings (realizable or not) of the oriented matroid of a zonotope. This correspondence is known as the *Bohne-Dress theorem*; we refer to Richter-Gebert and Ziegler [RZ94].

CYCLIC POLYTOPES

Cyclic polytopes can be constructed by taking the convex hull of $n > d$ points on the moment curve in $\mathbb{R}^d$. The "standard construction" is to define a cyclic polytope $C_d(n)$ as the convex hull of n integer points on this curve, such as

$$C_d(n) \;:=\; \operatorname{conv}\{\gamma(1), \gamma(2), \ldots, \gamma(n)\}.$$

However, the combinatorial type of $C_d(n)$ is given by the—entirely combinatorial—***Gale evenness criterion***: If $C_d(n) = \operatorname{conv}\{\gamma(t_1), \ldots, \gamma(t_n)\}$, with $t_1 < \ldots < t_n$, then $\gamma(t_{i_1}), \ldots, \gamma(t_{i_d})$ determine a facet if and only if the number of indices in $\{i_1, ..., i_d\}$ lying between any two indices *not* in that set is even. Thus, the combinatorial type does not depend on the specific choice of points on the moment curve [Zie95, Example 0.6; Theorem 0.7].

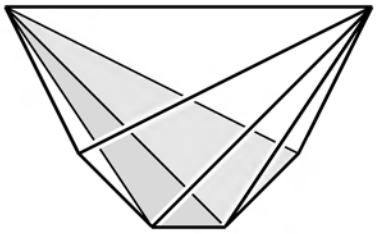


FIGURE 13.1.6
A 3-dimensional cyclic polytope $C_3(6)$ with 6 vertices. (In a projection of γ to the x_1x_2-plane, the curve γ and hence the vertices of $C_3(6)$ lie on the parabola $x_2 = x_1^2$.)

The first property of cyclic polytopes to notice is that they are simplicial. The second, more surprising, property is that they are neighborly. This implies that among all d-polytopes P with n vertices, the cyclic polytopes maximize the number $f_i(P)$ of i-dimensional faces for $i < \lfloor d/2 \rfloor$. The same fact holds for all i: this is part of McMullen's upper bound theorem (see below). In particular, cyclic polytopes have a very large number of facets,

$$f_{d-1}\big(C_d(n)\big) \;=\; \binom{n - \lceil \frac{d}{2} \rceil}{\lfloor \frac{d}{2} \rfloor} \;+\; \binom{n - 1 - \lceil \frac{d-1}{2} \rceil}{\lfloor \frac{d-1}{2} \rfloor}.$$

For example, we get that a cyclic 4-polytope $C_4(n)$ has $n(n-3)/2$ facets. Thus $C_4(8)$ has 8 vertices, any two of them adjacent, and 20 facets (this is more than the 16 facets of the 4-dimensional cross-polytope, which also has 8 vertices!).

NEIGHBORLY POLYTOPES

Here are a few observations about neighborly polytopes. For more information, see [BLS+93, Section 9.4] and the references quoted there.

The first observation is that if a polytope is k-neighborly for some $k > \lfloor d/2 \rfloor$, then it is a simplex. Thus, if one ignores the simplices, then $\lfloor d/2 \rfloor$-neighborly

polytopes form the extreme case, which motivates calling them simply "neighborly." However, only in even dimensions $d = 2m$ do the neighborly polytopes have very special structure. For example, one can show that even-dimensional neighborly polytopes are necessarily simplicial, but this is not true in general. For the latter, note that, for example, all 3-dimensional polytopes are neighborly by definition, and that if P is a neighborly polytope of dimension $d = 2m$, then $\mathrm{pyr}(P)$ is neighborly of dimension $2m+1$.

All simplicial neighborly d-polytopes with n vertices have the same number of facets (in fact, the same f-vector $(f_0, f_1, \ldots, f_{d-1})$) as $C_d(n)$. They constitute the class of polytopes with the maximal number of i-faces for all i: this is the statement of McMullen's upper bound theorem. We refer to Chapter 15 for a thorough discussion of f-vector theory.

For $n \leq d+3$, every neighborly polytope is combinatorially isomorphic to a cyclic polytope. (This covers, for instance, the polar of the product of two triangles, $(\Delta_2 \times \Delta_2)^\Delta$, which is easily seen to be a 4-dimensional neighborly polytope with 6 vertices; see Figure 13.1.9.) The first example of an even-dimensional neighborly polytope that is not cyclic appears for $d = 4$ and $n = 8$. It can easily be described in terms of its affine Gale diagram; see below.

Neighborly polytopes may at first glance seem to be very peculiar and rare objects, but there are several indications that they are not quite as unusual as they seem. In fact, the class of neighborly polytopes is believed to be very rich. Thus, Shemer [She82] has shown that for fixed even d the number of nonisomorphic neighborly d-polytopes with n vertices grows superexponentially with n. Also, many of the (0,1)-polytopes studied in combinatorial optimization turn out to be at least 2-neighborly. Both these effects illustrate that "neighborliness" is not an isolated phenomenon.

OPEN PROBLEMS

1. Can every neighborly d-polytope $P \subseteq \mathbb{R}^d$ with n vertices be extended by a new vertex $v \in \mathbb{R}^d$ to a neighborly polytope $P' := \mathrm{conv}(P \cup \{v\})$ with $n+1$ vertices? [She82, p. 314]

2. It is a classic problem of Perles whether every simplicial polytope is a quotient of a neighborly polytope. (For polytopes with at most $d+4$ vertices this was recently confirmed by Hund [Hun95].)

3. In some models of random polytopes is seems that

   - one obtains a neighborly polytope with high probability (which increases rapidly with the dimension of the space),
   - the most probable combinatorial type is a cyclic polytope,
   - but still this probability of a cyclic polytope tends to zero.

   However, none of this has been proved. (See Bokowski and Sturmfels [BS89, p. 101], Bokowski, Richter-Gebert, and Schindler [BRS92], and Vershik and Sporychev [VS92].)

(0,1)-POLYTOPES

There is a $(0,1)$-polytope (given in terms of a $\mathcal{V}$-presentation) associated with every finite set system $\mathcal{S} \subseteq 2^E$ (where E is a finite set, and 2^E denotes the collection of all of its subsets), via

$$P[\mathcal{S}] \;:=\; \text{conv}\Big\{ \sum_{i \in F} e^i \;\Big|\; F \in \mathcal{S} \Big\} \;\subseteq\; \mathbb{R}^E.$$

In combinatorial optimization, there is an extensive literature available on $\mathcal{H}$-presentations of special $(0,1)$-polytopes, such as

- the ***traveling salesman polytopes*** T^n, where E is the edge set of a complete graph K_n, and $\mathcal{F}$ is the set of all $(n-1)!$ Hamilton cycles (simple circuits through all the vertices) in E (see Grötschel and Padberg [GP85]);
- the ***cut*** and ***equicut polytopes***, where E is again the edge set of a complete graph, and $\mathcal{S}$ represents, for example, the family of all cuts, or all equicuts, of the graph (see Deza and Laurent [DL97]).

Besides their importance for combinatorial optimization, there is a great deal of interesting polytope theory associated with such polytopes. For a striking example, see the equicut polytopes used by Kahn and Kalai [KK93] in their recent disproof of Borsuk's conjecture.

Despite the detailed structure theory for the "special" $(0,1)$-polytopes of combinatorial optimization, there is very little known about "general" $(0,1)$-polytopes. For example, what is the "typical," or the maximal, number of facets of a $(0,1)$-polytope? What is the maximal number of faces in a 2-dimensional projection? (Such questions are not only intrinsically interesting; their answers might also provide new clues for basic questions of linear and combinatorial optimization.)

13.1.5 THREE-DIMENSIONAL POLYTOPES AND PLANAR GRAPHS

GLOSSARY

d-connected graph: A connected graph that remains connected if any $d-1$ vertices are deleted.

Drawing of a graph: A representation in the plane where the vertices are represented by distinct points, and simple Jordan arcs are drawn between the pairs of adjacent vertices.

Planar graph: A graph that can be drawn in the plane with Jordan arcs that are disjoint except for their endpoints.

Realization space: The set of all coordinatizations of a combinatorial structure, modulo affine coordinate transformations. (See Section 6.3.2.)

Isotopy property: A combinatorial structure (such as a combinatorial type of polytope) has the isotopy property if any two realizations can be deformed into each other continuously, while maintaining the combinatorial type. Equivalently, the isotopy property holds for a combinatorial structure if and only if its realization space is connected.

THEOREM 13.1.3 *Steinitz's Theorem* [SR34]

For every 3-dimensional polytope P, the graph $G(P)$ is a planar, 3-connected graph. Conversely, for every planar 3-connected graph, there is a unique combinatorial type of 3-polytope P with $G(P) \cong G$.

Furthermore, the realization space $\mathcal{R}(P)$ of a combinatorial type of 3-polytope is homeomorphic to $\mathbb{R}^{f_1(P)-6}$, and contains rational points. In particular, 3-dimensional polytopes have the isotopy property, and they can be realized with integer vertex coordinates.

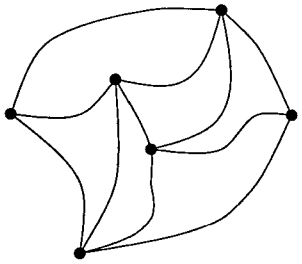


FIGURE 13.1.7
A (planar drawing of a) 3-connected, planar, unnamed graph. The formidable task of any proof of Steinitz's theorem is to construct a 3-polytope with this graph.

There are two essentially different ways known to prove Steinitz's theorem. The first one [SR34] provides a construction sequence for any type of 3-polytope, starting from a tetrahedron, and using only local operations such as cutting off vertices and polarity. The second type of proof realizes any combinatorial type by a global minimization argument, which as an intermediate step provides a special planar representation of the graph by a framework with a positive self-stress [McM94, OS94].

OPEN PROBLEMS

Because of Steinitz's theorem and its extensions and corollaries, the theory of 3-dimensional polytopes is quite complete and satisfactory. Nevertheless, some basic open problems remain.

1. It can be shown that every combinatorial type of 3-polytope with n vertices can be realized with integer coordinates in $\{1, 2, \ldots, 43^n\}^3$ (J. Richter-Gebert, improving on Onn and Sturmfels [OS94]), but it is not clear whether the bound of 43^n can be replaced by a polynomial bound.
2. If P has a group G of symmetries, then it also has a symmetric realization. However, it is not clear whether the space of all G-symmetric realizations $\mathcal{R}^{\mathrm{G}}(P)$ is still homeomorphic to some $\mathbb{R}^k$. (It does not contain rational points in general, e.g., for the icosahedron!)

13.1.6 FOUR-DIMENSIONAL POLYTOPES AND SCHLEGEL DIAGRAMS

GLOSSARY

Schlegel diagram: A $(d-1)$-dimensional representation $\mathcal{D}(P, F)$ of a d-dimensional polytope P, obtained as follows. Take a point of view very close to (an

interior point of) the facet F, and let $\mathcal{D}(P,F)$ be the decomposition of F given by all the other facets of P, as seen from this point of view.

(d−1)-diagram: A polytopal decomposition $\mathcal{D}$ of a $(d-1)$-polytope F such that (1) $\mathcal{D}$ is a polytopal complex (i.e., a finite collection of polytopes closed under taking faces, such that any intersection of two polytopes in the complex is a face of each), and
(2) the intersection of any polytope in $\mathcal{D}$ with the boundary of F is a face of F (which may be empty).

Basic primary semialgebraic set defined over $\mathbb{Z}$: The solution set $S \subseteq \mathbb{R}^k$ of a finite set of equations and strict inequalities of the form $f_i(x) = 0$ resp. $g_j(x) > 0$, where the f_i and g_j are polynomials in k variables with integer coefficients.

Stable equivalence: Equivalence relation between semialgebraic sets generated by rational changes of coordinates and certain types of "stable" projections with contractible fibers. (See Richter-Gebert [Ric96, Section 2.5].)

In particular, if two sets are stably equivalent, then they have the same homotopy type, and they have the same arithmetic properties with respect to subfields of $\mathbb{R}$; e.g., either both or neither of them contain a rational point.

The situation for 4-polytopes is fundamentally different from that for 3-dimensional polytopes. One reason is that there is no similar reduction of 4-polytope theory to a combinatorial (graph) problem.

The main results about graphs of d-polytopes are that they are d-connected (Balinski), and that each contains a subdivision of the complete graph on $d+1$ vertices, $K_{d+1} = G(T_d)$ (Grünbaum). In particular, all graphs of 4-polytopes are 4-connected, and none of them is planar. (See also Chapter 17.)

Schlegel diagrams provide a reasonably efficient tool for visualization of 4-polytopes: we have a fighting chance to understand some of their theory in terms of the 3-dimensional (!) geometry of Schlegel diagrams.

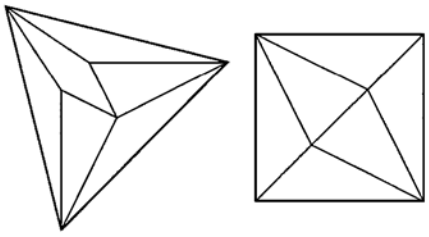


FIGURE 13.1.8
Two Schlegel diagrams of our unnamed 3-polytope, the first based on a triangle facet, the second on the "bottom square."

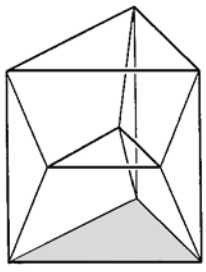


FIGURE 13.1.9
A Schlegel diagram of the product of two triangles. (This is a 4-dimensional polytope with 6 triangular prisms as facets, any two of them adjacent!)

A $(d-1)$-diagram is a polytopal complex that "looks like" a Schlegel diagram, although there are diagrams (even 2-diagrams) that are not Schlegel diagrams. The situation is somewhat nicer for *simple* 4-polytopes. These are determined by their graphs (Kalai), and they can be understood in terms of 3-diagrams: all simple 3-diagrams are projections of genuine 4-dimensional polytopes (Whiteley).

The fundamental difference between the theories for polytopes in dimensions 3 and 4 is most apparent in the contrast between Steinitz's theorem and the following (very recent) result, which states simply that all the "nice" properties of 3-polytopes established in Steinitz's theorem fail dramatically for 4-dimensional polytopes.

THEOREM 13.1.4 *Richter-Gebert's Universality Theorem for 4-Polytopes*
The realization space of a 4-dimensional polytope can be "arbitrarily wild": for every basic primary semialgebraic set S defined over $\mathbb{Z}$ there is a 4-dimensional polytope $P[S]$ whose realization space $\mathcal{R}(P[S])$ is stably equivalent to S.

In particular, this implies the following.

- *The isotopy property fails for 4-dimensional polytopes.*
- *There are nonrational 4-polytopes: combinatorial types that cannot be realized with rational vertex coordinates.*
- *The coordinates needed to represent all combinatorial types of rational 4-polytopes with integer vertices grow doubly exponentially with $f_0(P)$.*

The complete proof of this universality theorem is given in [Ric96]. One key component of the proof corresponds to another failure of a 3-dimensional phenomenon in dimension 4: for any facet (2-face) F of a 3-dimensional polytope P, the shape of F can be arbitrarily prescribed; in other words, the canonical map of realization spaces $\mathcal{R}(P) \longrightarrow \mathcal{R}(F)$ is always surjective. Richter-Gebert shows that a similar statement fails in dimension 4, even if F is a 2-dimensional pentagonal face: see Figure 13.1.10 for the case of a hexagon.

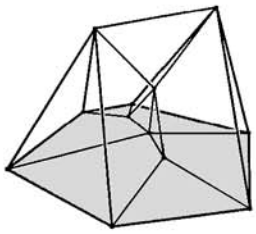


FIGURE 13.1.10
Schlegel diagram of a 4-dimensional polytope with 8 facets and 12 vertices, for which the shape of the base hexagon cannot be prescribed arbitrarily.

A problem that is left open is the structure of the realization spaces of simplicial 4-polytopes. All that is available now is a universality theorem for simplicial polytopes without a dimension bound (see Section 6.3.4), and a single example of a simplicial 4-polytope that violates the isotopy property, by Bokowski, Ewald, and Kleinschmidt [BEK84].

13.1.7 POLYTOPES WITH FEW VERTICES—GALE DIAGRAMS

GLOSSARY

Polytope with few vertices: A polytope that has only a few more vertices than its dimension; usually a d-polytope with at most $d+4$ vertices.

(Affine) Gale diagram: A configuration of n (positive and negative) points in affine space $\mathbb{R}^{n-d-2}$ that encodes a d-polytope with n vertices uniquely up to projective transformations.

The computation of a Gale diagram involves only simple linear algebra. For this, let $V \in \mathbb{R}^{d\times n}$ be a matrix whose columns consist of coordinates for the vertices of a d-polytope. For simplicity, we assume that P is not a pyramid, and that the vertices $\{v^1, \ldots, v^{d+1}\}$ affinely span $\mathbb{R}^d$. Let $\widetilde{V} \in \mathbb{R}^{(d+1)\times n}$ be obtained from V by adding an extra (terminal) row of ones. The vector configuration given by the columns of $\widetilde{V}$ represents the *oriented matroid* of P; see Chapter 6.

Now perform row operations on the matrix $\widetilde{V}$ to get it into the form $\widetilde{V} \sim (I_{d+1}|A)$, where I_{d+1} denotes a unit matrix, and $A \in \mathbb{R}^{(d+1)\times(n-d-1)}$ is a real matrix. (The row operations do not change the oriented matroid.) The columns of the matrix $\widetilde{V}^* := (-A^T|I_{n-d-1}) \in \mathbb{R}^{(n-d-1)\times n}$ then represent the dual oriented matroid. We find a vector $a \in \mathbb{R}^{n-d-1}$ that has nonzero scalar product with all the columns of $\widetilde{V}^*$, divide each column w^* of $\widetilde{V}^*$ by the value $\langle a, w\rangle$, and delete from the resulting matrix any row that affinely depends on the others, thus obtaining a matrix $W \in \mathbb{R}^{(n-d-2)\times n}$. The columns of W give a colored point configuration in $\mathbb{R}^{n-d-2}$, where *black* points are used for the columns where $\langle a, w\rangle > 0$, and *white* points for the others. This colored point configuration represents an affine Gale diagram of P.

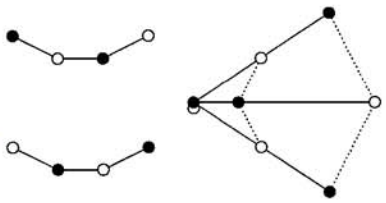


FIGURE 13.1.11
Two affine Gale diagrams of 4-dimensional polytopes: for a noncyclic neighborly polytope with 8 vertices, and for the polar (with 8 vertices) of the polytope with 8 facets from Figure 13.1.10, for which the shape of a hexagon face cannot be prescibed arbitrarily.

It turns out that an affine configuration of colored points (consisting of n points that affinely span $\mathbb{R}^e$) represents a polytope (with n vertices, of dimension $n-e-2$) if and only if the following criterion is met: For any hyperplane spanned by some of the points, and for each side of it, the number of black points on this side, plus the number of white points on the other side, is at least 2.

The final information one needs is how to read off properties of a polytope from its affine Gale diagram. Here the criterion is that a set of points represents a face if and only if the following condition is satisfied: the colored points *not* in the set support an affine dependency, with positive coefficients on the black points, and with negative coefficients on the white points. Equivalently, the convex hull of all

the black points not in our set, and the convex hull of all the white points not in the set, intersect in their relative interiors.

Affine Gale diagrams have been *very* successfully used to study and classify polytopes with few vertices.

d+1 vertices: The only d-polytopes with d+1 vertices are the d-simplices.

d+2 vertices: There are exactly $\lfloor d^2/4 \rfloor$ combinatorial types of d-polytopes with d+2 vertices; among these, $\lfloor d/2 \rfloor$ types are simplicial. This corresponds to the situation of 0-dimensional affine Gale diagrams.

d+3 vertices: All d-polytopes with d+3 vertices are realizable with (small) integral coordinates and satisfy the isotopy property: all this can be easily analyzed in terms of 1-dimensional affine Gale diagrams.

d+4 vertices: Here anything can go wrong: the universality theorem for oriented matroids of rank 3 yields a universality theorem for simplicial d-polytopes with d+4 vertices. (See Section 6.3.4.)

We refer to [Zie95, Lecture 6] for a detailed introduction to affine Gale diagrams.

13.2 METRIC PROPERTIES

The combinatorial data of a polytope—vertices, edges, ..., facets—have their counterparts in genuine geometric data, such as face volumes, surface areas, quermass-integrals, and the like. In this second half of the chapter, we give a brief sketch of some key geometric concepts related to polytopes.

However, the topics of combinatorial and of geometric invariants are not disjoint at all: much of the beauty of the theory stems from the subtle interplay between the two sides. Thus, the computation of volumes inevitably leads to the construction of triangulations (explicitly or implicitly), mixed volumes lead to mixed subdivisions of Minkowski sums (one "hot topic" for current research in the area), quermassintegrals relate to face enumeration, and so on.

Furthermore, the study of polytopes yields a powerful approach to the theory of convex bodies: sometimes one can extend properties of polytopes to arbitrary convex bodies by approximation [Sch93]. (However, there are also properties valid for polytopes that fail for convex bodies in general. This bug/feature is designed to keep the game interesting.)

13.2.1 VOLUME AND SURFACE AREA

GLOSSARY

Volume of a d-simplex T: $V(T) = \left|\det \begin{pmatrix} v^0 & \cdots & v^d \\ 1 & \cdots & 1 \end{pmatrix}\right|/d!$, where $T = \operatorname{conv}\{v^0, \ldots, v^d\}$ with $v^0, \ldots, v^d \in \mathbb{R}^d$.

Subdivision of a polytope P: A collection of polytopes $P_1, \ldots, P_l \subseteq \mathbb{R}^d$ such that $P = \bigcup P_i$, and for $i \neq j$ we have that $P_i \cap P_j$ is a proper face of P_i and P_j (possibly empty). In this case we write $P = \uplus P_i$.

Triangulation of a polytope: A subdivision into simplices. (See Chapter 14.)

Volume of a d-polytope: $\sum_{T\in\Delta(P)} V(T)$, where $\Delta(P)$ is a triangulation of P.

k-volume $V^k(P)$ of a k-polytope $P \subseteq \mathbb{R}^d$: The volume of P, computed with respect to the k-dimensional Euclidean measure induced on $\mathrm{aff}(P)$.

Surface area of a d-polytope P: $\sum_{T\in\Delta(P),\, F\in\mathcal{F}_{d-1}(P)} V^{d-1}(T \cap F)$, where $\Delta(P)$ is a triangulation of P.

The volume $V(P)$ (i.e., the d-dimensional Lebesgue measure) and the surface area $F(P)$ of a d-polytope $P \subseteq \mathbb{R}^d$ can be derived from any triangulation of P, since volumes of simplices are easy to compute. The crux for this is in the (efficient?) generation of a triangulation, a topic on which Chapters 14 and 22 of this Handbook have more to say.

The following recursive approach only implicitly generates a triangulation, but derives explicit volume formulas. Let $P \subseteq \mathbb{R}^d$ $(P \neq \emptyset)$ be a polytope. If $d = 0$ then we set $V(P) = 1$. Otherwise we set $\mathcal{S}_{d-1}(P) := \{u \in S^{d-1} \mid \dim(H(P,u) \cap P) = d-1\}$, and use this to define the volume of P as

$$V(P) \;:=\; \frac{1}{d} \sum_{u\in\mathcal{S}_{d-1}(P)} h(P,u) \cdot V^{d-1}(H(P,u) \cap P).$$

Thus, for any d-polytope the volume is a sum of its facet volumes, each weighted by $1/d$ times its signed distance from the origin. Geometrically, this can be interpreted as follows. Assume for simplicity that the origin is in the interior of P. Then the collection $\{\mathrm{conv}(F \cup \{0\}) \mid F \in \mathcal{F}_{d-1}(P)\}$ is a subdivision of P into d-dimensional pyramids, where the base of $\mathrm{conv}(F \cup \{0\})$ has $(d-1)$-dimensional volume $V^{d-1}(F)$—to be computed recursively, the height of the pyramid is $h(P, u^F)$, and thus its volume is $\frac{1}{d} h(P, u^F) \cdot V^{d-1}(F)$; compare to Figure 13.2.1. (The formula remains valid even if the origin is outside P or on its boundary.)

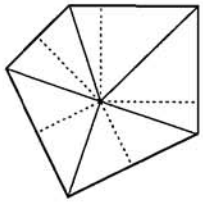


FIGURE 13.2.1
This pentagon, with the origin in its interior, is decomposed into five pyramids (triangles), each with one of the pentagon facets (edges) F_i as its base. For each pyramid, the height, of length $h(P, u^{F_i})$, is drawn as a dotted line.

Note that $V(P) \geq 0$. This holds with strict inequality if and only if the polytope P has full dimension d. The surface area $F(P)$ can also be expressed as

$$F(P) \;=\; \sum_{u\in\mathcal{S}_{d-1}(P)} V^{d-1}(H(P,u) \cap P).$$

Thus for a d-polytope the surface area is the sum of the $(d-1)$-volumes of its facets. If $\dim(P) = d-1$, then $F(P)$ is twice the $(d-1)$-volume of P. One has $F(P) = 0$ if and only if $\dim(P) < d-1$.

Both the volume and the surface area are continuous, monotone, and invariant with respect to rigid motions. $V(\cdot)$ is homogeneous of degree d, i.e., $V(\mu P) =$

$\mu^d V(P)$ for $\mu \geq 0$, and $F(\cdot)$ is homogeneous of degree $d-1$. For further properties of the functionals $V(\cdot)$ and $F(\cdot)$ see [Had57] and [BF34].

Table 13.2.1 gives the numbers of k-faces, the volume, and the surface area of the d-cube C_d (with edge length 2), of the cross-polytope C_d^{Δ} with edge length $\sqrt{2}$, and of the regular simplex T_d with edge length $\sqrt{2}$.

TABLE 13.2.1

POLYTOPE	$f_k(\cdot)$	VOLUME	SURFACE AREA
C_d	$2^{d-k}\binom{d}{k}$	2^d	$2d \cdot 2^{d-1}$
C_d^{Δ}	$2^{k+1}\binom{d}{k+1}$	$\frac{2^d}{d!}$	$2^d \frac{\sqrt{d}}{(d-1)!}$
T_d	$\binom{d+1}{k+1}$	$\frac{\sqrt{d+1}}{d!}$	$(d+1) \cdot \frac{\sqrt{d}}{(d-1)!}$

13.2.2 MIXED VOLUMES

GLOSSARY

Volume polynomial: The volume of the Minkowski sum $\lambda_1 P_1 + \lambda_2 P_2 + \ldots + \lambda_r P_r$, which is a homogeneous polynomial in $\lambda_1, \ldots, \lambda_r$. (Here the P_i may be convex polytopes of any dimension, or more general (closed, bounded) convex sets.)

Mixed volumes: The coefficients of the volume polynomial of $P_1, \ldots, P_r$.

Normal cone: The normal cone $N(F, P)$ of a face is the set of all vectors $v \in \mathbb{R}^d$ such that the supporting hyperplane $H(P, v)$ contains F, i.e.,

$$N(F, P) = \left\{ v \in \mathbb{R}^d \mid F \subseteq H(P, v) \cap P \right\}.$$

THEOREM 13.2.1 *Mixed Volumes*

Let $P_1, \ldots, P_r \subseteq \mathbb{R}^d$ be polytopes, $r \geq 1$, and $\lambda_1, \ldots, \lambda_r \geq 0$. The volume of $\lambda_1 P_1 + \ldots + \lambda_r P_r$ is a homogeneous polynomial in $\lambda_1, \ldots, \lambda_r$ of degree d. Thus it can be written in the form

$$V(\lambda_1 P_1 + \ldots + \lambda_r P_r) = \sum_{(i(1),\ldots,i(d)) \in \{1,2,\ldots,r\}^d} \lambda_{i(1)} \cdots \lambda_{i(d)} \cdot V(P_{i(1)}, \ldots, P_{i(d)}).$$

The coefficients in this expansion are symmetric in their indices. Furthermore, the coefficient $V(P_{i(1)}, \ldots, P_{i(d)})$ depends only on $P_{i(1)}, \ldots, P_{i(d)}$. It is called the ***mixed volume*** *of the polytopes $P_{i(1)}, \ldots, P_{i(d)}$.*

With the abbreviation

$$V(P_1, k_1; \ldots; P_r, k_r) := V(\underbrace{P_1, \ldots, P_1}_{k_1 \text{ times}}, \ldots, \underbrace{P_r, \ldots, P_r}_{k_r \text{ times}}),$$

the polynomial becomes

$$V(\lambda_1 P_1 + \ldots + \lambda_r P_r) \;=\; \sum_{\substack{k_1,\ldots,k_r \geq 0 \\ k_1+\ldots+k_r=d}} \binom{d}{k_1,\ldots,k_r} \lambda_1^{k_1} \cdots \lambda_r^{k_r} V(P_1, k_1; \ldots; P_r, k_r).$$

In particular, the volume of the polytope P_i is given by the mixed volume $V(P_1, 0; \ldots; P_i, d; \ldots; P_r, 0)$. The theorem is also valid for arbitrary convex bodies: a good example where the general case can be derived from the polytope case by approximation. For more about the properties of mixed volumes from different points of view see Schneider [Sch93], Sangwine-Yager [San93], and McMullen [McM93].

The definition of the mixed volumes as coefficients of a polynomial is somewhat unsatisfactory. Only recently, Schneider [Sch94] gave the following explicit rule, which generalizes an earlier result of Betke [Bet92] for the case $r = 2$. It uses information about the normal cones at certain faces. For this, note that $N(F, P)$ is a finitely generated cone, which can be written explicitly as the sum of the orthogonal complement of $\mathrm{aff}(P)$ and the positive hull of those unit vectors u that are both parallel to $\mathrm{aff}(P)$ and induce supporting hyperplanes $H(P, u)$ that contain a facet of P including F. Thus, for $P \subseteq \mathbb{R}^d$ the dimension of $N(F, P)$ is $d - \dim(F)$.

THEOREM 13.2.2 *Schneider's Summation Formula*
Let $P_1, \ldots, P_r \subseteq \mathbb{R}^d$ be polytopes, $r \geq 2$. Let $x^1, \ldots, x^r \in \mathbb{R}^d$ such that $x^1 + \ldots + x^r = 0$, $(x^1, \ldots, x^r) \neq (0, \ldots, 0)$, and

$$\bigcap_{i=1}^{r} \left(\mathrm{relint} N(F_i, P_i) - x^i\right) \;=\; \emptyset$$

whenever F_i is a face of P_i and $\dim(F_1) + \ldots + \dim(F_r) > d$. Then

$$\binom{d}{k_1,\ldots,k_r} V(P_1, k_1; \ldots; P_r, k_r) = \sum_{(F_1,\ldots,F_r)} V(F_1 + \ldots + F_r),$$

where the summation extends over the r-tuples $(F_1, \ldots, F_r)$ of k_i-faces F_i of P_i with $\dim(F_1 + \ldots + F_r) = d$ and $\bigcap_{i=1}^{r} \left(N(F_i, P_i) - x^i\right) \neq \emptyset$.

The choice of the vectors $x^1, \ldots, x^r$ implies that the selected k_i-faces $F_i \subseteq P_i$ of a summand $F_1 + \ldots + F_r$ are contained in complementary subspaces. Hence one may also write

$$\binom{d}{k_1,\ldots,k_r} V(P_1, k_1; \ldots; P_r, k_r) = \sum_{(F_1,\ldots,F_r)} [F_1, \ldots, F_r] \cdot V^{k_1}(F_1) \cdots V^{k_r}(F_r),$$

where $[F_1, \ldots, F_r]$ denotes the volume of the parallelepiped that is the sum of unit cubes in the affine hulls of $F_1, \ldots, F_r$.

Finally, we remark that the selected sums of faces in the formula of the theorem form a subdivision of the polytope $P_1 + \ldots + P_r$, i.e.,

$$P_1 + \ldots + P_r \;=\; \biguplus_{(F_1,\ldots,F_r)} (F_1 + \ldots + F_r)\,.$$

See Figure 13.2.2 for an example.

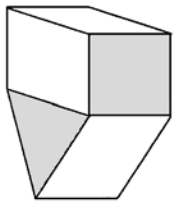


FIGURE 13.2.2
Here the Minkowski sum of a square P_1 and a triangle P_2 is decomposed into translates of P_1 and of P_2 (this corresponds to two summands with $F_1 = P_1$ resp. $F_2 = P_2$), together with three "mixed" faces that arise as sums $F_1 + F_2$, where F_1 and F_2 are faces of P_1 and P_2 (corresponding to summands with $\dim(F_1) = \dim(F_2) = 1$).

VOLUMES OF ZONOTOPES

If all summands in a Minkowski sum $Z = P_1 + \ldots + P_r$ are line segments, say $P_i = p^i + [0,1]z^i = \text{conv}\{p^i, p^i + z^i\}$ with $p^i, z^i \in \mathbb{R}^d$ for $1 \le i \le r$, then the resulting polytope Z is a zonotope. In this case the summation rule immediately gives $V(P_1, k_1; \ldots; P_r, k_r) = 0$ if the vectors

$$\underbrace{z^1, \ldots, z^1}_{k_1 \text{ times}}, \ldots, \underbrace{z^r, \ldots, z^r}_{k_r \text{ times}}$$

are linearly dependent. (This can also be seen directly from dimension considerations.) Otherwise, for $k_{i(1)} = k_{i(2)} = \ldots = k_{i(d)} = 1$, say,

$$V(P_1, k_1; \ldots; P_r, k_r) = \frac{1}{d!} \left| \det\left(z^{i(1)}, z^{i(2)}, \ldots, z^{i(d)}\right) \right|.$$

Therefore, one obtains McMullen's formula for the volume of the zonotope Z:

$$V(Z) = \sum_{1 \le i(1) < i(2) < \cdots < i(d) \le r} \left| \det(z^{i(1)}, \ldots, z^{i(d)}) \right|.$$

13.2.3 QUERMASSINTEGRALS AND INTRINSIC VOLUMES

GLOSSARY

***i*-th quermassintegral $W_i(P)$:** The mixed volume $V(P, d-i; B_d, i)$ of a polytope P and the d-dimensional unit ball B_d.

κ_d: The volume (Lebesgue measure) of B_d. (Hence $\kappa_0 = 1$, $\kappa_1 = 2$, $\kappa_2 = \pi$, etc.)

***i*-th intrinsic volume $V_i(P)$:** The $(d-i)$-th quermassintegral, scaled by the constant $\binom{d}{i}/\kappa_{d-i}$.

Outer parallel body of P at distance λ: The convex body $P + \lambda B_d$ for some $\lambda > 0$.

External angle $\gamma(F, P)$: The volume of $\big(\text{lin}(F - x^F) + N(F, P)\big) \cap B_d$ divided by κ_d, for $x^F \in \text{relint}(F)$. Thus $\gamma(F, P)$ is the "fraction of $\mathbb{R}^d$ taken up by $\text{lin}(F - x^F) + N(F, P)$." Equivalently, the external angle at a k-face F is the fraction of the spherical volume of S covered by $N(F, P) \cap S$, where S denotes the $(d-k-1)$-dimensional unit sphere in $\text{lin}(N(F, P))$.

Internal angle $\beta(F,G)$ for faces $F \subseteq G$: The "fraction" of $\mathrm{lin}\{G - x^F\}$ taken up by the cone $\mathrm{pos}\{x - x^F \mid x \in G\}$, for $x^F \in \mathrm{relint}(F)$. (A detailed discussion of relations between external and internal angles can be found in McMullen [McM75].)

The quermassintegrals are generalizations of both the volume and the surface area of P. In fact, they can also be seen as the continuous convex geometry analogs of face numbers.

For a polytope $P \subseteq \mathbb{R}^d$ and the d-dimensional unit ball B_d, the mixed volume formula, applied to the outer parallel body $P + \lambda B_d$, gives

$$V(P + \lambda B_d) = \sum_{i=0}^{d} \binom{d}{i} \lambda^i W_i(P),$$

with the convention $W_i(P) = V(P, d-i; B_d, i)$. This formula is known as the ***Steiner polynomial***. The mixed volume $W_i(P)$, the i-th quermassintegral of P, is an important quantity and of significant geometric interest [Had57] [Sch93]. As special cases, $W_0(P) = V(P)$ is the volume, $dW_1(P) = F(P)$ is the surface area, and $W_d(P) = \kappa_d$.

For the geometric interpretation of $W_i(P)$ for polytopes, we use a normalization of the quermassintegrals due to McMullen [McM75]: For $0 \le i \le d$, the i-th intrinsic volume of P is defined by

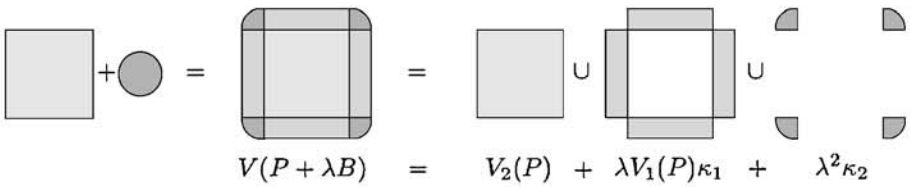
$$V_i(P) \;:=\; \frac{\binom{d}{i}}{\kappa_{d-i}} W_{d-i}(P).$$

With this notation the Steiner polynomial can be written as

$$V(P + \lambda B_d) = \sum_{i=0}^{d} \lambda^{d-i} \kappa_{d-i} V_i(P).$$

(See Figure 13.2.3 for an example.) $V_d(P)$ is the volume of P, $V_{d-1}(P)$ is half the surface area, and $V_0(P) = 1$. One advantage of this normalization is that the intrinsic volumes are unchanged if P is embedded in some Euclidean space of different dimension. Thus, for $\dim(P) = k \le d$, $V_k(P)$ is the ordinary k-volume of P with respect to the Euclidean structure induced in $\mathrm{aff}(P)$.

FIGURE 13.2.3
The Minkowski sum of a square P with a ball λB^2 yields the outer parallel body. This outer parallel body can be decomposed into pieces, whose volumes, $V(P)$, $\lambda V_1(P)\kappa_1$, and $\lambda^2\kappa_2$, correspond to the three terms in the Steiner polynomial.



For a $(\dim(P) - 2)$-face F, the concept of external angle (see the glossary) reduces to the "usual" concept: then the external angle is given by $\frac{1}{2\pi} \arccos\langle u^{F_1}, u^{F_2} \rangle$

for unit normal vectors $u^{F_1}, u^{F_2} \in S^{d-1}$ to the facets F_1, F_2 with $F_1 \cap F_2 = F$. (One has $\gamma(P,P) = 1$ for the polytope itself and $\gamma(F,P) = 1/2$ for each facet F.) Using this concept, we get

$$V_k(P) = \sum_{F \in \mathcal{F}_k(P)} \gamma(F,P) \cdot V^k(F).$$

SOME COMPUTATIONS

In principle, one can use the external angle formula to determine the intrinsic volumes of a given polytope, but in general it is hard to calculate external angles. Indeed, for the computation of spherical volumes there are explicit formulas only in small dimensions.

In what follows, we give formulas for the intrinsic volumes of the polytopes C_d, C_d^Δ, and T_d. For this, we identify the k-faces of C_d with the k-cube C_k and the k-faces of C_d^Δ and of T_d with T_k, for $0 \le k < d$.

The case of the cube C_d is rather trivial. Since $\gamma(C_k, C_d) = 2^{-(d-k)}$ one gets (see Table 13.2.1)

$$V_k(C_d) = 2^k \binom{d}{k}.$$

For the regular simplex T_d we have

$$V_k(T_d) = \binom{d+1}{k+1} \cdot \frac{\sqrt{k+1}}{k!} \cdot \gamma(T_k, T_d).$$

An explicit formula for the external angles of a regular simplex by Ruben (see [Had79]) is:

$$\gamma(T_k, T_d) = \sqrt{\frac{k+1}{\pi}} \int_{-\infty}^{\infty} e^{-(k+1)x^2} \left(\frac{1}{\sqrt{\pi}} \int_{-\infty}^{x} e^{-y^2} dy \right)^{d-k} dx.$$

For the regular cross-polytope we find for $k \le d-1$ that

$$V_k(C_d^\Delta) = 2^{k+1} \binom{d}{k+1} \cdot \frac{\sqrt{k+1}}{k!} \cdot \gamma(T_k, C_d^\Delta).$$

For this, the external angles of C_d^Δ were determined by Betke and Henk [BH93]:

$$\gamma(T_k, C_d^\Delta) = \sqrt{\frac{k+1}{\pi}} \int_{0}^{\infty} e^{-(k+1)x^2} \left(\frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-y^2} dy \right)^{d-k-1} dx.$$

AN APPLICATION

External angles and internal angles play a crucial role in work by Affentranger and Schneider [AS92] (see also [BV94]), who computed the expected number of k-faces of the orthogonal projection of a polytope $P \subseteq \mathbb{R}^d$ onto a randomly chosen isotropic subspace of dimension n. Let $E[f_k(P;n)]$ be that number. Then for $0 \le k < n \le d-1$ it was shown that

$$E[f_k(P;n)] \;=\; 2 \sum_{m \ge 0} \sum_{F \in \mathcal{F}_k(P)} \sum_{\substack{G \in \mathcal{F}_{n-1-2m}(P) \\ F \subseteq G}} \beta(F,G)\gamma(G,P),$$

where $\beta(F,G)$ is the internal angle of the face F with respect to a face $G \supseteq F$.

In the sequel we apply the above formula to the polytopes C_d, C_d^Δ, and T_d. For the cubes one has $\beta(C_k, C_l) = (1/2)^{l-k}$, while the number of l-faces of C_d containing any given k-face is equal to $\binom{d-k}{l-k}$. Hence

$$E[f_k(C_d;n)] = 2\binom{d}{k}\sum_{m\geq 0}\binom{d-k}{n-1-k-2m}.$$

In particular, $E[f_k(C_d;d-1)] = (2^{d-k}-2)\binom{d}{k}$.

For the cross-polytope C_d^Δ the number of l-faces that contain a k-face is equal to $2^{l-k}\binom{d-k-1}{l-k}$. Thus

$$E[f_k(C_d^\Delta;n)] = 2\binom{d}{k+1}\sum_{m\geq 0}2^{n-2m}\binom{d-k-1}{n-1-k-2m}\beta(T_k, T_{n-1-2m})\gamma(T_{n-1-2m}, C_d^\Delta).$$

In the same way one obtains for T_d

$$E[f_k(T_d;n)] = 2\binom{d+1}{k+1}\sum_{m\geq 0}\binom{d-k}{n-1-k-2m}\beta(T_k, T_{n-1-2m})\gamma(T_{n-1-2m}, T_d).$$

For the last two formulas one needs the internal angles $\beta(T_k, T_l)$ of the regular simplex T_d, for $0 \leq k \leq l \leq d$. For this, one has the following complex integral [BH95]:

$$\beta(T_k, T_l) = \frac{(k+1+l)^{1/2}(k+1)^{(l-1)/2}}{\pi^{(l+1)/2}}\int_{-\infty}^{\infty} e^{-w^2}\left(\int_0^\infty e^{-(k+1)y^2+2iwy}dy\right)^l dw.$$

Using this formula one can determine the asymptotic behavior of $E[f_k(C_d^\Delta;n)]$ and $E[f_k(T_d;n)]$ as n tends to infinity [BH95].

13.3 SOURCES AND RELATED MATERIAL

FURTHER READING

The classic account of the combinatorial theory of convex polytopes was given by Grünbaum in 1967 [Grü67]. It inspired and guided a great part of the subsequent research in the field. Besides the related chapters of this Handbook, we refer to the recent handbook surveys by Klee and Kleinschmidt [KK95] and by Bayer and Lee [BL93] for further reading.

For the geometric theory of convex bodies, and especially convex polytopes, a classic is Bonnesen and Fenchel [BF34]. Here we refer to the Handbook of Convex Geometry [GW93] for recent surveys, and to Schneider [Sch93] for an excellent recent monograph. As for the algorithmic aspects of computing volumes, etc., we

refer to Chapter 27 of this Handbook, on Computational Convexity, and to the additional references given there.

RELATED CHAPTERS

Chapter 3: Tilings
Chapter 6: Oriented matroids
Chapter 7: Lattice points and lattice polytopes
Chapter 9: Discrete aspects of stochastic geometry
Chapter 14: Subdivisions and triangulations of polytopes
Chapter 15: Face numbers of polytopes and complexes
Chapter 16: Symmetry of polytopes and polyhedra
Chapter 17: Polytope skeletons and paths
Chapter 19: Convex hull computations
Chapter 22: Triangulations
Chapter 27: Computational convexity
Chapter 51: Crystals and quasicrystals

REFERENCES

[AS92] F. Affentranger and R. Schneider. Random projections of regular simplices. *Discrete Comput. Geom.*, 7:219–226, 1992.

[BV94] Y. Baryshnikov and R.A. Vitale. Regular simplices and Gaussian samples. *Discrete Comput. Geom.*, 11:141–147, 1994.

[BL93] M.M. Bayer and C.W. Lee. Combinatorial aspects of convex polytopes. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 485–534. North-Holland, Amsterdam, 1993.

[Bet92] U. Betke. Mixed volumes of polytopes. *Arch. Math.*, 58:388–391, 1992.

[BH93] U. Betke and M. Henk. Intrinsic volumes and lattice points of crosspolytopes. *Monatshefte Math.*, 115:27–33, 1993.

[BLS+93] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G.M. Ziegler. *Oriented Matroids.* Volume 46 of *Encyclopedia Math. Appl.*, Cambridge University Press, 1993.

[BRS92] J. Bokowski, J. Richter-Gebert, and W. Schindler. On the distribution of order types. *Comput. Geom. Theory Appl.*, 1:127–142, 1992.

[BEK84] J. Bokowski, G. Ewald, and P. Kleinschmidt. On combinatorial and affine automorphisms of polytopes. *Israel J. Math.*, 47:123–130, 1984.

[BS89] J. Bokowski and B. Sturmfels. *Computational Synthetic Geometry.* Volume 1355 of *Lecture Notes in Math.*, Springer-Verlag, Berlin/Heidelberg, 1989.

[BF34] T. Bonnesen and W. Fenchel. *Theorie der konvexen Körper.* Volume 3 of *Ergeb. Math. Grenzgeb.*, Springer-Verlag, Berlin, 1934/1974. Translation: *Theory of Convex Bodies,* BCS Associates Pub., Moscow, Idaho, 1987.

[BH95] K. Böröczky, Jr. and M. Henk. Random projections of regular polytopes. Preprint, TU Berlin, 1995.

[Chr96] T. Christof. PORTA – A Polyhedron Representation Transformation Algorithm. [Available at `ftp://ftp.zib-berlin.de/pub/mathprog/polyth`.]

[DL97] M. Deza and M. Laurent. *Geometry of Cuts and Metrics.* Volume 15 of *Algorithms Combin.*, Springer-Verlag, Heidelberg, 1997.

[GP85] M. Grötschel and M. Padberg. Polyhedral theory. In E.L. Lawler et al., editors, *The Traveling Salesman Problem*, pages 251–360. Wiley, Chichester/New York, 1985.

[GW93] P.M. Gruber and J.M. Wills, editors. *Handbook of Convex Geometry,* Volumes A and B. North-Holland, Amsterdam, 1993.

[Grü67] B. Grünbaum. *Convex Polytopes.* Interscience, London, 1967; revised edition (V. Klee and P. Kleinschmidt, editors), *Graduate Texts in Math.*, Springer-Verlag, in preparation.

[Had57] H. Hadwiger. *Vorlesungen über Inhalt, Oberfläche und Isoperimetrie.* Springer-Verlag, Berlin, 1957.

[Had79] H. Hadwiger. Gitterpunktanzahl im Simplex und Wills'sche Vermutung. *Math. Ann.*, 239:271–288, 1979.

[Hun95] U. Hund. Every simplicial d-polytope with at most $d+4$ vertices is a quotient of a neighborly polytope. Preprint 463/1995, TU Berlin, 1995.

[KK93] J. Kahn and G. Kalai. A counterexample to Borsuk's conjecture. *Bull. Amer. Math. Soc.*, 29:60–62, 1993.

[Kal89] G. Kalai. The number of faces of centrally-symmetric polytopes (Research Problem). *Graphs Combin.*, 5:389–391, 1989.

[KK95] V. Klee and P. Kleinschmidt. Polyhedral complexes and their relatives. In R. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, pages 875–917. North-Holland, Amsterdam, 1995.

[McM75] P. McMullen. Non-linear angle-sum relations for polyhedral cones and polytopes. *Math. Proc. Comb. Phil. Soc.*, 78:247–261, 1975.

[McM93] P. McMullen. Valuations and dissections. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, Volume B, pages 933–988. North-Holland, Amsterdam, 1993.

[McM94] P. McMullen. Duality, sections and projections of certain euclidean tilings. *Geom. Dedicata*, 49:183–202, 1994.

[OS94] S. Onn and B. Sturmfels. A quantitative Steinitz' theorem. *Beiträge Algebra Geom./ Contrib. Algebra Geom.*, 35:125–129, 1994.

[Ric96] J. Richter-Gebert. *Realization Spaces of Polytopes.* Volume 1643 of *Lecture Notes in Math.*, Springer-Verlag, Berlin/Heidelberg, 1996.
[Available at `http://www.math.tu-berlin.de/~richter`.]

[RZ94] J. Richter-Gebert and G.M. Ziegler. Zonotopal tilings and the Bohne-Dress theorem. In H. Barcelo and G. Kalai, editors, *Jerusalem Combinatorics '93*, pages 211–232. Volume 178 of *Contemp. Math.*, Amer. Math. Soc., Providence, 1994.

[San93] J.R. Sangwine-Yager. Mixed volumes. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, Volume A, pages 43–71. North-Holland, Amsterdam, 1993.

[Sch93] R. Schneider. *Convex Bodies: The Brunn-Minkowski Theory.* Volume 44 of *Encyclopedia Math. Appl.*, Cambridge University Press, 1993.

[Sch94] R. Schneider. Polytopes and the Brunn-Minkowski theory. In T. Bisztriczky, P. McMullen, R. Schneider, and A. Ivić Weiss, editors, *Polytopes: Abstract, Convex and Computational*, volume 440 of NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci., pages 273–299. Kluwer, Dordrecht, 1994.

[She82] I. Shemer. Neighborly polytopes. *Israel J. Math.*, 43:291–314, 1982.

[SR34] E. Steinitz and H. Rademacher. *Vorlesungen über die Theorie der Polyeder.* Springer-Verlag, Berlin, 1934; reprint, Springer-Verlag, Berlin, 1976.

[VS92] A.M. Vershik and P.V. Sporyshev. Asymptotic behavior of the number of faces of random polyhedra and the neighborliness problem. *Selecta Math. Soviet.*, 11:181–201, 1992.

[Zie95] G.M. Ziegler. *Lectures on Polytopes.* Volume 152 of *Graduate Texts in Math.*, Springer-Verlag, New York, 1995.
[*Updates, corrections, etc.* available at http://www.math.tu-berlin.de/~ziegler.]

14 SUBDIVISIONS AND TRIANGULATIONS OF POLYTOPES

Carl W. Lee

INTRODUCTION

Starting from a given finite set of points V, we consider subdivisions of the convex hull of V into polytopes $\{P_1, \ldots, P_m\}$. A subdivision is a triangulation if each P_i is a simplex. We start with definitions and properties, then turn to methods of constructing subdivisions and triangulations, face-counting results, some particular triangulations, and secondary and fiber polytopes. We confine ourselves to triangulations of convex structures for the most part.

14.1 BASIC CONCEPTS

GLOSSARY

Affine span: The affine span of a set V is the smallest affine space, or flat, containing V. It is denoted by $\text{aff}(V)$.

Convex hull: The convex hull of a set V is the smallest convex set containing V. It is denoted by $\text{conv}(V)$.

Polytope: A polytope P is the convex hull of a finite set of points. If it is d-dimensional, its boundary consists of faces of dimension -1 (the empty set), 0 (vertices), 1 (edges), 2, ..., and $d-1$ (facets). Its set of vertices will be denoted by $\text{vert}(P)$.

Subdivision: Suppose V is a finite set of points such that $P = \text{conv}(V)$ is d-dimensional (a d-polytope). A subdivision of V is a finite collection $S = \{P_1, \ldots, P_m\}$ of d-polytopes such that:

- The vertices of each P_i are drawn from V (though it is not required that every point in V be used as a vertex of some P_i);
- P is the union of $P_1, \ldots, P_m$; and
- If $i \neq j$ then $P_i \cap P_j$ is a common (possibly empty) face of the boundaries of P_i and P_j.

In this case we will also say that S is a subdivision of the polytope P.

Trivial subdivision: The trivial subdivision of V is the subdivision $\{\text{conv}(V)\}$.

Simplex: A d-dimensional simplex is a d-polytope with exactly $d+1$ vertices.

Triangulation: A subdivision in which each P_i is a simplex.

Faces: The faces of a subdivision $\{P_1, \ldots, P_m\}$ consist of $P_1, \ldots, P_m$ together with their faces.

EXAMPLES

In Figure 14.1.1, (a) shows a set of points. The collection of three polygons in (b) is not a subdivision of that set since not every pair of polygons meets along a common edge or vertex; (c) shows a subdivision that is not a triangulation; and (d) gives a triangulation.

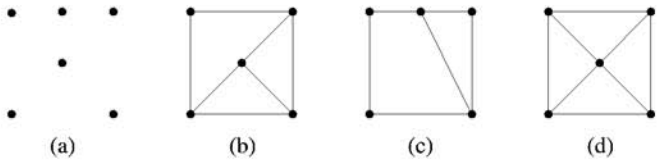


FIGURE 14.1.1
(a) *A set of points.*
(b) *A nonsubdivision.*
(c) *A subdivision.*
(d) *A triangulation.*

14.2 SEQUENTIAL CONSTRUCTION PROCEDURES

The convex hull of a finite set of points $V = \{v_1, \ldots, v_n\}$ can be constructed sequentially by successively constructing $R_1 = \text{conv}(\{v_1\})$, $R_2 = \text{conv}(R_1 \cup \{v_2\})$, $R_3 = \text{conv}(R_2 \cup \{v_3\}), \ldots, R_n = \text{conv}(R_{n-1} \cup \{v_n\})$. With little additional effort, a triangulation of each R_i can also be constructed, resulting finally in a triangulation of V. Another method of constructing a triangulation of V is to begin with the trivial subdivision of V, and then obtain a sequence of refinements. See [Lee91, Zie95].

GLOSSARY

Refinement of a subdivision: Suppose $S = \{P_1, \ldots, P_l\}$ and $T = \{Q_1, \ldots, Q_m\}$ are two subdivisions of V. Then T is a refinement of S if for each j, $1 \leq j \leq m$, there exists i, $1 \leq i \leq l$, such that $Q_j \subseteq P_i$. In this case we will write $T \leq S$.

Visible facet: Suppose P is a d-polytope in $\mathbb{R}^d$, F is a facet of P, and v is a point in $\mathbb{R}^d$. There is a unique hyperplane H (affine set of dimension $d-1$) containing F. The polytope P is contained in exactly one of the closed halfspaces determined by H. If v is contained in the opposite open halfspace, then F is said to be visible from v. If P is a k-polytope in $\mathbb{R}^d$ with $k < d$ and $v \in \text{aff}(P)$, then the above definition is modified in the obvious way so that everything is considered relative to the ambient space $\text{aff}(P)$.

Placing a vertex: Suppose $S = \{P_1, \ldots, P_m\}$ is a subdivision of V and $v \not\in V$. The subdivision T of $V \cup \{v\}$ that results from placing v is obtained as follows:

- If $v \not\in \text{aff}(V)$, then for each $P_i \in S$, include $\text{conv}(P_i \cup \{v\})$ in T.
- If $v \in \text{aff}(V)$, then for each $P_i \in S$, $P_i \in T$; and if F is a facet of P_i that is contained in a facet of $\text{conv}(V)$ visible from v, then $\text{conv}(F \cup \{v\}) \in T$.
- Note: if $v \in \text{conv}(V)$, then $S = T$.

Pulling a vertex: Suppose $S = \{P_1, \ldots, P_m\}$ is a subdivision of V and $v \in V$. The result of pulling v is the subdivision T of V obtained by modifying each $P_i \in S$ as follows:

- If $v \notin P_i$, then $P_i \in T$.
- If $v \in P_i$, then for every facet F of P_i not containing v, $\text{conv}(F \cup \{v\}) \in T$.

Note that T is a refinement of S.

Pushing a vertex: Suppose $S = \{P_1, \ldots, P_m\}$ is a subdivision of V (where $\dim(\text{conv}(V)) = d$) and $v \in V$. The result of pushing v is the subdivision T of V obtained by modifying each $P_i \in S$ as follows:

- If $v \notin P_i$, then $P_i \in T$.
- If $v \in P_i$ and $\text{conv}(\text{vert}(P_i) \setminus \{v\})$ is $(d-1)$-dimensional (i.e., P_i is a pyramid with apex v), then $P_i \in T$.
- If $v \in P_i$ and $P_i' = \text{conv}(\text{vert}(P_i) \setminus \{v\})$ is d-dimensional, then $P_i' \in T$. Also, if F is any facet of P_i' that is visible from v, then $\text{conv}(F \cup \{v\}) \in T$.

Note that T is a refinement of S.

Lexicographic subdivisions: If T is any subdivision of V constructed by starting with the trivial subdivision of V and then pushing and/or pulling some/all of the points in V in some order, then T is a lexicographic subdivision.

Diameter of a subdivision: Suppose $\{P_1, \ldots, P_m\}$ is a subdivision. Polytopes $P_i \neq P_j$ are ***adjacent*** if they share a common facet. A sequence $P_{i_0}, \ldots, P_{i_k}$ is a ***path*** if P_{i_j} and $P_{i_{j-1}}$ are adjacent for each $1 \leq j \leq k$. The ***length*** of such a path is k. The ***distance*** between polytopes P_i and P_j is the length of the shortest path connecting them. The diameter of the subdivision is the maximum distance occurring between pairs of polytopes P_i, P_j.

MAIN RESULTS

1. If the points of V are ordered $\{v_1, \ldots, v_n\}$ and T is the subdivision obtained by placing the points of V in that order, then
   (a) T is a triangulation of V.
   (b) The same triangulation is obtained by starting with the trivial subdivision of V and pushing the points of V in the opposite order $v_n, \ldots, v_1$.
   (c) The diameter of T does not exceed $2(n-d-1)$, where $d = \dim(\text{conv}(V))$ [Lee91].
2. If S is any subdivision of $V = \{v_1, \ldots, v_n\}$, then S can be refined to a triangulation by sequentially pushing and/or pulling all the vertices in some order.
3. For any specified point $v_k \in V = \{v_1, \ldots, v_n\}$, there is a triangulation of V in which every simplex of maximum dimension contains v_k as a vertex—begin with the trivial subdivision $S = \{\text{conv}(V)\}$, pull v_k first, then pull the remaining points in any order.
4. For any specified simplex F with vertices in $V = \{v_1, \ldots, v_n\}$, there is a triangulation of V in which F is a face—first place the vertices of F, then place the remaining vertices in any order.

5. If $\dim(\text{conv}(V)) = d$ and $\text{card}(V) \leq d+3$, then every triangulation of V can be obtained by placing (respectively, pushing) the points of V in some order [Lee91].

6. If V is the set of vertices of a convex n-gon in $\mathbb{R}^2$, then every triangulation of V can be obtained by placing (respectively, pushing) the points of V in some order.

7. Suppose $V = \{v_1, \ldots, v_n\}$ is the set of vertices of some d-polytope P. For a face F of P, define $v(F) = v_k$, where $k = \min\{i \mid v_i \in F\}$. A ***full flag*** of P is a chain C of faces $F_0 \subset F_1 \subset F_2 \subset \cdots \subset F_{d-1} \subset F_d = P$ such that $\dim(F_i) = i$, $0 \leq i \leq d$, and $v(F_i) \neq v(F_{i-1})$, $1 \leq i \leq d$. For a full flag C, write $v(C) = \{v(F_0), \ldots, v(F_d)\}$. Then the simplices of the triangulation of P determined by pulling the vertices of P in the order $v_1, \ldots, v_n$ are $\{\text{conv}(v(C)) \mid C \text{ is a full flag of } P\}$.

EXAMPLES

Figure 14.2.1 gives three triangulations of a set of seven points that can be obtained from the trivial subdivision by pulling and pushing [Lee91]. The triangulation in (a) is obtained by pulling point 1, but cannot be obtained by pushing alone. The triangulation in (b) is obtained by pushing the points in the indicated order, or placing them in the opposite order, but cannot be obtained by pulling points alone. The lexicographic triangulation in (c) is obtained by pushing point 1 and then pulling point 2, but cannot be obtained by pulling points alone or by pushing points alone.

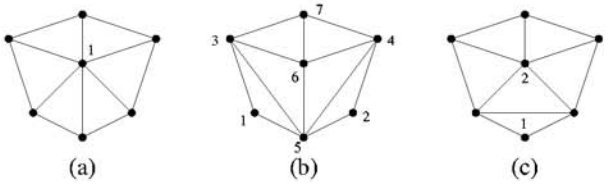


FIGURE 14.2.1
(a) *A pulling triangulation.*
(b) *A pushing triangulation.*
(c) *A lexicographic triangulation.*

14.3 REGULAR TRIANGULATIONS AND SUBDIVISIONS

GLOSSARY

Regular subdivision: Any convex hull algorithm for points in $\mathbb{R}^{d+1}$ can be used to compute subdivisions of sets of points $V = \{v_1, \ldots, v_n\}$ in $\mathbb{R}^d$ (see Chapter 19 of this Handbook). Such subdivisions are called regular and are obtained in the following way:

(i) Regard V as sitting naturally in $(\mathbb{R}^d, 0)$.

(ii) Choose arbitrary real numbers $\alpha_1, \ldots, \alpha_n$.

(iii) Determine $Q = \text{conv}(\{(v_1, \alpha_1), \ldots, (v_n, \alpha_n)\})$.

(iv) Project the lower facets of Q onto $(\mathbb{R}^d, 0)$.

Here, a lower facet is a facet of Q that is visible from the point $(0, -\alpha)$ for α sufficiently large. See [GKZ94, Lee91, Zie95]. Some algorithmic aspects of computing regular triangulations can be found in [ES96].

Weakly regular subdivision: A subdivision S of a set V is weakly regular if there exists a set V' having a regular subdivision S' such that (V', S') is combinatorially isomorphic to (V, S). That is, there is a one-to-one correspondence between the points of V and the points of V' such that for every subset $F \subseteq V$ and corresponding subset $F' \subseteq V'$, $\text{conv}(F)$ is a face of S if and only if $\text{conv}(F')$ is a face of S'.

Polytopal complex: A polytopal complex is a finite, nonempty collection S of polytopes in $\mathbb{R}^d$ that contains all the faces of its polytopes, and such that the intersection of any two of its polytopes is a common face of each of them. The dimension of S, $\dim(S)$, is the largest dimension of a polytope in S, and S is ***pure*** if every polytope in S is contained in one of dimension $\dim(S)$ [Zie95]. (Thus every subdivision is a pure polytopal complex.)

Shellable: A pure polytopal complex S is shellable if it is 0-dimensional (i.e., a finite set of points) or else $\dim(S) = k > 0$ and S has a ***shelling***, i.e., an ordering of its maximal faces $P_1, \ldots, P_m$ such that for $2 \leq j \leq m$ the intersection of P_j with $P_1 \cup \cdots \cup P_{j-1}$ is nonempty and is the beginning segment of a shelling of the $(k-1)$-dimensional boundary complex of P_j [Zie95].

MAIN RESULTS

1. All regular subdivisions are shellable. On the other hand, there exist nonshellable subdivisions, starting in dimension 3 (see [Zie95]).

2. All lexicographic triangulations are regular. In particular, if $v_1, \ldots, v_n$ are pushed/pulled in that order, then the corresponding triangulation is obtained by choosing $|\alpha_1| \gg |\alpha_2| \gg \cdots \gg |\alpha_n| \gg 0$, where $\alpha_i > 0$ if v_i is pushed and $\alpha_i < 0$ if v_i is pulled [Lee91].

3. If $\text{card}(V) = \dim(\text{conv}(V)) + 2$, then there are exactly two triangulations of V, and both are regular.

4. If $\text{card}(V) = \dim(\text{conv}(V)) + 3$, then all subdivisions of V are regular [Lee91].

5. If V is the set of vertices of a convex n-gon in $\mathbb{R}^2$, then all subdivisions of V are regular.

6. If $V \subset \mathbb{R}^2$, then all subdivisions of V are weakly regular as a consequence of Steinitz's Theorem. However, there exists a set V of 6 points in $\mathbb{R}^2$ having a nonregular triangulation [Lee91] (see Figure 14.3.2(b)).

7. There exists a set V of 7 points that are the vertices of a 3-polytope having a nonregular triangulation that is not even weakly regular [Lee91] (see Figure 14.3.3(b)).

8. If V is the vertex set of $C_{4n-4}(4n)$, the cyclic polytope of dimension $4n-4$ with $4n$ vertices, then V has at least 2^n triangulations, of which only $O(n^4)$ are regular [dHSS96]. (See Chapter 13 of this Handbook for the definition of the cyclic polytope.)

9. If $\alpha_i = \|v_i\|^2$, then the resulting subdivision is the Delaunay subdivision. If $\alpha_i = -\|v_i\|^2$, then the resulting subdivision is the "farthest site" Delaunay subdivision. (See Chapters 20 and 22 of this Handbook.)

10. Given a subdivision of V, one can test its regularity by using linear programming to check the existence of appropriate α_i, $1 \leq i \leq n$. On the other hand, checking weak regularity is quite hard, perhaps as difficult as determining solutions to systems of real polynomial inequalities (see comments on the Universality Theorem in Chapter 13 and in [Zie95]).

EXAMPLES

Figure 14.3.1 shows the two triangulations (both regular) of the vertices of a 3-dimensional bipyramid over a triangle. In (a) there are two tetrahedra in the triangulation, sharing a common internal triangle; in (b) there are three, sharing a common internal edge.

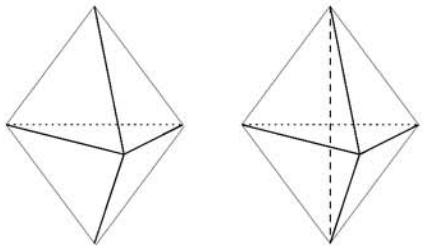


FIGURE 14.3.1
The two triangulations of a set of 5 *points in* $\mathbb{R}^3$.

Figure 14.3.2 shows triangulations of two different sets of 6 points in $\mathbb{R}^2$. The first triangulation is regular, the second is not. But by virtue of the first triangulation, the second is weakly regular.

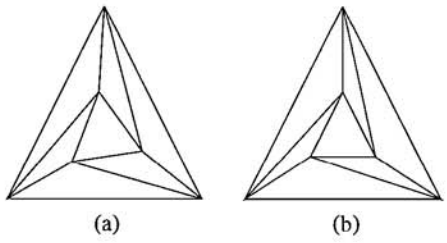


FIGURE 14.3.2
A regular and a nonregular (but weakly regular) triangulation.

Figure 14.3.3 shows two 3-polytopes, each with 7 vertices. The polytope in (a) is a "capped triangular prism" and its vertex set admits two nonregular triangulations. Denoting the simplices by their vertex sets, these are: $\{1257, 1457, 1236, 1267, 1345,$

1346, 1467} and {1245, 1247, 1237, 1367, 1356, 1456, 1467}. Both triangulations are, however, weakly regular. The polytope in (b) is obtained from the capped triangular prism by rotating the top triangle by a small amount. Its vertex set has one nonregular triangulation, which is not even weakly regular: {1245, 1247, 1237, 1367, 1356, 1456, 1467, 2457, 2367, 2345}. See [Lee91].

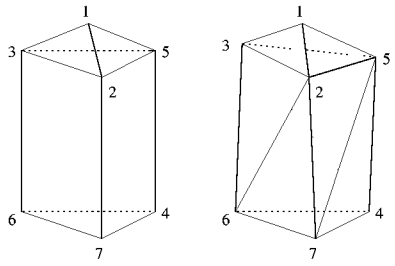


FIGURE 14.3.3
Two polytopes with nonregular triangulations.

14.3.1 TRIANGULATING REGIONS BETWEEN POLYTOPES

Suppose P and Q are two d-polytopes in $\mathbb{R}^d$ with disjoint vertex sets V and W, respectively, and Q is contained in P. One can triangulate the region inside of P and outside of Q by the following procedure [GP88]:

1. Construct the regular subdivision of $V \cup W$ by setting $\alpha_i = 1$ for the $v_i \in V$ and $\alpha_i = 0$ for $v_i \in W$.
2. Refine this subdivision to a triangulation by pushing and/or pulling each point in $V \cup W$.
3. Ignore the portion of the triangulation within Q.

Now suppose P and Q are two d-polytopes in $\mathbb{R}^d$ with disjoint vertex sets V and W, respectively, and that there is a hyperplane H for which P and Q are contained in opposite open halfspaces. One can triangulate the region in $\text{conv}\,(P \cup Q)$ that is exterior to P and Q by the following procedure [GP88]:

1. Construct the regular subdivision of $V \cup W$ by setting α equal to the distance of v_i to H for each $v_i \in V \cup W$. For example, if $H = \{x \mid a \cdot x = \beta\}$, then α_i can be taken to equal $|a \cdot v_i - \beta|$. (It would also suffice to use these values of α_i for $v_i \in V$ and to set $\alpha_i = 0$ for $v_i \in W$.)
2. Refine this subdivision to a triangulation by pushing and/or pulling each point in $V \cup W$.
3. Ignore the portion of the triangulation within P or Q.

14.4 SUBDIVISIONS, TRIANGULATIONS, AND FACE VECTORS

Suppose $S = \{P_1, \ldots, P_m\}$ is a subdivision of V with $\dim(\text{conv}\,(V)) = d$. In this section we examine some of the properties of its face numbers. See [Bay94, BL93].

GLOSSARY

Boundary: Suppose S is as above. The boundary complex ∂S of S is the set of those faces of S given by $\{F \in S \mid F \subseteq G$ for some face G of dimension $d-1$ contained in exactly one $P_i\}$. In particular, the empty set is a member of ∂S.

Interior: Suppose S is as above. The interior $\operatorname{int} S$ is the set of those faces of S that are not in the boundary.

f-vector: Suppose S is as above. Let $f_j(S)$ denote the number of j-dimensional faces of S, $-1 \leq j \leq d$. Note that $f_{-1}(S) = 1$ since the empty set is the unique face of S of dimension -1. The f-vector of S is $f(S) = (f_0(S), \ldots, f_d(S))$. In an analogous way we define $f(\partial S)$ and $f(\operatorname{int} S)$. Note that $f_{-1}(\partial S) = 1$ and $f_{-1}(\operatorname{int} S) = 0$.

Simplicial polytope: A simplicial polytope is one for which every facet (and hence every face) is a simplex.

14.4.1 h-VECTORS and g-VECTORS

Suppose S is any polytopal complex of dimension d. For example, S might be the boundary complex of a polytope of dimension $d+1$ or a subdivision of a finite set V such that $\operatorname{conv}(V)$ has dimension d.

We define the h-vector $h(S) = (h_0(S), \ldots, h_{d+1}(S))$ with generating function $h(S,x) = \sum_{i=0}^{d+1} h_i x^{d+1-i}$, and the g-vector $g(S) = (g_0(S), \ldots, g_{\lfloor (d+1)/2 \rfloor}(S))$ with generating function $g(S,x) = \sum_{i=0}^{\lfloor (d+1)/2 \rfloor} g_i x^i$ in the following recursive way:

1. $g_0(S) = h_0(S)$.
2. $g_i(S) = h_i(S) - h_{i-1}(S)$, $1 \leq i \leq \lfloor (d+1)/2 \rfloor$.
3. $g(\emptyset, x) = h(\emptyset, x) = 1$.
4. $h(S,x) = \displaystyle\sum_{G \text{ face of } S} g(\partial G, x)(t-1)^{d - \dim(G)}$.

For more information on f-vectors, g-vectors, and h-vectors, refer to Chapter 15. The formulas are simpler when all of the faces of S are simplices.

MAIN RESULTS

1. Assume that T is a triangulation of a d-polytope [BL93].
   (a) The number of d-simplices in T equals the sum of the components of the h-vector.
   (b) The h-vector is nonnegative.
   (c) The h-vectors of T, ∂T, and $\operatorname{int} T$ are related in the following ways:

$$h_i(T) - h_{d+1-i}(T) = h_i(\partial T) - h_{i-1}(\partial T),\ 0 \leq i \leq d+1.$$

$$h_i(T) = h_{d+1-i}(\operatorname{int} T),\ 0 \leq i \leq d+1.$$

In particular, the h-vectors and the f-vectors of ∂T and $\operatorname{int} T$ are completely determined by the h-vector (and hence the f-vector) of T.

(d) Assume further that T is shellable and that $P_1, \ldots, P_m$ is a shelling order of the d-dimensional simplices in T. In particular, each P_j meets $\bigcup_{i=1}^{j-1} P_i$ in some positive number s_j of facets of P_j, $2 \leq j \leq m$. Define also $s_1 = 0$. Then $h_i(T)$ equals $\operatorname{card}\{j \mid s_j = i\}$, $0 \leq i \leq d+1$.

2. If S is the trivial subdivision of a convex d-polytope P consisting of P itself, then

$$h_i(S) = \begin{cases} h_i(\partial P) - h_{i-1}(\partial P), & 1 \leq i \leq \lfloor d/2 \rfloor, \\ 0, & \lfloor d/2 \rfloor < i \leq d. \end{cases}$$

See [Bay94].

3. Suppose V is a finite set of points with rational coordinates, S is a subdivision of V, and $P = \operatorname{conv}(V)$. Then for all i, $h_i(S) \geq h_i(P)$ and $h_i(\partial S) \geq h_i(\partial P)$. Further, if P is simplicial and S is a triangulation, the result holds even without the rationality assumption. In either case, $f_d(S) \geq h_{\lfloor d/2 \rfloor}(\partial S) \geq h_{\lfloor d/2 \rfloor}(\partial P)$ [Bay94, Sta92].

14.4.2 SHALLOW TRIANGULATIONS

The concept of shallow triangulation is motivated by an attempt to understand the case of equality in the last result mentioned above. See [Bay94, BL93].

GLOSSARY

The following definitions concern triangulations T of a finite set V of vertices of a convex d-polytope P.

Carrier: If F is a face of T, the carrier $C(F)$ of F is the smallest face of P containing F.

Shallow: If $\dim(C(F)) \leq 2\dim(F)$ for every face F of T, then T is a shallow triangulation.

Weakly neighborly: If all triangulations of V are shallow, then P is weakly neighborly.

Equidecomposable: If all triangulations of V have the same f-vector, then P is equidecomposable.

Stacked: If P has a triangulation in which there are no interior faces of dimension less than $d-1$, then P is stacked.

k-stacked: If P is a simplicial d-polytope that has a triangulation in which there are no interior faces of dimension less than $d-k$, then P is k-stacked. In particular, a simplicial polytope is stacked if and only if it is 1-stacked.

MAIN RESULTS

1. A polytope P is weakly neighborly if and only if every set of $k+1$ vertices is contained in a face of dimension at most $2k$ for all k [Bay94].

2. If P is weakly neighborly, then P is equidecomposable.

3. If T is a shallow triangulation of a polytope P, then $h(T) = h(P)$ and $h(\partial T) = h(\partial P)$ [Bay94].

4. If T is a triangulation of a polytope P with rational vertices and $h(T) = h(P)$, then T is shallow. Hence, if P is a rational polytope and $h(T) = h(P)$ for all triangulations T of P, then P is weakly neighborly [Bay94].

5. If P is a simplicial d-polytope, then it has a shallow triangulation if and only if it is k-stacked for some $1 \leq k \leq d/2$. In this case there is exactly one triangulation T of P having no interior faces of dimension less than $d - k$ (and this triangulation is the unique shallow one) [Bay94].

6. Suppose P is a d-polytope where $d > 3$. Then P is 1-stacked if and only if $g_2(\partial P) = 0$. See [BL93].

7. Suppose P is a simplicial d-polytope such that $g_k(\partial P) = 0$ for some k with $3 \leq k \leq \lfloor d/2 \rfloor$. Then there is another simplicial d-polytope that has the same f-vector and is $(k-1)$-stacked. It is an open problem whether P itself is always $(k-1)$-stacked under this hypothesis; this is known to be true if $f_0(P) \leq d+3$ or $k < f_0(P)/(f_0(P) - d)$. (See [BL93], but note that there are places where "k" appears instead of the correct "$k - 1$.")

Some classes of weakly neighborly polytopes are given below [Bay94]:

- In dimension less than 3, all polytopes are weakly neighborly.
- In dimension 3, the only weakly neighborly polytopes are pyramids (over polygons) and the triangular prism.
- The product of two simplices of any dimensions is weakly neighborly. (See Section 14.5.1 for the definition of product.)
- The only simplicial weakly neighborly polytopes are simplices and even-dimensional neighborly polytopes (those for which every subset of $d/2$ vertices determines a face of the polytope).
- Lawrence polytopes are weakly neighborly. (See [Bay94, Zie95] for the definition of Lawrence polytopes.)
- Pyramids over weakly neighborly polytopes are weakly neighborly.
- Subpolytopes of weakly neighborly polytopes are weakly neighborly.

14.4.3 RELATIONSHIPS TO COUNTING LATTICE POINTS

Triangulations of polytopes can be used to enumerate lattice points in polytopes. See [BL93].

GLOSSARY

Integral: A polytope is integral if every vertex has integer coordinates.

i (P, n)***:*** For an integral polytope P and a nonnegative integer n, $i(P, n)$ is the number of points $x \in P$ for which nx has integer coordinates. Equivalently, it is the number of integer points in nP.

Compressed ordering: An ordering of the vertices of an integral polytope P is compressed if every d-dimensional simplex of the triangulation obtained by pulling the vertices of P in that order has volume $1/d!$. P itself is ***compressed*** if every ordering of its vertices is compressed. (For example, the standard d-dimensional unit cube is compressed.)

MAIN RESULTS

1. $i(P, n)$ is a polynomial in n of degree d, called the Ehrhart polynomial of P (see Chapter 7).
2. For integral d-polytope P write $J(P, t) = 1 + \sum_{n=1}^{\infty} i(P, n)t^n$. Then $J(P, t) = W(P, t)/(1 - t)^{d+1}$, where $W(P, t)$ is a polynomial of degree at most d with nonnegative integer coefficients.
3. If P is an integral d-polytope with compressed order σ, then

$$i(P, n) = \sum_{i=0}^{d} \binom{n-1}{i} f_i(T),$$

and $W(P, t) = h_0(T) + h_1(T)t + \cdots + h_d(T)t^d$, where T is the pulling triangulation induced by σ.

4. If P is a compressed integral d-polytope and σ is an ordering of its vertices, then the f-vector of the triangulation induced by σ depends only on P, not on σ.

For example, if P is the standard unit 3-cube, then any ordering σ produces a compressed triangulation T with h-vector $h(T) = (1, 4, 1, 0, 0)$. Thus $J(P, t) = (1 + 4t + t^2)/(1 - t)^4 = (1 + 4t + t^2)(1 + 4t + 10t^2 + 20t^3 + 35t^4 + \cdots) = 1 + 8t + 27t^2 + 64t^3 + 125t^4 + \cdots$.

14.5 SOME PARTICULAR TRIANGULATIONS

We gather together some information on some particular triangulations, including triangulations of the product of two simplices, the d-dimensional cube, the convex n-gon, and complete barycentric subdivisions.

14.5.1 PRODUCT OF TWO SIMPLICES

Consider the $(k+l)$-polytope $P = \Delta_k \times \Delta_l$, the product of a k-dimensional simplex Δ_k and an l-dimensional simplex Δ_l. We consider triangulations of P using the points in its vertex set V. See [BCS88, deL96, GKZ94, Hai91].

GLOSSARY

Product: If P is a subset of $\mathbb{R}^k$ and Q is a subset of $\mathbb{R}^l$, then the product of P and Q is the subset of $\mathbb{R}^{k+l}$ given by $\{(v, w) \mid v \in P,\ w \in Q\}$.

MAIN RESULTS

1. As mentioned before, $P = \Delta_k \times \Delta_l$ is weakly neighborly, and so every triangulation has the same f-vector and h-vector. In particular, if T is a triangulation of P, then $f_{k+l}(T) = (k+l)!/(k!l!)$, and $h_i(T) = \binom{k}{i}\binom{l}{i}$ for $0 \le i \le k+l$ (with $h_i(t)$ taken to be zero if $i > \min\{k, l\}$) [BCS88].

2. Given a triangulation $\{P_1, \ldots, P_s\}$ of a k-polytope P and a triangulation $\{Q_1, \ldots, Q_t\}$ of an l-polytope Q, then there is a triangulation of $P \times Q$ using $s \cdot t \cdot (k+l)!/(k!l!)$ simplices of dimension $k+l$. To see this, observe that $\{P_i \times Q_j \mid 1 \le i \le s,\ 1 \le j \le t\}$ is a subdivision of $P \times Q$. Now refine this subdivision to a triangulation by, for example, pulling the vertices of $P \times Q$. Each $P_i \times Q_j$ will thereby be refined into $(k+l)!/(k!l!)$ simplices [Hai91].

3. All triangulations of $\Delta_2 \times \Delta_3$ and $\Delta_2 \times \Delta_4$ are regular. On the other hand, if $k, l \ge 3$, then there exist nonregular triangulations of $\Delta_k \times \Delta_l$ [deL96].

To describe one triangulation of $\Delta_k \times \Delta_l$ explicitly [BCS88, GKZ94], assume that Δ_k has vertex set $\{v_0, \ldots, v_k\}$ and that Δ_l has vertex set $\{w_0, \ldots, w_l\}$. Then $P = \Delta_k \times \Delta_l$ has vertex set $\{(v_i, w_j) \mid 0 \le i \le k,\ 0 \le j \le l\}$.

Consider paths from the vertex (v_0, w_0) to the vertex (v_k, w_l) in which each step involves increasing either the index of v or the index of w by one. Each such path selects a subset of $k+l+1$ vertices of P, which determines a $(k+l)$-dimensional simplex. The collection of simplices associated with all such paths constitutes a triangulation of P. This is the same triangulation of P as the one obtained by starting with the trivial subdivision of P and pulling the vertices in the order

$$\begin{gathered}(v_0, w_0), (v_0, w_1), \ldots, (v_0, w_l),\\ (v_1, w_0), (v_1, w_1), \ldots, (v_1, w_l),\\ \vdots\\ (v_k, w_0), (v_k, w_1), \ldots, (v_k, w_l).\end{gathered}$$

Figure 14.5.1 shows this triangulation for $\Delta_2 \times \Delta_1$, a prism. The label ij on a vertex is an abbreviation for (v_i, w_j).

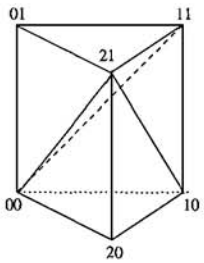


FIGURE 14.5.1
A triangulation of $\Delta_2 \times \Delta_1$.

14.5.2 d-CUBES

Here we consider triangulations of a d-dimensional cube using only the set V of its vertices. See [Hai91].

GLOSSARY

d-cube: The unit d-dimensional cube I^d is the d-fold product of the unit interval $I = [0, 1]$ with itself.

Index: A vertex of the d-dimensional cube is a point of the form $(a_1, \ldots, a_d) \in \{0, 1\}^d$. Define the index of the vertex to be $\sum_{i=0}^{d-1} a_{i+1} 2^i$.

Size: The size of a triangulation T is the number $f_d(T)$ of d-simplices in T.

$\varphi(d)$: The size of the smallest triangulation of I^d. That is, $\varphi(d) = \min\{f_d(T) \mid T$ is a triangulation of $I^d\}$.

MAIN RESULTS

1. The maximum size of a triangulation of I^d is $d!$ (since the minimum volume of a d-simplex using the vertices of I^d is $1/d!$), and this is achievable for every d by pulling the vertices in any order.

2. $\varphi(d) \geq 2^d(d+1)^{-(d+1)/2}d!$. This bound is derived by observing that I^d can be inscribed in a sphere of diameter $\sqrt{d}$, and that the maximum volume of a simplex contained in this sphere is $(d+1)^{(d+1)/2}/(2^d d!)$ (the volume of a regular simplex) [Hai91].

3. There are precisely 74 triangulations of the 3-cube, and these fall into 6 classes of combinatorially different types [Big91, deL95]. All are regular. On the other hand, if $d \geq 4$, then not all triangulations of the d-cube are regular [deL96].

4. If I^d can be triangulated into $T(d)$ simplices, then I^{kd} can be triangulated into $[(kd)!/(d!)^k]T(d)^k = \rho^{kd}(kd)!$ simplices, where $\rho = (T(d)/d!)^{1/d}$. One measure of the efficiency of a triangulation is $\rho = (T(d)/d!)^{1/d}$. This result shows that any value of ρ achievable for one triangulation is achievable asymptotically. The smallest value of ρ obtainable from triangulations to date is $\rho = (13,248/40,320)^{1/8} \approx 0.870$ [Hai91].

Table 14.5.1 lists the known values of $\varphi(d)$ [Hug93].

TABLE 14.5.1 Minimal triangulations of d-cubes.

d	1	2	3	4	5
$\varphi(d)$	1	2	5	16	67

It is also known that the smallest size of a triangulation of I^6 that slices off alternate vertices of I^6 is 324 [Hug93].

Figure 14.5.2 shows a triangulation of the 3-cube of size 5.

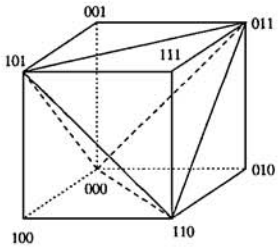


FIGURE 14.5.2
A minimum size triangulation of the 3-cube.

SOME SPECIFIC TRIANGULATIONS OF I^d

Pushing vertices: Start from the trivial subdivision of I^d and push the vertices in order of decreasing index. The resulting triangulation will have $d!$ simplices [Big91].

Pulling vertices: Start from the trivial subdivision of I^d and pull the vertices in order of increasing index to obtain the triangulation T. Pulling the vertices in any order yields a triangulation with $d!$ simplices, so that $f_d(T) = d!$. $h_d(T) = h_{d+1}(T) = 0$ and $h_i(T) = A(d,i)$, $0 \le i \le d-1$, where $A(d,i)$ is the Eulerian number (it equals the number of permutations of $\{1,\ldots,d\}$ having exactly i descents). There is a one-to-one correspondence between the simplices in T and the permutations of $\{1,\ldots,d\}$, given in the following way: For a given permutation σ, the corresponding simplex has vertices $(0,\ldots,0) + e_{\sigma(1)} + e_{\sigma(2)} + \cdots + e_{\sigma(k)}$, $0 \le k \le d$, where e_i denotes the standard ith unit vector. This is also known as Kuhn's triangulation [Big91, Tod76].

Sallee's corner slicing triangulation: Assume $d \ge 3$. For each vertex with an odd number of coordinates equaling 1, construct the simplex consisting of this vertex and its d neighbors (those joined to this vertex by an edge). These simplices, together with the central polytope remaining when these simplices are removed, constitute a subdivision of I^d. Refine this subdivision to a triangulation by pulling the vertices in order of increasing index. This triangulation has size $O(d!)$ [Hai91, Sal82].

Sallee's middle cut triangulation: Assume $d \ge 2$. Slice the cube into two polytopes by the hyperplane $x_1 + \cdots + x_d = \lfloor d/2 \rfloor$. Refine this subdivision to a triangulation by pulling the vertices in order of increasing index. This triangulation has size $O(d!/d^2)$ [Sal84].

Haiman's triangulation: This triangulation method, which bootstraps a triangulation of I^d to a triangulation of I^{kd} as described in Section 14.5.1, Main Result 2, has size $O(\rho^d d!)$, where $\rho < 1$ [Hai91].

EXAMPLES

Figure 14.5.3 shows two triangulations of the 3-cube: (a) the one resulting from pulling the vertices in order of increasing index, and (b) the one resulting from pushing the vertices in order of decreasing index.

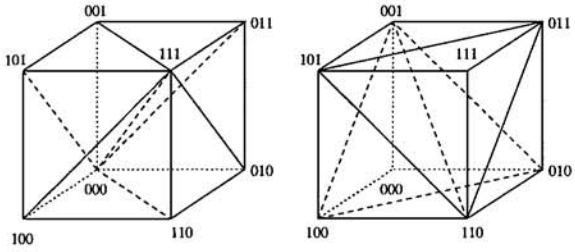


FIGURE 14.5.3
(a) *The pulling triangulation of the* 3*-cube.*
(b) *The pushing triangulation of the* 3*-cube.*

14.5.3 CONVEX n-GONS

There is no difficulty in finding subdivisions and triangulations of a convex n-gon using its set V of vertices. All subdivisions are regular, and all triangulations are constructible by pushing (or placing). Any subdivision is determined by a collection of mutually noncrossing internal diagonals. The set of all triangulations of the n-gon is isomorphic to many other combinatorial structures, including the set of all ways to parenthesize a string of $n-1$ symbols and the set of all rooted binary trees with $n-2$ nodes. See [Lee89, Zie95].

MAIN RESULTS

1. There are $\frac{1}{n-1}\binom{n-3}{j}\binom{n+j-1}{j+1}$ subdivisions of a convex n-gon having exactly j diagonals, $0 \le j \le n-3$. In particular, the number of triangulations is the ***Catalan number*** $\frac{1}{n-1}\binom{2n-4}{n-2}$.
2. Two triangulations are ***adjacent*** if and only if they share all but one diagonal. The ***distance*** between two triangulations T and T' is the length k of the shortest path $T = T_0, T_1, T_2, \ldots, T_k = T'$ of triangulations in which T_i and T_{i-1} are adjacent for all $1 \le i \le k$. The distance between two triangulations of a convex n-gon does not exceed $2n-6$ [Luc89]. This bound is achievable for infinitely many values of n [STT88].
3. The set of all triangulations of a convex n-gon is connected by a Hamiltonian cycle—a closed path $T_0, T_1, T_2, \ldots, T_m = T_0$ containing each triangulation exactly once (except for T_0, which starts and ends the path), in which T_i and T_{i-1} are adjacent for all $1 \le i \le m$ [Luc87].

14.5.4 COMPLETE BARYCENTRIC SUBDIVISIONS

For a given d-polytope P, let V be the collection of the centroids of the nonempty faces. Give the centroid of each k-dimensional face the label k, $0 \le k \le d$. Note that points labeled 0 are the vertices of P. Triangulate P by pulling the points of V in order of nonincreasing label. The resulting triangulation is the ***complete barycentric subdivision*** of P. The procedure can be extended in the obvious way to be applied to any polytopal complex. See [Bay88].

Figure 14.5.4 shows the complete barycentric subdivision of a 3-cube, a triangulation of size 48—there are eight pyramids into the center of the cube from each of the six original facets.

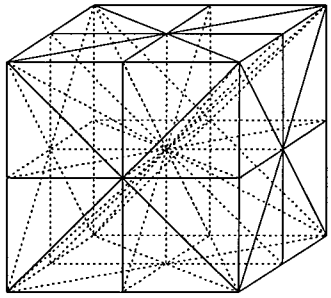


FIGURE 14.5.4
The complete barycentric subdivision of a 3-cube.

MAIN RESULTS

1. For every polytope P, there is a ***dual polytope*** (or polar polytope: see Chapter 13) P^* of the same dimension, whose face lattice is anti-isomorphic to that of P. The complete barycentric subdivisions T and T^* of P and P^* are combinatorially isomorphic. That is to say, there is a bijection between the vertices of T and of T^* such that a subset of vertices of T determines a simplex in T precisely when the corresponding subset of vertices of T^* determines a simplex in T^*.

2. If T is the complete barycentric subdivision of a polytope P, then the combinatorial structure of the face lattice of P (up to lattice reversal by the previous result) can be recovered from the combinatorial structure of T, even if one is not given the specific geometric realization or the labels of the points [Bay88].

3. Suppose T is the complete barycentric subdivision of a d-dimensional simplex. Then $f_d(T) = d!$, $h_{d+1}(T) = 0$, and $h_i(T) = A(d+1, i)$, $0 \leq i \leq d$. These are the Eulerian numbers encountered in Kuhn's triangulation of I^{d+1}. In fact, Kuhn's triangulation is combinatorially isomorphic to the join of T to a new point (make a pyramid with this new point over every d-simplex in T) [Big91].

4. If T is the complete barycentric subdivision of I^d, then $f_d(T) = d!2^d$. Also, $h_{d+1}(T) = 0$, and $h_i(T)$ equals the number of signed permutations of $\{1, \ldots, d\}$ with exactly i descents [Bre94].

14.6 SECONDARY AND FIBER POLYTOPES

This section concerns itself with the structure of the collection of all regular subdivisions of a given finite set of points $V = \{v_1, \ldots, v_n\} \subset \mathbb{R}^d$. See [GKZ94, Lee91, Zie95]. Assume that $\dim(\text{conv}(V)) = d$.

GLOSSARY

z-vector: Suppose T is a regular triangulation of V. Define the z-vector $z(T) = (z_1, \ldots, z_n) \in \mathbb{R}^n$ by $z_i = \sum \text{vol}(F)$, where the sum is taken over all d-simplices F in T having v_i as a vertex.

Secondary polytope: The secondary polytope $\Sigma(V)$ is the convex hull of the z-vectors of all regular triangulations of V.

Link: If F is a face of a triangulation T, then the link of F is the set $\{G \mid G$ is a face of T, $\text{conv}(F \cup G)$ is a face of T of dimension $\dim F + \dim G + 1$, and $F \cap G = \emptyset\}$.

Adjacent triangulations: Suppose T is a triangulation of V (not necessarily regular). Suppose there is a subset W of $k+2$ points in V such that $\dim(\text{aff}(W)) = k$, T contains one of the (only) two triangulations of W, and the links with respect to T of all the k-dimensional faces F in the triangulation of W are identical. Then it is possible to interchange the triangulations of W, giving the new k-simplices the same links with respect to T, and thereby obtain a new triangulation of V. This operation is called a ***flip***, and the resulting triangulation is said to be adjacent to T.

Connected: Two triangulations (not necessarily regular) are said to be connected if one can be obtained from the other by a sequence of flips.

The secondary polytope plays an important role in the study of Gröbner bases [Stu96] and generalized discriminants and determinants [GKZ94].

MAIN RESULTS

1. The collection of all regular subdivisions of the set V, partially ordered by refinement, is combinatorially equivalent to the boundary complex of the polytope $\Sigma(V)$, which has dimension $n-d-1$ [GKZ94].

2. The vertices of $\Sigma(V)$ are precisely the z-vectors. In particular, no two regular triangulations have the same z-vector. The edges of $\Sigma(V)$ correspond to adjacent regular triangulations [GKZ94].

3. As an immediate consequence of the existence of $\Sigma(V)$, every regular triangulation has at least $n-d-1$ adjacent triangulations, and every pair of regular triangulations is connected. There are examples of nonregular triangulations with fewer than $n-d-1$ adjacent triangulations [deL95]. It is an open problem to determine in general whether or not every pair of triangulations, regular or not, is connected, although this is true of point sets in $\mathbb{R}^2$ and of vertex sets of cyclic polytopes [Ram96]. In particular, it is unknown if a (necessarily nonregular) triangulation exists with no adjacent triangulations.

4. $\Sigma(V)$ can also be expressed as a discrete or continuous Minkowski sum of polytopes coming from a representation of V as a projection of the vertices of an $(n-1)$-dimensional simplex. See [Zie95].

5. In the special case that $n = d+2$, there are precisely two nontrivial subdivisions of V (both regular), so that $\Sigma(V)$ is a line segment.

6. In the special case that $n = d+3$, all subdivisions are regular, and $\Sigma(V)$ is a convex polygon.

7. In the special case that V is the set of vertices of a convex n-gon, $\Sigma(V)$ is called the ***associahedron*** [Lee89]. Its dual is a simplicial polytope Q of

dimension $n-3$ having the following f-vector and h-vector:

$$f_{j-1}(Q) = \frac{1}{n-1}\binom{n-3}{j}\binom{n+j-1}{j+1}, \quad 0 \le j \le n-3,$$

$$h_i(Q) = \frac{1}{n-1}\binom{n-3}{i}\binom{n-1}{i+1}, \quad 0 \le i \le n-3.$$

From the discussion in Section 14.5.3, $f_{j-1}(Q)$ is the number of subdivisions of the n-gon having exactly j diagonals. There are various combinatorial interpretations of the h-vector. Explicit coordinates and inequalities for $\Sigma(V)$ can be found in [Zie95].

Figure 14.6.1 shows the five regular triangulations of a set of 5 points in $\mathbb{R}^2$, marking which pairs of triangulations are adjacent.

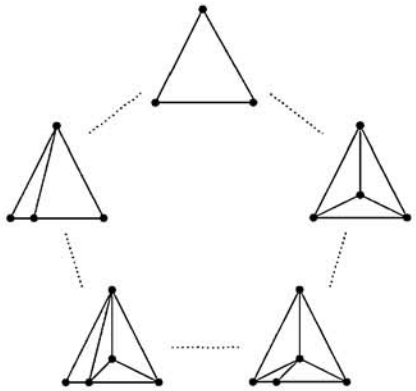


FIGURE 14.6.1
A polygon of regular triangulations.

The secondary polytope of the product of two simplices is discussed, for example, in [deL96, GKZ94]. See [dHSS96] for properties of the polytope that is the convex hull of the $(0,1)$ incidence vectors of all triangulations of V, and for the relationship of this polytope to $\Sigma(V)$. The special case when V is the set of vertices of a convex n-gon was first described in [DHH85].

14.6.1 FIBER POLYTOPES

A secondary polytope is a special case of a fiber polytope, which is associated with an affine map $\pi : P \to Q$ from a polytope P in $\mathbb{R}^p$ onto a polytope Q in $\mathbb{R}^q$. Such a map induces certain regular subdivisions of Q (called *π-coherent subdivisions*). The fiber polytope $\Sigma(P,Q)$ has dimension $\dim(P) - \dim(Q)$, and its nonempty faces correspond to these π-coherent subdivisions.

A ***section*** is a continuous map $\gamma : Q \to P$ with $\pi(\gamma(x)) = x$ for all $x \in Q$. The ***fiber polytope*** is defined to be the set of all average values of the sections of π:

$$\Sigma(P,Q) = \left\{ \frac{1}{\mathrm{vol}(Q)} \int_Q \gamma(x)dx \mid \gamma \text{ is a section of } \pi \right\}.$$

The associahedron and the permutohedron (see Chapter 13) are examples of fiber polytopes, and there are applications to zonotopal subdivisions and oriented matroids. For more details, see [Zie95].

14.7 SOURCES AND RELATED MATERIAL

FURTHER READING

Chapter 22 discusses triangulations of more general (e.g., nonconvex) objects. Chapter 20 provides details on Delaunay triangulations and Voronoi diagrams. Refer also to Chapter 13, on basic properties of convex polytopes.

A section on triangulations and subdivisions of convex polytopes can be found in the survey article [BL93]. The book [Zie95] and the article [Lee91] contain information on regular subdivisions and triangulations; for their important role in generalized discriminants and determinants see the book [GKZ94], and for their significance in computational algebra see the book [Stu96]. Additional references can be found in the above-mentioned sources, as well as the citations given in this chapter.

RELATED CHAPTERS

Chapter 3: Tilings
Chapter 7: Lattice points and lattice polytopes
Chapter 13: Basic properties of convex polytopes
Chapter 15: Face numbers of polytopes and complexes
Chapter 19: Convex hull computations
Chapter 20: Voronoi diagrams and Delaunay triangulations
Chapter 22: Triangulations
Chapter 27: Computational convexity

REFERENCES

[Bay88] M.M. Bayer. Barycentric subdivisions. *Pacific J. Math.*, 135:1–16, 1988.

[Bay94] M.M. Bayer. Face numbers and subdivisions of convex polytopes. In T. Bisztriczky, P. McMullen, R. Schneider, and A. Ivić Weiss, editors., *Polytopes: Abstract, Convex and Computational*, volume 440 of *NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci.*, pages 155–171. Kluwer, Dordrecht, 1994.

[BL93] M.M. Bayer and C.W. Lee. Combinatorial aspects of convex polytopes. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 485–534. Elsevier, Amsterdam, 1993.

[Big91] F. Bigdeli. *Regular Triangulations of Convex Polytopes and d-Cubes.* PhD thesis, Univ. of Kentucky, Lexington, 1991.

[BCS88] L.J. Billera, R. Cushman, and J.A. Sanders. The Stanley decomposition of the harmonic oscillator. *Nederl. Akad. Wetensch. Indag. Math.*, 50:375–393, 1988.

[Bre94] F. Brenti. q-Eulerian polynomials arising from Coxeter groups. *European J. Combin.*, 15:417–441, 1994.

[DHH85] G.B. Dantzig, A.J. Hoffman, and T.C. Hu. Triangulations (tilings) and certain block triangular matrices. *Math. Programming*, 31:1–14, 1985.

[deL95] J. de Loera. *Triangulations of Polytopes and Computational Algebra.* PhD thesis, Cornell Univ., Ithaca, 1995.

[deL96] J. de Loera. Nonregular triangulations of products of simplices. *Discrete Comput. Geom.*, 15:253–264, 1996.

[dHSS96] J. de Loera, S. Hoşten, F. Santos, and B. Sturmfels. The polytope of all triangulations of a point configuration. *Doc. Math.*, 1:103–119, 1996.

[ES96] H. Edelsbrunner and N.R. Shah. Incremental topological flipping works for regular triangulations. *Algorithmica*, 15:223–241, 1996.

[GKZ94] I.M. Gel'fand, M.M. Kapranov, and A.V. Zelevinsky. *Discriminants, Resultants and Multidimensional Determinants.* Birkhäuser, Boston, 1994.

[GP88] J.E. Goodman and J. Pach. Cell decomposition of polytopes by bending. *Israel J. Math.*, 64:129–138, 1988.

[Hai91] M. Haiman. A simple and relatively efficient triangulation of the n-cube. *Discrete Comput. Geom.*, 6:287–289, 1991.

[Hug93] R.B. Hughes. Minimum-cardinality triangulations of the d-cube for $d = 5$ and $d = 6$. *Discrete Math.*, 118:75–118, 1993.

[Lee89] C.W. Lee. The associahedron and triangulations of the n-gon. *European J. Combin.*, 10:551–560, 1989.

[Lee91] C.W. Lee. Regular triangulations of convex polytopes. In P. Gritzmann and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, volume 4 of *DIMACS Series in Discrete Math. and Theor. Comput. Sci.*, pages 443–456. Amer. Math. Soc., Providence, 1991.

[Luc87] J.M. Lucas. The rotation graph of binary trees is Hamiltonian. *J. Algorithms*, 8:503–535, 1987.

[Luc89] F. Luccio. On the upper bound on the rotation distance of binary trees. *Inform. Process. Lett.*, 31:57–60, 1989.

[Ram96] J. Rambau. Triangulations of cyclic polytopes and higher Bruhat orders. Tech. Report No. 496/1996, Fachbereich Mathematik, Technische Universät Berlin, 1996.

[Sal82] J.F. Sallee. A triangulation of the n-cube. *Discrete Math.*, 40:81–86, 1982.

[Sal84] J.F. Sallee. The middle-cut triangulations of the n-cube. *SIAM J. Algebraic Discrete Meth.*, 5:407–419, 1984.

[STT88] D.D. Sleator, R.E. Tarjan, and W.P. Thurston. Rotation distance, triangulations, and hyperbolic geometry. *J. Amer. Math. Soc.*, 1:647–681, 1988.

[Sta80] R.P. Stanley. Decompositions of rational convex polytopes. *Ann. Discrete Math.*, 6:333–342, 1990.

[Sta92] R.P. Stanley. Subdivisions and local h-vectors. *J. Amer. Math. Soc.*, 5:805–851, 1992.

[Stu96] B. Sturmfels. *Gröbner Bases and Convex Polytopes.* Volume 8 of *Univ. Lecture Ser.*, Amer. Math. Soc., Providence, 1996.

[Tod76] M.J. Todd. *The Computation of Fixed Points and Applications.* Volume 124 of *Lecture Notes in Econom. and Math. Systems*, Springer-Verlag, Berlin/New York, 1976.

[Zie95] G.M. Ziegler. *Lectures on Polytopes.* Volume 152 of *Graduate Texts in Math.*, Springer-Verlag, New York, 1995.

15 FACE NUMBERS OF POLYTOPES AND COMPLEXES

Louis J. Billera and Anders Björner

INTRODUCTION

Geometric objects are often put together from simple pieces according to certain combinatorial rules. As such, they can be described as *complexes* with their constituent *cells*, which are usually polytopes and often simplices. Many constraints of a combinatorial and topological nature govern the incidence structure of cell complexes and are therefore relevant in the analysis of geometric objects. Since these incidence structures are in most cases too complicated to be well understood, it is worthwhile to focus on simpler invariants that still say something nontrivial about their combinatorial structure. The invariants to be discussed in this chapter are the *f-vectors* $f = (f_0, f_1, \ldots)$, where f_i is the number of i-dimensional cells in the complex.

The theory of f-vectors can be discussed at two levels: (1) the numerical relations satisfied by the f_i numbers, and (2) the algebraic, combinatorial, and topological facts and constructions that give rise to and explain these relations. This chapter will summarize the main facts in the numerology of f-vectors (i.e., at level 1), with emphasis on cases of geometric interest.

The chapter is organized as follows. To begin with, we treat simplicial complexes, first the general case (Section 15.1), then complexes with various Betti number constraints (Section 15.2), and finally triangulations of spheres, polytope boundaries, and manifolds (Section 15.3). Then we move on to nonsimplicial complexes, discussing first the general case (Section 15.4) and then polytopes and spheres (Section 15.5).

15.1 SIMPLICIAL COMPLEXES

GLOSSARY

The convex hull of any set of $j+1$ affinely independent points in $\mathbb{R}^n$ is called a ***j-simplex***. See Chapter 13 for more about this definition, and for the notions of *faces* and *vertices* of a simplex.

A ***geometric simplicial complex*** Γ is a finite nonempty family of simplices in $\mathbb{R}^n$ such that (i) $\sigma \in \Gamma$ implies that $\tau \in \Gamma$ for every face τ of σ, and (ii) if $\sigma, \tau \in \Gamma$ and $\sigma \cap \tau \neq \emptyset$ then $\sigma \cap \tau$ is a face of both σ and τ.

An ***abstract simplicial complex*** Δ is a finite nonempty family of subsets of

some ground set V (the ***vertex set***) such that if $A \in \Delta$ and $B \subseteq A$ then $B \in \Delta$. (Note that always $\emptyset \in \Delta$.) The elements $A \in \Delta$ are called ***faces***. Define the ***dimension*** of a face A and of Δ itself by $\dim A = |A| - 1$; $\dim \Delta = \max_{A \in \Delta} \dim A$. By a ***d-complex*** we mean a d-dimensional complex.

With every geometric simplicial complex Γ we associate an abstract simplicial complex by taking the family of vertex sets of its simplices. Conversely, every d-dimensional abstract simplicial complex Δ can be realized in $\mathbb{R}^n$ for $n \geq 2d+1$ (and sometimes less) by some geometric simplicial complex. The latter is unique up to homeomorphism, so it is correct to think of the realization map as a one-to-one correspondence between abstract and geometric simplicial complexes. We will therefore drop the adjectives "abstract" and "geometric" and speak only of a ***simplicial complex***.

For a simplicial complex Δ, let $\Delta^i = \{i\text{-dimensional faces}\}$ and let $f_i = |\Delta^i|$. The integer sequence $f(\Delta) = (f_0, f_1, \ldots)$ is called the ***f-vector*** of Δ. (The entry $f_{-1} = 1$ is usually suppressed.) The subcomplex $\Delta^{\leq i} = \bigcup_{j \leq i} \Delta^j$ is called the ***i-skeleton*** of Δ.

A simplicial complex Δ is called ***pure*** if all maximal faces are of equal dimension. It is called ***r-colorable*** if there exists a partition of the vertex set $V = V_1 \cup \ldots \cup V_r$ such that $|A \cap V_i| \leq 1$ for all $A \in \Delta$ and $1 \leq i \leq r$. Equivalently, Δ is r-colorable if and only if its 1-skeleton $\Delta^{\leq 1}$ is r-colorable in the standard sense of graph theory. An $(r-1)$-complex that is both pure and r-colorable is sometimes called ***balanced***.

For integers $k, n \geq 1$ there is a unique way of writing

$$n = \binom{a_k}{k} + \binom{a_{k-1}}{k-1} + \ldots + \binom{a_i}{i}$$

so that $a_k > a_{k-1} > \ldots > a_i \geq i \geq 1$. Then define

$$\partial_k(n) = \binom{a_k}{k-1} + \binom{a_{k-1}}{k-2} + \ldots + \binom{a_i}{i-1},$$

and

$$\partial^k(n) = \binom{a_k - 1}{k-1} + \binom{a_{k-1} - 1}{k-2} + \ldots + \binom{a_i - 1}{i-1}.$$

Also let $\partial_k(0) = \partial^k(0) = 0$.

Let $\mathbb{N}^\infty$ denote the set of sequences $(n_0, n_1, \ldots)$ of nonnegative integers, and $\mathbb{N}^{(\infty)}$ the subset of sequences such that $n_k = 0$ for sufficiently large k. We call $n \in \mathbb{N}^{(\infty)}$ a ***K-sequence*** if

$$\partial_{k+1}(n_k) \leq n_{k-1} \qquad \text{for all } k \geq 1.$$

We call $n \in \mathbb{N}^\infty$ an ***M-sequence*** if

$$n_0 = 1 \text{ and } \partial^k(n_k) \leq n_{k-1} \qquad \text{for all } k \geq 2.$$

THE KRUSKAL-KATONA THEOREM AND SOME RELATIVES

The following basic result characterizes the f-vectors of simplicial complexes.

THEOREM 15.1.1 *Kruskal-Katona Theorem*
For $f = (f_0, f_1, \ldots) \in \mathbb{N}^{(\infty)}$ the following are equivalent:

(i) *f is the f-vector of a simplicial complex;*
(ii) *f is a K-sequence.*

A simplicial complex is ***connected*** if its 1-skeleton is connected in the sense of graph theory.

THEOREM 15.1.2
For $f \in \mathbb{N}^{(\infty)}$ the following are equivalent:

(i) *f is the f-vector of a connected simplicial complex;*
(ii) *f is a K-sequence and $\partial^3(f_2) \leq f_1 - f_0 + 1$.*

Theorem 15.1.1 has a generalization to colored complexes, whose statement will require some additional definitions. Fix an integer $r > 0$. Then define $\binom{n}{k}_r$ as follows: partition $\{1, \ldots, n\}$ into r subsets $V_1, \ldots, V_r$ as evenly as possible (so every subset V_i will have $\lfloor \frac{n}{r} \rfloor$ or $\lfloor \frac{n}{r} \rfloor + 1$ elements), and let $\binom{n}{k}_r$ be the number of k-subsets $F \subseteq \{1, \ldots, n\}$ such that $|F \cap V_i| \leq 1$ for $1 \leq i \leq r$. For $k \leq r$ every positive integer n can be uniquely written

$$n = \binom{a_k}{k}_r + \binom{a_{k-1}}{k-1}_{r-1} + \ldots + \binom{a_i}{i}_{r-k+i},$$

where $a_k > a_{k-1} > \ldots > a_i \geq i \geq 1$. Then define

$$\partial_k^{(r)}(n) = \binom{a_k}{k-1}_r + \binom{a_{k-1}}{k-2}_{r-1} + \ldots + \binom{a_i}{i-1}_{r-k+i},$$

and let $\partial_k^{(r)}(0) = 0$.

THEOREM 15.1.3
For $f = (f_0, \ldots, f_{d-1})$, $d \leq r$, the following are equivalent:

(i) *f is the f-vector of an r-colorable simplicial complex;*
(ii) *$\partial_{k+1}^{(r)}(f_k) \leq f_{k-1}$, for all $1 \leq k \leq d-1$.*

Note that for r sufficiently large Theorem 15.1.3 specializes to Theorem 15.1.1.

MULTICOMPLEXES AND MACAULAY'S THEOREM

A ***multicomplex*** $\mathcal{M}$ is a nonempty collection of monomials in finitely many variables such that if m is in $\mathcal{M}$ then so is every divisor of m. Let $f_i(\mathcal{M})$ be the number of degree i monomials in $\mathcal{M}$; $f(\mathcal{M}) = (f_0, f_1, \ldots)$ is called the ***f-vector*** of $\mathcal{M}$.

THEOREM 15.1.4 *Macaulay's Theorem*
For $f \in \mathbb{N}^{\infty}$ the following are equivalent:

(i) *f is the f-vector of a multicomplex;*

(ii) *f is an M-sequence;*

(iii) *$f_i = \dim_{\boldsymbol{k}} R_i$, $i \geq 0$, for some finitely generated commutative graded $\boldsymbol{k}$-algebra $R = \oplus_{i \geq 0} R_i$ such that $R_0 \cong \boldsymbol{k}$ (a field) and R_1 generates R.*

A simplicial complex can be identified with a multicomplex of squarefree monomials. Hence, a K-sequence is (except for a shift in the indexing) an M-sequence: If $(f_0, \ldots, f_{d-1})$ is a K-sequence then $(1, f_0, \ldots, f_{d-1})$ is an M-sequence. For this reason (and others, see, e.g., Theorem 15.2.2), properties of M-sequences are relevant also even if one is interested only in the special case of simplicial complexes.

A multicomplex is ***pure*** if all its maximal (under divisibility) monomials have the same degree.

THEOREM 15.1.5

Let $(f_0, \ldots, f_r)$ be the f-vector of a pure multicomplex, $f_r \neq 0$. Then $f_i \leq f_j$ for all $i < j \leq r - i$.

COMMENTS

Simplicial complexes (abstract and geometric) are treated in most books on algebraic topology; see, e.g., [Mun84, Spa66]. The Kruskal-Katona theorem (independently discovered by M.-P. Schützenberger, J.B. Kruskal, G.O.H. Katona, L.H. Harper, and B. Lindström during the years 1959-1966) is discussed in many places and several proofs have appeared; see, e.g., [And87, Zie95]. Theorems 15.1.2 and 15.1.3 are from [Bjö96] and [FFK88] respectively. A Kruskal-Katona type theorem for simplicial complexes with vertex-transitive symmetry group appears in [FK96].

For Macaulay's theorem we refer to [And87, Sta96]. There is a common generalization of Macaulay's theorem and the Kruskal-Katona theorem due to Clements and Lindström; see [And87]. Theorem 15.1.5 is from [Hib89].

15.2 BETTI NUMBER CONSTRAINTS

GLOSSARY

The ***Euler characteristic*** $\chi(\Delta)$ of a simplicial complex Δ with f-vector $(f_0, \ldots, f_{d-1})$ is $\chi(\Delta) = \sum_{i=0}^{d-1} (-1)^i f_i$.

The ***h-vector*** $(h_0, \ldots, h_d)$ of a $(d-1)$-dimensional simplicial complex is defined by

$$\sum_{i=0}^{d} h_i x^{d-i} = \sum_{i=0}^{d} f_{i-1} (x-1)^{d-i}.$$

The corresponding ***g-vector*** $(g_0, \ldots, g_{\lfloor d/2 \rfloor})$ is defined by $g_0 = 1$ and $g_i = h_i - h_{i-1}$, for $i \geq 1$.

The ***Betti number*** $\beta_i(\Delta)$ is the dimension (as a $\mathbb{Q}$-vector space) of the i-th reduced simplicial homology group $\widetilde{H}_i(\Delta, \mathbb{Q})$; see any textbook on algebraic topology

(e.g., [Mun84]) for the definition. We call $(\beta_0, \ldots, \beta_{\dim \Delta})$ the ***Betti sequence*** of Δ.

The ***link*** $\ell k_\Delta(F)$ of a face F is the subcomplex of Δ defined by $\ell k_\Delta(F) = \{A \in \Delta \mid A \cap F = \emptyset, A \cup F \in \Delta\}$. Note that $\ell k_\Delta(\emptyset) = \Delta$.

A simplicial complex Δ is ***acyclic*** if $\beta_i(\Delta) = 0$ for all i.

A simplicial complex Δ is ***Cohen-Macaulay*** if $\beta_i(\ell k_\Delta(F)) = 0$ for all $F \in \Delta$ and all $i < \dim \ell k_\Delta(F)$.

A simplicial complex Δ is ***m-Leray*** if $\beta_i(\ell k_\Delta(F)) = 0$ for all $F \in \Delta$ and all $i \geq m$.

FIXED BETTI NUMBERS

The most basic relationship between f-vectors and Betti numbers is the ***Euler-Poincaré formula***:

$$\chi(\Delta) = f_0 - f_1 + f_2 - \ldots = 1 + \beta_0 - \beta_1 + \beta_2 - \ldots$$

This is in fact the only linear one in the following complete set of relations.

THEOREM 15.2.1

For $f = (f_0, f_1, \ldots) \in \mathbb{N}^{(\infty)}$ and $\beta = (\beta_0, \beta_1, \ldots) \in \mathbb{N}^{(\infty)}$ the following are equivalent:

(i) *f is the f-vector of some simplicial complex with Betti sequence β;*

(ii) *if $\chi_{k-1} = \sum_{j \geq k} (-1)^{j-k}(f_j - \beta_j)$, $k \geq 0$, then $\chi_{-1} = 1$ and $\partial_{k+1}(\chi_k + \beta_k) \leq \chi_{k-1}$ for all $k \geq 1$.*

By putting $\beta_i = 0$ for all i one gets as a special case a characterization of the f-vectors of acyclic simplicial complexes, viz., $\sum_{i \geq 0} f_{i-1} x^i = (1 + x) \sum_{i \geq 0} f'_{i-1} x^i$, where $(f'_0, f'_1, \ldots)$ is a K-sequence.

COHEN-MACAULAY COMPLEXES

Examples of Cohen-Macaulay complexes are triangulations of manifolds whose Betti numbers vanish below the top dimension, in particular triangulations of spheres and balls. Other examples are matroid complexes (the independent sets of a matroid), Tits buildings, and the order complexes (simplicial complex of totally ordered subsets) of several classes of posets, e.g., semimodular lattices (including distributive and geometric lattices). Shellable complexes (see Chapters 14 and 17) are Cohen-Macaulay. Cohen-Macaulay complexes are always pure.

The definition of h-vector given in the glossary shows that the h-vector and the f-vector of a complex mutually determine each other via the formulas:

$$h_i = \sum_{j=0}^{i} (-1)^{i-j} \binom{d-j}{i-j} f_{j-1}\,, \qquad f_{i-1} = \sum_{j=0}^{i} \binom{d-j}{i-j} h_j\,,$$

for $0 \leq i \leq d$. Hence, we may state f-vector results in terms of h-vectors whenever convenient.

THEOREM 15.2.2

For $h = (h_0, \ldots, h_d) \in \mathbb{Z}^{d+1}$ the following are equivalent:

(i) *h is the h-vector of a $(d-1)$-dimensional Cohen-Macaulay complex;*
(ii) *h is the h-vector of a $(d-1)$-dimensional shellable complex;*
(iii) *h is an M-sequence.*

Since there are a total of $\binom{n+k-1}{k}$ monomials of degree k in n variables, and by Theorems 15.1.4 and 15.2.2 the h-vector of a $(d-1)$-dimensional Cohen-Macaulay complex counts certain monomials in $h_1 = f_0 - d$ variables, we derive the inequalities

$$0 \leq h_i \leq \binom{f_0 - d + i - 1}{i}$$

for the h-vectors of Cohen-Macaulay complexes. The lower bound can be improved for complexes with fixed-point-free involutive symmetry.

THEOREM 15.2.3

Let $h = (h_0, \ldots, h_d)$ be the h-vector of a Cohen-Macaulay complex admitting an automorphism α of order 2, such that $\alpha(F) \neq F$ for all $F \in \Delta \setminus \{\emptyset\}$. Then

$$h_i \geq \binom{d}{i} \qquad \text{for } 0 \leq i \leq d.$$

Consequently, $f_{d-1} = h_0 + \ldots + h_d \geq 2^d$.

Another condition on a Cohen-Macaulay complex that forces stricter conditions on its h-vector is being r-colorable.

THEOREM 15.2.4

For $h = (h_0, \ldots, h_d) \in \mathbb{Z}^{d+1}$ the following are equivalent:

(i) *h is the h-vector of a $(d-1)$-dimensional and d-colorable Cohen-Macaulay complex;*
(ii) *$(h_1, \ldots h_d)$ is the f-vector of a d-colorable simplicial complex.*

Hence in this case the h-vector is not only an M-sequence, but the special kind of K-sequence characterized in Theorem 15.1.3.

LERAY COMPLEXES

Examples of Leray complexes arise as follows. Let $\mathcal{K} = \{K_1, \ldots, K_t\}$ be a family of convex sets in $\mathbb{R}^m$, and let $\Delta(\mathcal{K}) = \{A \subseteq \{1, \ldots, t\} \mid \bigcap_{i \in A} K_i \neq \emptyset\}$. Then the simplicial complex $\Delta(\mathcal{K})$ is m-Leray.

Fix $m \geq 0$, and let $f = (f_0, \ldots, f_{d-1})$ be the f-vector of a simplicial complex Δ. Define

$$h_k^* = \begin{cases} f_k & \text{for } 0 \leq k \leq m-1 \\ \sum_{j \geq 0} (-1)^j \binom{k+j-m}{j} f_{k+j} & \text{for } k \geq m. \end{cases}$$

The sequence $h^* = (h_0^*, \ldots, h_{d-1}^*)$ is the ***h**-vector** of Δ. The two vectors f and h^* mutually determine each other.

THEOREM 15.2.5

For $h^* = (h_0^*, h_1^*, \ldots) \in \mathbb{Z}^{(\infty)}$ the following are equivalent:

(i) *h^* is the h^*-vector of an m-Leray complex;*

(ii) *h^* is the h^*-vector of $\Delta(\mathcal{K})$ for some family $\mathcal{K}$ of convex sets in $\mathbb{R}^m$;*

(iii)
$$\begin{cases} h_k^* \geq 0 & \text{for } k \geq 0 \\ \partial_{k+1}(h_k^*) \leq h_{k-1}^* & \text{for } 1 \leq k \leq m-1 \\ \partial_m(h_k^*) \leq h_{k-1}^* - h_k^* & \text{for } k \geq m. \end{cases}$$

COMMENTS

The Euler-Poincaré formula (due to Poincaré 1899) is proved in most books on algebraic topology. Theorem 15.2.1 is from [BK88]. A good general source on Cohen-Macaulay complexes is [Sta96]; Theorems 15.2.2, 15.2.3, and 15.2.4, as well as references to the original sources, can be found there. A generalization of Theorem 15.2.2 to complexes whose k-skeleton is Cohen-Macaulay appears in [Bjö96]. There are several additional results about h-vectors of Cohen-Macaulay complexes. For instance, for complexes with nontrivial automorphism groups, see [Sta96, Section III.8]; for matroid complexes, see [Sta96, Section III.3]; and for Cohen-Macaulay complexes that are r-colorable for $r < d$, see the references mentioned in [Sta96, Section III.4].

Cohen-Macaulay complexes are closely related to certain commutative rings [Sta96], and via this connection such complexes have also been of use in the theory of splines; see [Sta96, Section III.5] and also Chapter 45.

Theorem 15.2.5 was conjectured by Eckhoff and proved by Kalai [Kal84, Kal86].

15.3 SIMPLICIAL POLYTOPES, SPHERES, AND MANIFOLDS

GLOSSARY

A ***triangulated d-ball*** is a simplicial complex Δ whose realization $\|\Delta\|$ is homeomorphic to the ball $\{x \in \mathbb{R}^d \mid x_1^2 + \cdots + x_d^2 \leq 1\}$. A ***triangulated (d−1)-sphere*** is a simplicial complex whose realization is homeomorphic to the sphere $\{x \in \mathbb{R}^d \mid x_1^2 + \cdots + x_d^2 = 1\}$. Equivalently, it is the boundary of a triangulated d-ball. Examples of triangulated $(d-1)$-spheres are given by the boundary complexes of simplicial d-polytopes.

A ***pseudomanifold*** is a pure simplicial complex Δ such that

(i) each face of codimension 1 is contained in precisely two maximal faces; and

(ii) the dual graph (whose vertices are the maximal faces of Δ and whose edges are the faces of codimension 1) is connected.

An ***Eulerian pseudomanifold*** is a pseudomanifold Δ such that Δ and the link of each face have the Euler characteristic of a sphere of the corresponding dimension.

A pure $(d-1)$-dimensional simplicial complex Δ is a ***homology manifold*** if it is connected and the link of each nonempty face has the Betti numbers of a sphere of the same dimension. It is a ***homology sphere*** if, in addition, Δ itself has the Betti numbers of a $(d-1)$-sphere. Examples of homology manifolds are given by triangulations of compact connected topological manifolds, i.e., spaces that are locally Euclidean.

The ***cyclic d-polytope with n vertices*** $C_d(n)$ is the convex hull of any n points on the moment curve in $\mathbb{R}^d$. (See Section 13.1.4.)

The following implications hold among these various classes, all of them strict:

$$\begin{gathered}\textit{polytope boundary} \Rightarrow \textit{sphere} \Rightarrow \textit{homology sphere} \Rightarrow \\ \textit{Eulerian pseudomanifold} \Rightarrow \textit{pseudomanifold}\end{gathered}$$

$$\textit{homology sphere} \Rightarrow \textit{homology manifold} \Rightarrow \textit{pseudomanifold}$$

$$\textit{homology sphere} \Rightarrow \textit{Cohen-Macaulay complex}$$

PSEUDOMANIFOLDS

The following results give the basic lower and upper bounds on f-vectors of pseudomanifolds.

THEOREM 15.3.1 *Lower Bound Theorem*

For a $(d-1)$-dimensional pseudomanifold Δ with n vertices,

$$f_k(\Delta) \geq \begin{cases} \binom{d}{k}n - \binom{d+1}{k+1}k & \text{for } 1 \leq k \leq d-2 \\ (d-1)n - (d-2)(d+1) & \text{for } k = d-1. \end{cases}$$

THEOREM 15.3.2 *Upper Bound Theorem*

Let Δ be a $(d-1)$-dimensional homology manifold with n vertices, such that either

(i) *d is even, or*

(ii) *$d = 2k+1$ is odd, and either $\chi(\Delta) = 2$ or $\beta_k \leq 2\beta_{k-1} + 2\sum_{i=0}^{k-3} \beta_i$.*

Then $f_k(\Delta) \leq f_k(C_d(n))$ for $1 \leq k \leq d-1$.

This upper bound theorem applies when the homology manifold is Eulerian (irrespective of dimension); in particular, it applies to all simplicial polytopes and spheres. By the geometric operation of "pulling vertices" (Section 14.2), one can extend this to all convex polytopes.

THEOREM 15.3.3

If P is any *convex d-polytope with n vertices, then $f(P) \leq f(C_d(n))$.*

The given lower and upper bounds are best possible within the class of simplicial polytope boundaries. The lower bound is attained by the class of stacked polytopes

(Sections 14.4.2 and 17.2). To make the upper bound numerically explicit, we give the formula for the f-vector of a cyclic polytope.

THEOREM 15.3.4

For $d \geq 2$ and $0 \leq k \leq d-1$, the number of k-faces of the cyclic polytope $C_d(n)$ with n vertices is

$$f_k(C_d(n)) = \frac{n-\delta(n-k-2)}{n-k-1} \sum_{j=0}^{\lfloor d/2 \rfloor} \binom{n-1-j}{k+1-j}\binom{n-k-1}{2j-k-1+\delta},$$

where $\delta = d - 2\lfloor d/2 \rfloor$. In particular,

$$f_{d-1}(C_d(n)) = \binom{n-\lfloor \frac{d+1}{2} \rfloor}{n-d} + \binom{n-\lfloor \frac{d+2}{2} \rfloor}{n-d}.$$

POLYTOPES AND SPHERES

For boundaries of simplicial d-polytopes and, more generally, for Eulerian pseudomanifolds, we have the following basic relations.

THEOREM 15.3.5 *Dehn-Sommerville Equations*

For d-dimensional Eulerian pseudomanifolds,

$$h_i = h_{d-i} \qquad \text{for all } 0 \leq i \leq d.$$

These equations give a complete description of the linear span of all f-vectors of d-polytopes (equivalently, $(d-1)$-spheres). (The affine span is defined by including the relation $h_0 = 1$.)

One consequence of the Dehn-Sommerville equations is the following relation between the h-vector of a triangulated ball K and the g-vector of its boundary ∂K.

THEOREM 15.3.6

For a triangulated d-ball K and its boundary $(d-1)$-sphere ∂K,

$$g_i(\partial K) = h_i(K) - h_{d+1-i}(K) \qquad \text{for } i \geq 1.$$

A complete characterization of the f-vectors of simplicial (and, by duality, simple) convex polytopes is given in terms of the h-vector and g-vector.

THEOREM 15.3.7 *g-Theorem*

A nonnegative integer vector $h = (h_0, \ldots, h_d)$ is the h-vector of a simplicial convex polytope if and only if

(i) $h_i = h_{d-i}$, *and*

(ii) $(g_0, \ldots, g_{\lfloor d/2 \rfloor})$ *is an M-sequence.*

One consequence of (ii) is that $g_i \geq 0$. For centrally symmetric polytopes, we get a better lower bound.

THEOREM 15.3.8

For centrally symmetric simplicial d-polytopes,

$$g_i = h_i - h_{i-1} \geq \binom{d}{i} - \binom{d}{i-1} \qquad \text{for } i \leq \lfloor d/2 \rfloor.$$

COMMENTS

The Lower Bound Theorem 15.3.1 is due to Kalai and Gromov in the generality given here; see [Kal87] including the note added in proof. The $k = d - 1$ case had earlier been done by Klee and the case of polytope boundaries by Barnette. See [Kal87] for a discussion of the history of this result.

The Upper Bound Theorem 15.3.2 is due to Novik [Nov]. The case of polytopes (Theorem 15.3.3) was first proved by McMullen (see [MS71]), and extended to spheres by Stanley (see [Sta96]). The computation of the f-vector of the cyclic polytope can be found in [Grü67, Sections 4.7.3 and 9.6.1] or [MS71].

The Dehn-Sommerville equations for polytopes are classical; proofs can be found in [Grü67, Sta86, Zie95]. The extension to Eulerian pseudomanifolds is due to Klee [Kle64]; an equivariant version appears in [Bar92]. The g-theorem was conjectured by McMullen and proved by Billera, Lee, and Stanley [BL81, Sta80]. More recently, another proof of the necessity of these conditions was given by McMullen [McM93]. It is not known whether the second condition of Theorem 15.3.7 holds for general triangulated spheres. The g-theorem has a convenient reformulation as a one-to-one correspondence (via matrix multiplication) between f-vectors of simplicial polytopes and M-sequences, see [Bjö87, Zie95]. Theorem 15.3.8 was proved by Stanley [Sta87a].

The study of f-vectors of unbounded polyhedra can be approached by studying the f-vectors of ***polytope pairs*** (P, F), where P is a polytope and F is a maximal face of P. See [BL93] for a summary of such results.

15.4 CELL COMPLEXES

GLOSSARY

Convex polytopes and *faces* of such are defined in Chapter 13.

A ***polyhedral complex*** Γ is a finite collection of convex polytopes in $\mathbb{R}^n$ such that (i) if $\pi \in \Gamma$ and σ is a face of π, then $\sigma \in \Gamma$; and (ii) if $\pi, \sigma \in \Gamma$ and $\pi \cap \sigma \neq \emptyset$, then $\pi \cap \sigma$ is a face of both. The ***space*** of Γ is $\|\Gamma\| = \bigcup_{\pi \in \Gamma} \pi$, a subspace of $\mathbb{R}^n$. Examples of polyhedral complexes are given by ***boundary complexes*** ∂P of convex polytopes P (i.e., the collection of all proper faces). A geometric simplicial complex (defined in Section 15.1) is a polyhedral complex all of whose cells are simplices. A ***cubical complex*** is a polyhedral complex all of whose cells are (combinatorially isomorphic to) cubes.

A ***regular cell complex*** Γ is a family of closed balls (homeomorphs of $\{x \in \mathbb{R}^j \mid |x| \leq 1\}$) in a Hausdorff space $\|\Gamma\|$ such that (i) the interiors of the balls

partition $\|\Gamma\|$ and (ii) the boundary of each ball in Γ is a union of other balls in Γ. The members of Γ are called (closed) cell***cells*** or ***faces***. The ***dimension*** of a cell is its topological dimension and $\dim \Gamma = \max_{\sigma \in \Gamma} \dim \sigma$.

A regular cell complex has the ***intersection property*** if, whenever the intersection of two cells is nonempty, then this intersection is also a cell in the complex. Polyhedral complexes are examples of regular cell complexes with the intersection property. Regular cell complexes with the intersection property can be reconstructed up to homeomorphism from the corresponding "abstract" complex consisting of the family of vertex sets of its cells.

For a regular cell complex Γ, let f_i be the number of i-dimensional cells, and let $\beta_i = \dim_{\mathbb{Q}} \widetilde{H}_i(\|\Gamma\|, \mathbb{Q})$. The latter denotes i-dimensional reduced singular homology with rational coefficients of the space $\|\Gamma\|$; see [Mun84, Spa66] for explanations of this concept. Then we have the ***f-vector*** $f = (f_0, f_1, \ldots)$ and the ***Betti sequence*** $\beta = (\beta_0, \beta_1, \ldots)$ of Γ. These definitions generalize those previously given in the simplicial case.

BASIC f-VECTOR RELATIONS

Among the classes of complexes

- simplicial complexes
- polyhedral complexes
- regular cell complexes with the intersection property
- regular cell complexes

each is a subclass of its successor. Thus one may wonder how many of the relations for f-vectors of simplicial complexes given in Sections 15.1–15.3 can be extended to these broader classes of complexes. Also, what new phenomena (not visible in the simplicial case) arise? Some answers will be given in this section and the following one, but current knowledge is quite fragmentary. We begin here with the most general relations.

THEOREM 15.4.1

$(f_0, \ldots, f_d)$ is the f-vector of a d-dimensional regular cell complex if and only if $f_d \geq 1$ and $f_i \geq 2$ for all $0 \leq i < d$.

THEOREM 15.4.2

f is the f-vector of a regular cell complex with the intersection property if and only if f is a K-sequence.

Let $\beta = (\beta_0, \beta_1, \ldots) \in \mathbb{N}^{(\infty)}$ be fixed, and for every sequence $f = (f_0, f_1, \ldots)$ let

$$\chi_{k-1} = \sum_{j \geq k} (-1)^{j-k} (f_j - \beta_j) \qquad \text{for } k \geq 0.$$

THEOREM 15.4.3

$(f_0, \ldots, f_d)$ is the f-vector of a d-dimensional regular cell complex with Betti sequence β if and only if $\chi_{-1} = 1$ and $\chi_k \geq 1$ for $0 \leq k < d$.

THEOREM 15.4.4

For $f \in \mathbb{N}^{(\infty)}$ the following are equivalent:

(i) *f is the f-vector of a regular cell complex with the intersection property and with Betti sequence β;*

(ii) *$\chi_{-1} = 1$ and $\partial_{k+1}(\chi_k + \beta_k) \leq \chi_{k-1}$ for all $k \geq 1$.*

These results show that the f-vectors of regular cell complexes (with or without Betti number constraints) are considerably more general than the f-vectors of simplicial complexes, but that the two classes of f-vectors agree in the presence of the intersection property.

COMMENTS

Regular cell complexes are known as ***regular CW complexes*** in the topological literature [LW69]. The nonregular CW complexes offer an even more general class of cell complexes [LW69, Mun84, Spa66], but there is very little one can say about f-vectors in that generality. See [BLS+93, Section 4.7] for a detailed discussion of regular cell complexes from a combinatorial point of view.

For the results of this section see [BK88, BK91, BK89]. A characterization of f-vectors of (cubical) subcomplexes of a cube can be found in [Lin71], and of regular cell decompositions of spheres in [Bay88].

15.5 GENERAL POLYTOPES AND SPHERES

GLOSSARY

A ***flag*** of faces in a (polyhedral) $(d-1)$-complex Δ is a chain $F_1 \subsetneq F_2 \subsetneq \cdots \subsetneq F_k$ in Δ. It is an ***S-flag*** if

$$S = \{\dim F_1, \ldots, \dim F_k\} \subseteq \{0, 1, \ldots, d-1\}.$$

If $f_S = f_S(\Delta)$ denotes the number of S-flags in Δ, then the function $S \mapsto f_S$, $S \subseteq \{0, 1, \ldots, d-1\}$, is called the ***flag f-vector*** of Δ.

If

$$h_S = \sum_{T \subseteq S} (-1)^{|S|-|T|} f_T,$$

then the function $S \mapsto h_S$, $S \subseteq \{0, 1, \ldots, d-1\}$, is called the ***flag h-vector***.

For $S \subseteq \{0, \ldots, d-1\}$ and noncommuting symbols $\boldsymbol{a}$ and $\boldsymbol{b}$, let $u_S = u_0 u_1 \cdots u_{d-1}$ be the $\boldsymbol{ab}$-word defined by $u_i = \boldsymbol{a}$ if $i \notin S$ and $u_i = \boldsymbol{b}$ otherwise. When Δ is spherical (or, more generally, Eulerian), then the $\boldsymbol{ab}$-polynomial $\sum h_S u_S$ is also a polynomial in $\boldsymbol{c} = \boldsymbol{a} + \boldsymbol{b}$ and $\boldsymbol{d} = \boldsymbol{ab} + \boldsymbol{ba}$. (Note that the degree of $\boldsymbol{c}$ is 1 and the degree of $\boldsymbol{d}$ is 2.) The resulting $\boldsymbol{cd}$-polynomial

$$\sum h_S u_S = \sum \phi_w w,$$

where the right-hand sum is over all $\boldsymbol{cd}$-words w of degree d, is called the ***cd-index*** of Δ. For 2- and 3-polytopes, the $\boldsymbol{cd}$-index is $\boldsymbol{c}^2 + (f_0 - 2)\boldsymbol{d}$ and $\boldsymbol{c}^3 + (f_0 - 2)\boldsymbol{dc} + (f_2 - 2)\boldsymbol{cd}$, respectively.

For *any* convex d-polytope P, we define the ***toric h-vector*** and ***toric g-vector*** recursively by $h(P,x) = \sum_{i=0}^{d} h_i x^{d-i}$ and $g(P,x) = \sum_{i=0}^{\lfloor d/2 \rfloor} g_i x^i$, where $g_i = h_i - h_{i-1}$ and the following relations hold:

(i) $g(\emptyset, x) = h(\emptyset, x) = 1$; and

(ii) $h(P,x) = \sum_{G \text{ face of } P,\, G \neq P} g(G,x)(x-1)^{d-1-\dim G}$.

(Compare to Chapter 14.4.2, where this toric h-vector is defined for any polyhedral complex. In the notation given there, we have defined h and g for the complex ∂P.) When P is simplicial, this definition coincides with that of the usual h-vector, as defined in Section 15.2.

A ***rational polytope*** is one whose vertices all have rational coordinates.

A ***cubical polytope*** is one that has a cubical boundary complex. For any cubical $(d-1)$-complex with f-vector $(f_0, \ldots, f_{d-1})$, define the ***cubical h-vector*** $h^c = (h_0^c, \ldots, h_d^c)$ by

$$h_i^c = (-1)^i 2^{d-1} + \sum_{j=1}^{i} (-1)^{i-j} 2^{j-1} f_{j-1} \sum_{k=0}^{i-j} \binom{d-j}{k} \qquad \text{for } i = 0, \ldots, d.$$

The ***cubical g-vector*** $g^c = (g_0^c, \ldots, g_{\lfloor d/2 \rfloor}^c)$ is defined by $g_0^c = h_0^c = 2^{d-1}$ and $g_i^c = h_i^c - h_{i-1}^c$ for $i \geq 1$.

An ***Eulerian polyhedral complex*** is one whose first barycentric subdivision is an Eulerian pseudomanifold. Examples are boundary complexes of polytopes and ***spherical*** polyhedral complexes, i.e., those whose underlying space is homeomorphic to a sphere.

A (central) ***hyperplane arrangement*** is a collection $\mathcal{H}$ of n linear hyperplanes in $\mathbb{R}^d$, given by normal vectors $x_1, \ldots, x_n$ (see Section 6.1.3). The arrangement is ***essential*** if the normals x_i span $\mathbb{R}^d$. The associated ***zonotope*** is the Minkowski sum of the n line segments $[-x_i, x_i]$, i.e., $Z = \{\sum \lambda_i x_i \mid -1 \leq \lambda_i \leq 1\}$ (see Section 13.1.4).

LINEAR RELATIONS

We give the linear relations on the invariants defined above that are known to hold for all boundary complexes of polytopes and, more generally, for all Eulerian polyhedral complexes.

THEOREM 15.5.1

For $(d-1)$-dimensional Eulerian polyhedral complexes, the following relations always hold for the flag h, the toric h, and the flag f:

(i) $h_S = h_{\{0,\ldots,d-1\} \setminus S}$ *for all* $S \subseteq \{0, \ldots, d-1\}$;

(ii) $h_i = h_{d-i}$ *for* $0 \leq i \leq d$; *and*

(iii) $\sum_{j=i+1}^{k-1} (-1)^{j-i-1} f_{S \cup \{j\}} = (1 - (-1)^{k-i-1}) f_S$ *whenever* $i, k \in S \cup \{-1, d\}$ *with* $i \leq k-2$ *and* $S \cap \{i+1, \ldots, k-1\} = \emptyset$.

It is known that the relations in Theorem 15.5.1(iii), the ***generalized Dehn-Sommerville equations***, completely describe the linear span of all flag f-vectors of Eulerian complexes, and so they imply those in (i). Since the toric h is known to be a linear function of the flag f, they imply those in (ii) as well. The linear span of flag f-vectors has dimension e_d, where e_d is the d^{th} Fibonacci number (defined by

the recurrence $e_d = e_{d-1} + e_{d-2}$, $e_0 = e_1 = 1$). There are e_d ***cd***-words of degree d. Furthermore, the coefficients ϕ_w of the ***cd***-index, considered as linear expressions in the f_S, form a linear basis for the span of flag f-vectors of d-polytopes. The affine span of all flag f-vectors is defined by including the relation $f_\emptyset = 1$.

For cubical polytopes and spheres, the cubical h-vector satisfies the analogue of the Dehn-Sommerville equations.

THEOREM 15.5.2

For cubical d-polytopes and cubical $(d-1)$-spheres,

$$h_i^c = h_{d-i}^c \qquad \textit{for all } 0 \le i \le d.$$

These give all linear relations satisfied by f-vectors of cubical polytopes and spheres. The cubical h-vector satisfies, as well, the equations of Theorem 15.3.6, linking the h of a cubical ball to the g of its boundary sphere.

LINEAR INEQUALITIES

Some linear inequalities that hold for flag f-vectors of all polytope boundaries are given in this section. The list is not thought to be complete, although there are no conjectures for what the complete set might be.

For a Cohen-Macaulay polyhedral complex, i.e., one whose first barycentric subdivision is a Cohen-Macaulay simplicial complex, the flag h is always nonnegative.

THEOREM 15.5.3

For a Cohen-Macaulay polyhedral $(d-1)$-complex Γ,

$$h_S(\Gamma) \ge 0 \qquad \textit{for all } S \subseteq \{0, \ldots, d-1\}.$$

For general convex polytopes, we also have nonnegativity of the ***cd***-index.

THEOREM 15.5.4

*For a convex d-polytope P, $\phi_w \ge 0$ for all **cd**-words w of degree d.*

For *rational* convex polytopes, it is known, further, that the toric h is unimodal.

THEOREM 15.5.5

For a rational convex d-polytope, $g_i \ge 0$ for $i \le \lfloor d/2 \rfloor$.

Related to this is the following *nonlinear* inequality holding between the g-vectors of a polytope P and any of its faces F. We denote by P/F the ***link*** of F in P, i.e., the polytope whose lattice of faces is (isomorphic to) the interval $[F, P]$ in the face lattice of P.

THEOREM 15.5.6

For a rational polytope P and any face F, we have the polynomial inequality

$$g(P,t) - g(F,t)g(P/F,t) \ge 0,$$

i.e., all coefficients of this polynomial are nonnegative.

Finally, we have the following lower bound for the number of vertices of polytopes with no triangular faces (this includes the class of cubical polytopes).

THEOREM 15.5.7

A d-polytope with no triangular 2-face has at least 2^d vertices.

HYPERPLANE ARRANGEMENTS AND ZONOTOPES

An essential hyperplane arrangement $\mathcal{H}$ defines a decomposition of $\mathbb{R}^d$ into polyhedral cones (as in Section 6.1.3). This decomposition $\Gamma_{\mathcal{H}}$, a regular cell complex if intersected with the unit sphere, has a flag f-vector dual to that of its associated zonotope Z, in the sense that $f_S(\Gamma_{\mathcal{H}}) = f_{d-S}(Z)$, where $S = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, d\}$ and $d - S = \{d - i_k, \ldots, d - i_1\}$.

THEOREM 15.5.8

The flag f-vector of an arrangement (or zonotope) depends only on the matroid (linear dependency structure) of the underlying point configuration $\{x_1, \ldots, x_n\}$.

Although a fairly special subclass of polytopes, the zonotopes nonetheless are varied enough to carry all the linear information carried by flag numbers of general polytopes.

THEOREM 15.5.9

The flag f-vectors of zonotopes (and thus of hyperplane arrangements) satisfy the generalized Dehn-Sommerville equations, and no other linear relations not implied by these.

When it comes to linear *inequalities*, however, the difference between zonotopes and general polytopes emerges. The following result has the most direct interpretation when it is stated for arrangements, where it bounds the average number of $S = \{i_1, \ldots, i_k\}$-flags in an i_k-face by the number of S-flags in an i_k-cube.

THEOREM 15.5.10

For a hyperplane arrangement $\mathcal{H}$ in $\mathbb{R}^d$ and $S = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, d\}$ with $k \geq 2$,

$$\frac{f_S(\Gamma_{\mathcal{H}})}{f_{i_k}(\Gamma_{\mathcal{H}})} < \binom{i_k}{i_1, i_2 - i_1, \ldots, i_k - i_{k-1}} 2^{i_k - i_1}.$$

There is a straightforward reformulation of Theorem 15.5.10 for zonotopes that is easily seen not to be valid for all polytopes.

GENERAL 3- AND 4-POLYTOPES

We describe here the situation for flag f-vectors of 3- and 4-polytopes. The equations in Theorem 15.5.1(iii) reduce consideration to (f_0, f_2) when $d = 3$ and to (f_0, f_1, f_2, f_{02}) when $d = 4$.

THEOREM 15.5.11

For 3-polytopes, the following is known about the vector (f_0, f_2).

(i) *An integer vector* (f_0, f_2) *is the* f*-vector of a 3-polytope if and only if* $f_0 \leq 2f_2 - 4$ *and* $f_2 \leq 2f_0 - 4$.

(ii) *An integer vector* (f_0, f_2) *is the* f*-vector of a cubical 3-polytope if and only if* $f_2 = f_0 - 2$, $f_0 \geq 8$, *and* $f_0 \neq 9$.

(iii) *If* $(f_0, f_2) = (f_0(Z), f_2(Z))$ *for a 3-zonotope* Z, *then* f_0 *and* f_1 *are both even integers,* $f_0 \leq 2f_2 - 4$, *and* $f_2 \leq f_0 - 2$.

For 4-polytopes, much less is known.

THEOREM 15.5.12

Flag f*-vectors* (f_0, f_1, f_2, f_{02}) *of 4-polytopes satisfy the following inequalities.*

(i) $f_{02} \geq 3f_2$

(ii) $f_{02} \geq 3f_1$

(iii) $f_{02} + f_1 + 10 \geq 3f_2 + 4f_0$

(iv) $6f_1 \geq 6f_0 + f_{02}$

(v) $f_0 \geq 5$

(vi) $f_0 + f_2 \geq f_1 + 5$

(vii) $2(f_{02} - 3f_2) \leq \binom{f_0}{2}$

(viii) $2(f_{02} - 3f_1) \leq \binom{f_2 - f_1 + f_0}{2}$

(ix) $f_{02} - 4f_2 + 3f_1 - 2f_0 \leq \binom{f_0}{2}$

(x) $f_{02} + f_2 - 2f_1 - 2f_0 \leq \binom{f_2 - f_1 + f_0}{2}$.

It is not known, for example, whether (i)–(vi) give all linear inequalities holding for flag f-vectors of 4-polytopes.

COMMENTS

It is thought that the best route to an eventual characterization of f-vectors of general polytopes lies in an understanding of their flag f-vectors. The latter inherit many of the algebraic properties of f-vectors of simplicial polytopes that led to their characterization.

The relations in Theorem 15.5.1 hold more generally for the case of enumeration of chains in Eulerian posets; see the article by Stanley in [BMSW94]. The relations in Theorem 15.5.1(iii) are proved in [BB85]. An expression for the toric h in terms of the flag f, as well as a discussion of the convolution product (originally due to Kalai), can be found in the article by Bayer in [BMSW94]. The article by Kalai in the same volume contains an extensive discussion of g-vectors for both simplicial and general polytopes. The form of the cubical Dehn-Sommerville equations given in Theorem 15.5.2 appeared in [Adi96].

Theorem 15.5.4 holds as well for certain shellable spheres (see [Sta96, Section III.4]). Theorem 15.5.3 can also be found in [Sta96, Theorem III.4.4] (where h_S is denoted $\beta(S)$). Theorem 15.5.5 appears in [Sta87b]. Theorem 15.5.7 is due to Blind and Blind [BB90]. There is a notion of convolution product of flag f

numbers that can be used to produce new linear inequalities from given ones; see [BL93, Section 3.10].

Note that Theorem 15.5.6, due to Braden and MacPherson [BM97], gives a connection between the g-vector of a polytope P and that of one of its faces. This is an example of a "monotonicity theorem" related to face numbers. For similar theorems relating h-vectors of subcomplexes and subdivisions of a simplicial complex Δ, see Sections III.9–10 of [Sta96] and the references given there.

For the fact that the flag f-vector of a zonotope or arrangement (or, more generally, of an oriented matroid) depends only on the underlying matroid, see [BLS$^+$93, Cor. 4.6.3]. That the only linear relations satisfied by zonotopes are the generalized Dehn-Sommerville equations of Theorem 15.5.1(iii) is proved in [BER]. Theorem 15.5.10 is due to Varchenko for the case $k = 2$ (see [BLS$^+$93, Proposition 4.6.9]) and to Liu [Liu95, Theorem 4.8.2] in the form given here.

Theorem 15.5.11(i) can be found in [Grü67, Section 10.3]; 15.5.11(ii) appears in dual form (for 4-valent 3-polytopes) in [Bar83]; 15.5.11(iii) can be derived using the methods of [Grü67, Section 18.2] (see also [BER]). Theorem 15.5.12 can be found in [Bay87].

15.6 OPEN PROBLEMS

PROBLEM 15.6.1

Characterize the f-vectors of triangulations of the $(d-1)$-sphere. It has been conjectured that the conditions of the g-theorem provide the answer.

PROBLEM 15.6.2

Characterize the f-vectors of triangulations of the d-ball.

PROBLEM 15.6.3

Characterize the f-vectors of triangulations of the d-torus. It is known that $f(2\text{-torus}) = \{(n, 3n, 2n) \mid n \geq 7\}$, but the question is open for $d \geq 3$.

PROBLEM 15.6.4

Characterize the f-vectors of d-polytopes. The answer is known for $d \leq 3$ (Theorem 15.5.11(i)), but for $d \geq 4$ there is not even a conjectured answer.

PROBLEM 15.6.5 *I. Bárány*

Does there exist a constant $c_d > 0$ such that $f_i \geq c_d \cdot \min\{f_0, f_{d-1}\}$ for all d-polytopes and all i? Will $c_d = 1$ do?

PROBLEM 15.6.6

Characterize the f-vectors of centrally symmetric d-polytopes. The question is open in the simplicial as well as in the general case. Even an upper bound conjecture in the simplicial and centrally symmetric case is missing.

PROBLEM 15.6.7 *Conjecture of G. Kalai*

The total number of faces (counting P but not $\emptyset$) of a centrally symmetric convex

d-polytope P is $\geq 3^d$. (Verified in the simplicial case as a consequence of Theorem 15.3.8.*)*

PROBLEM 15.6.8

*The **clique complex** of a graph is the collection of vertex sets of all its cliques (complete induced subgraphs). Characterize the f-vectors of clique complexes.*

PROBLEM 15.6.9 *G. Kalai*

Is the f-vector of an $(r-1)$-dimensional clique complex the f-vector of an r-colorable complex?

PROBLEM 15.6.10 *Conjecture of Charney and Davis* [Sta96, p. 100]

Let $(g_0, \ldots, g_k)$ be the g-vector of a clique complex homeomorphic to the sphere S^{2k-1}. Then $g_k - g_{k-1} + \ldots + (-1)^k g_0 \geq 0$.

PROBLEM 15.6.11 *Conjecture of Stanley* [Sta96, p. 102]

*Every coefficient ϕ_w of the **cd**-index of a sphere is nonnegative.*

PROBLEM 15.6.12 *Adin* [Adi96] (The case $i = 1$ is implied by Theorem 15.5.7.)

The generalized lower bound conjecture for cubical d-polytopes and d-spheres: $g_i^c \geq 0$ for $i \leq \lfloor d/2 \rfloor$. This has been shown to be the best possible set of linear inequalities for cubical d-spheres [BBC]. *More generally, characterize f-vectors of cubical polytopes.*

PROBLEM 15.6.13

Characterize the flag f-vectors of polytopes and of zonotopes. In particular, determine a complete set of linear inequalities holding for flag f-vectors of polytopes and of zonotopes.

PROBLEM 15.6.14

Characterize (toric) h-vectors of general polytopes.

15.7 SOURCES AND RELATED MATERIAL

FURTHER READING

Surveys of f-vector theory are given in [BL93, Bjö87, BK89, KK95, Sta85]. Books treating f-vectors (among other things) include [And87, BMSW94, Grü67, MS71, Sta96, Zie95].

RELATED CHAPTERS

Chapter 6: Oriented matroids
Chapter 13: Basic properties of convex polytopes
Chapter 14: Subdivisions and triangulations of polytopes

REFERENCES

[Adi96] R.M. Adin. A new cubical h-vector. *Discrete Math.*, 157:3–14, 1996.

[And87] I. Anderson. *Combinatorics of Finite Sets.* Clarendon Press, Oxford, 1987.

[BBC] E.K. Babson, L.J. Billera, and C. Chan. Neighborly cubical spheres and a cubical lower bound conjecture. *Israel J. Math.*, to appear.

[Bar83] D.W. Barnette. *Map Coloring, Polyhedra, and the Four Color Theorem.* Number 8 of *Dolciani Math. Exp.*, Math. Assoc. America, Washington, 1983.

[Bar92] A.I. Barvinok. On equivariant generalization of Dehn-Sommerville equation. *Europ. J. Combin.*, 13:419–428, 1992.

[Bay87] M.M. Bayer. The extended f-vectors of 4-polytopes. *J. Combin. Theory. Ser. A*, 44:141–151, 1987.

[Bay88] M.M. Bayer. *Barycentric subdivisions. Pacific J. Math.*, 135:1–16, 1988.

[BB85] M.M. Bayer and L.J. Billera. Generalized Dehn-Sommerville relations for polytopes, spheres and Eulerian partially ordered sets. *Invent. Math.*, 79:143–157, 1985.

[BL93] M.M. Bayer and C.W. Lee. Combinatorial aspects of convex polytopes. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 485–534. North-Holland, Amsterdam, 1993.

[BER] L.J. Billera, R. Ehrenborg, and M. Readdy. The *cd*-index of zonotopes and arrangements. In *Mathematical Essays in Honor of Gian-Carlo Rota*, Birkhäuser, Boston, to appear.

[BL81] L.J. Billera and C.W. Lee. A proof of the sufficiency of McMullen's conditions for f-vectors of simplicial polytopes. *J. Combin. Theory Ser. A*, 31:237–255, 1981.

[BMSW94] T. Bisztriczky, P. McMullen, R. Schneider, and A. Ivić Weiss, editors. *Polytopes: Abstract, Convex and Computational.* Volume 440 of *NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci.* Kluwer, Dordrecht, 1994.

[Bjö87] A. Björner. Face numbers of complexes and polytopes. In *Proc. Internat. Cong. Math., Berkeley, 1986*, pages 1408–1418. Amer. Math. Soc., Providence, 1987.

[Bjö96] A. Björner. Nonpure shellability, f-vectors, subspace arrangements and complexity. In L.J. Billera, C. Greene, R. Simion, and R. Stanley, editors, *Formal Power Series and Algebraic Combinatorics, DIMACS Ser. in Discrete Math. and Theor. Comput. Sci.*, pages 25–53. Amer. Math. Soc., Providence, 1996.

[BK88] A. Björner and G. Kalai. An extended Euler-Poincaré theorem. *Acta Math.*, 161:279–303, 1988.

[BK89] A. Björner and G. Kalai. On f-vectors and homology. In G. Bloom, R. Graham, and J. Malkevitch, editors, *Combinatorial Mathematics: Proc. 3rd Internat. Conf., New York, 1985*, volume 555 of *Ann. New York Acad. Sci.*, pages 63–80. New York Acad. Sci., 1989.

[BK91] A. Björner and G. Kalai. Extended Euler-Poincaré relations for cell complexes. In P. Gritzmann and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics—The Victor Klee Festschrift*, pages 81–89, volume 4 of *DIMACS Series in Discrete Math. and Theor. Comput. Sci.* Amer. Math. Soc., Providence, 1991.

[BLS+93] A. Björner, M. LasVergnas, B. Sturmfels, N. White, and G.M. Ziegler. *Oriented Matroids.* Volume 46 of *Encyclopedia Math. Appl.*, Cambridge University Press, 1993.

[BB90] G. Blind and R. Blind. Convex polytopes without triangular faces. *Israel Jour. Math.*, 71:129–134, 1990.

[BM97] T.C. Braden and R. MacPherson. Intersection homology of toric varieties and a conjecture of Kalai. Preprint, 1997.

[FFK88] P. Frankl, Z. Füredi, and G. Kalai. Shadows of colored complexes. *Math. Scand.*, 63:169–178, 1988.

[FK96] E. Friedgut and G. Kalai. Every monotone graph property has a sharp threshold. *Proc. Amer. Math. Soc.*, 124:2993–3002, 1996.

[Grü67] B. Grünbaum. *Convex Polytopes.* Interscience, London, 1967; revised edition (V. Klee and P. Kleinschmidt, editors), *Grad. Texts in Math.*, Springer-Verlag, in preparation.

[Hib89] T. Hibi. What can be said about pure O-sequences? *J. Combin. Theory Ser. A*, 50:319–322, 1989.

[Kal84] G. Kalai. A characterization of f-vectors of families of convex sets in $\mathbb{R}^d$. Part I: Necessity of Eckhoff's conditions. *Israel J. Math.*, 48:175–195, 1984.

[Kal86] G. Kalai. A characterization of f-vectors of families of convex sets in $\mathbb{R}^d$. Part II: Sufficiency of Eckhoff's conditions. *J. Combin. Theory Ser. A*, 41:167–188, 1986.

[Kal87] G. Kalai. Rigidity and the lower bound theorem 1. *Invent. Math.*, 88:125–151, 1987.

[Kle64] V. Klee. A combinatorial analogue of Poincaré's duality theorem. *Canad. J. Math.*, 16:517–531, 1964.

[KK95] V. Klee and P. Kleinschmidt. Convex polytopes and related complexes. In R. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, pages 875–917. North-Holland, Amsterdam, 1995.

[Lin71] B. Lindström. The optimal number of faces in cubical complexes. *Ark. Mat.*, 8:245–257, 1971.

[Liu95] N. Liu. *Algebraic and Combinatorial Methods for Face Enumeration in Polytopes.* Ph.D. Thesis, Cornell Univ., Ithaca, 1995.

[LW69] A.T. Lundell and S. Weingram. *The Topology of CW Complexes.* Van Nostrand, New York, 1969.

[McM93] P. McMullen. On simple polytopes. *Invent. Math.*, 113:419–444, 1993.

[MS71] P. McMullen and G.C. Shephard. *Convex Polytopes and the Upper Bound Conjecture.* Volume 3 of *London Math. Soc. Lecture Note Ser.*, Cambridge University Press, 1971.

[Mun84] J.R. Munkres. *Elements of Algebraic Topology.* Addison-Wesley, Reading, 1984.

[Nov] I. Novik. Upper bound theorems for simplicial manifolds. *Israel J. Math.*, to appear.

[Spa66] E.H. Spanier. *Algebraic Topology.* McGraw-Hill, New York, 1966.

[Sta80] R.P. Stanley. The number of faces of simplicial convex polytopes. *Adv. Math.*, 35:236–238, 1980.

[Sta85] R.P. Stanley. The number of faces of simplicial polytopes and spheres. In J.E. Goodman, E. Lutwak, J. Malkevitch, and R. Pollack, editors, *Discrete Geometry and Convexity*, volume 440 of *Ann. New York Acad. Sci.*, pages 212–223. New York Acad. Sci., 1985.

[Sta86] R.P. Stanley. *Enumerative Combinatorics*, Volume I. Wadsworth, Monterey, 1986.

[Sta87a] R.P. Stanley. On the number of faces of centrally-symmetric simplicial polytopes. *Graphs Combin.*, 3:55–66, 1987.

[Sta87b] R.P. Stanley. Generalized h-vectors, intersection cohomology of toric varieties, and related results. In M. Nagata and H. Matsumura, editors, *Commutative Algebra and Combinatorics*, volume 11 of *Adv. Stud. Pure Math.*, pages 187–213. Kinokuniya, Tokyo and North-Holland, Amsterdam/New York, 1987.

[Sta96] R.P. Stanley. *Combinatorics and Commutative Algebra*, 2nd Ed. Volume 41 of *Progr. Math.*, Birkhäuser, Boston, 1996.

[Zie95] G.M. Ziegler. *Lectures on Polytopes.* Volume 152 of *Graduate Texts in Math.*, Springer-Verlag, New York, 1995.

16 SYMMETRY OF POLYTOPES AND POLYHEDRA

Egon Schulte

INTRODUCTION

Symmetry of geometric figures is among the most frequently recurring themes in science. The present chapter discusses symmetry of discrete geometric structures, namely of polytopes, polyhedra, and related polytope-like figures. These structures have an outstanding history of study unmatched by almost any other geometric object. The most prominent symmetric figures, the regular solids, occur from very early times and are attributed to Plato (427-347 B.C.E.). Since then, many changes in point of view have occurred about these figures and their symmetry. With the arrival of group theory in the 19th century, many of the early approaches were consolidated and the foundations were laid for a more rigorous development of the theory. In this vein, Schläfli (1814-1895) extended the concept of regular polytopes and tessellations to higher dimensional spaces and explored their symmetry groups as reflection groups.

Today we owe much of our present understanding of symmetry in geometric figures (in a broad sense) to the influential work of Coxeter, which provided a unified approach to regularity of figures based on a powerful interplay of geometry and algebra [Cox73]. Coxeter's work also greatly influenced modern developments in this area, which received a further impetus from Grünbaum [Grü77a]. In the past 20 years, the study of regular figures has been extended in several directions that are all centered around an abstract combinatorial polytope theory and a combinatorial notion of regularity [MS].

History teaches us that the subject has shown an enormous potential for revival. One explanation for this is the appearance of polyhedral structures in many contexts that have little apparent relation to regularity, such as the occurrence of many of them in nature as crystals (Fejes Tòth [Fej64]).

16.1 REGULAR CONVEX POLYTOPES AND REGULAR EUCLIDEAN TESSELATIONS IN $\mathbb{E}^d$

Perhaps the most important (but certainly the most investigated) symmetric polytopes are the regular convex polytopes in Euclidean spaces. See [Grü67] for general properties of convex polytopes, or Chapter 13 in this Handbook. The most comprehensive text on regular convex polytopes and regular tessellations is [Cox73].

GLOSSARY

Convex d-polytope: The intersection P of finitely many closed halfspaces in a Euclidean space, which is bounded and d-dimensional.

Face: The empty set and P itself are ***improper faces*** of dimension -1 and d, respectively. A ***proper face*** F of P is the (nonempty) intersection of P with a supporting hyperplane of P. (Recall that a hyperplane H ***supports*** P at F if $P \cap H = F$ and P lies in one of the closed halfspaces bounded by H.)

Vertex, edge, i-face, facet: Face of P of dimension 0, 1, i, or $d-1$, respectively.

Vertex figure: A vertex figure of P at a vertex x is the intersection of P with a hyperplane H that strictly separates x from the other vertices of P. (If P is regular, one can take H to be the hyperplane passing through the midpoints of the edges that contain x.)

Face lattice of a polytope: The set $\mathcal{F}(P)$ of all faces of P, ordered by inclusion. As a partially ordered set, this is a ranked lattice. Also, $\mathcal{F}(P) \setminus \{P\}$ is called the ***boundary complex*** of P.

Flag: A maximal totally ordered subset of $\mathcal{F}(P)$.

Isomorphism of polytopes: A bijection $\varphi : \mathcal{F}(P) \mapsto \mathcal{F}(Q)$ between the face lattices of two polytopes P and Q such that φ preserves incidence in both directions; that is, $F \subseteq G$ in $\mathcal{F}(P)$ if and only if $F\varphi \subseteq G\varphi$ in $\mathcal{F}(Q)$. If such an isomorphism exists, P and Q are ***isomorphic***.

Dual of a polytope: A convex d-polytope Q is the dual of P if there is a ***duality*** $\varphi : \mathcal{F}(P) \mapsto \mathcal{F}(Q)$; that is, a bijection reversing incidences in both directions, meaning that $F \subseteq G$ in $\mathcal{F}(P)$ if and only if $F\varphi \supseteq G\varphi$ in $\mathcal{F}(Q)$. A polytope has many duals but any two are isomorphic, justifying speaking of "the dual". (If P is regular, one can take Q to be the convex hull of the facet centers of P, or a rescaled copy of this.)

Self-dual polytope: A polytope that is isomorphic to its dual.

Symmetry: A Euclidean isometry of the ambient space (affine hull of P) that maps P to itself.

Symmetry group of a polytope: The group $S(P)$ of all symmetries of P.

Regular polytope: A polytope whose symmetry group $S(P)$ is transitive on the flags.

Schläfli symbol: A symbol $\{p_1, \ldots, p_{d-1}\}$ that encodes the local structure of a regular polytope. For each $i = 1, \ldots, d-1$, if F is any $(i+1)$-face of P, then p_i is the number of i-faces of F that contain a given $(i-2)$-face of F.

Tessellation: A family T of convex d-polytopes in Euclidean d-space $\mathbb{E}^d$, called the ***tiles*** of T, such that the union of all tiles of T is $\mathbb{E}^d$, and any two distinct tiles do not have interior points in common. All tessellations are assumed to be ***locally finite***, meaning that each point of $\mathbb{E}^d$ has a neighborhood meeting only finitely many tiles, and ***face-to-face***, meaning that the intersection of any two tiles is a face of each (possibly the empty face); see Chapter 3. The concept of a tessellation extends to other spaces including spherical space (Euclidean unit sphere) and hyperbolic space.

Face lattice of a tessellation: A ***proper face*** of T is a nonempty face of a tile of T. ***Improper faces*** of T are the empty set and the whole space $\mathbb{E}^d$. The set $\mathcal{F}(T)$ of all (proper and improper) faces is a ranked lattice called the face lattice of T. Concepts like isomorphism and duality carry over from polytopes.

Symmetry group of a tessellation: The group $S(T)$ of all symmetries of T; that is, of all isometries of the ambient (spherical, Euclidean, or hyperbolic)

space that preserve T. Concepts like regularity and Schläfli symbol carry over from polytopes.

Apeirogon: A tessellation of the real line with closed intervals of the same length. This can also be regarded as an infinite polygon whose edges are given by the intervals.

ENUMERATION AND CONSTRUCTION

The convex regular polytopes P in $\mathbb{E}^d$ are known for each d. If $d = 1$, P is a line segment and $|S(P)| = 2$. In all other cases, up to similarity, P can be uniquely described by its Schläfli symbol $\{p_1, \ldots, p_{d-1}\}$. For convenience one writes $P = \{p_1, \ldots, p_{d-1}\}$. If $d = 2$, P is a convex regular p-gon for some $p \geq 3$, and $P = \{p\}$; also, $S(P) = D_p$, the dihedral group of order $2p$.

The regular polytopes P with $d \geq 3$ are summarized in Table 16.1.1, which also includes the numbers f_0 and f_{d-1} of vertices and facets, the order of $S(P)$, and the diagram notation for the group (following [Hum90]). Here and below, p^n will be used to denote a string of n consecutive p's. For $d = 3$ the list consists of the five Platonic solids (Figure 16.1.1). The regular d-simplex, d-cube, and d-cross-polytope occur in each dimension d. (These are line segments if $d = 1$, and triangles or squares if $d = 2$.) The dimensions 3 and 4 are exceptional in that there are 2 respectively 3 more regular polytopes. If $d \geq 3$, the facets and vertex figures of $\{p_1, \ldots, p_{d-1}\}$ are the regular $(d-1)$-polytopes $\{p_1, \ldots, p_{d-2}\}$ and $\{p_2, \ldots, p_{d-1}\}$, respectively, whose Schläfli symbols, when superposed, give the original. The dual of $\{p_1, \ldots, p_{d-1}\}$ is $\{p_{d-1}, \ldots, p_1\}$. Self-duality occurs only for $\{3^{d-1}\}$, $\{p\}$, and $\{3, 4, 3\}$. Except for $\{3^{d-1}\}$ and $\{p\}$ with p odd, all regular polytopes are centrally symmetric.

TABLE 16.1.1 The convex regular polytopes in $\mathbb{E}^d$ $(d \geq 3)$.

DIMENSION	NAME	SCHLÄFLI SYMBOL	f_0	f_{d-1}	$\lvert S(P)\rvert$	DIAGRAM
$d \geq 3$	d-simplex	$\{3^{d-1}\}$	d+1	d+1	(d+1)!	A_d
	d-cross-polytope	$\{3^{d-2}, 4\}$	$2d$	2^d	$2^d d!$	B_d (or C_d)
	d-cube	$\{4, 3^{d-2}\}$	2^d	$2d$	$2^d d!$	B_d (or C_d)
$d = 3$	icosahedron	$\{3, 5\}$	12	20	120	H_3
	dodecahedron	$\{5, 3\}$	20	12	120	H_3
$d = 4$	24-cell	$\{3, 4, 3\}$	24	24	1152	F_4
	600-cell	$\{3, 3, 5\}$	120	600	14400	H_4
	120-cell	$\{5, 3, 3\}$	600	120	14400	H_4

The regular tessellations T in $\mathbb{E}^d$ are also known. If $d = 1$, T is an apeirogon and $S(T)$ is the infinite dihedral group. For $d \geq 2$ see the list in Table 16.1.2. The first $d-1$ entries in $\{p_1, \ldots, p_d\}$ give the Schläfli symbol for the tiles of T, the last $d-1$ that for the vertex figures. (A vertex figure at a vertex x is the convex hull of the midpoints of the edges emanating from x.) The cubical tessellation occurs for each d, while for $d = 2$ and $d = 4$ there is a dual pair of exceptional tessellations.

As vertices of the plane polygon $\{p\}$ we can take the points corresponding to

FIGURE 16.1.1
The five Platonic solids.

TABLE 16.1.2 The regular tessellations in $\mathbb{E}^d$ ($d \geq 2$).

DIMENSION	SCHLÄFLI SYMBOL	TILES	VERTEX-FIGURES
$d \geq 2$	$\{4, 3^{d-2}, 4\}$	d-cubes	d-cross-polytopes
$d = 2$	$\{3, 6\}$	triangles	hexagons
	$\{6, 3\}$	hexagons	triangles
$d = 4$	$\{3, 3, 4, 3\}$	4-cross-polytopes	24-cells
	$\{3, 4, 3, 3\}$	24-cells	4-cross-polytopes

the p-th roots of unity. The d-simplex can be defined as the convex hull of the $d+1$ points in $\mathbb{E}^{d+1}$ corresponding to the permutations of $(1, 0, \ldots, 0)$. As vertices of the d-cross-polytope in $\mathbb{E}^d$ choose the $2d$ permutations of $(\pm 1, 0, \ldots, 0)$, and for the d-cube take the 2^d points $(\pm 1, \ldots, \pm 1)$. The midpoints of the edges of a 4-cross-polytope are the 24 vertices of a regular 24-cell. The coordinates for the remaining regular polytopes are more complicated [Cox73, pp. 52,157].

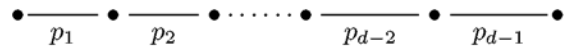
For the cubical tessellation $\{4, 3^{d-2}, 4\}$ take the vertex set to be $\mathbb{Z}^d$ (giving the square tessellation if $d = 2$). For the triangle tessellation $\{3, 6\}$ choose as vertices the integral linear combinations of two unit vectors inclined at $\pi/3$. Locating the face centers gives the vertices of the hexagonal tessellation $\{6, 3\}$. For $\{3, 3, 4, 3\}$ in $\mathbb{E}^4$ take the alternating vertices of the cubical tessellation; that is, the integral points with an even coordinate sum. Its dual $\{3, 4, 3, 3\}$ (with 24-cells as tiles) has the vertices at the centers of the tiles of $\{3, 3, 4, 3\}$.

SYMMETRY GROUPS

For a regular d-polytope P in $\mathbb{E}^d$, pick a fixed (***base***) flag Φ, and consider the maximal simplex C (***chamber***) in the barycentric subdivision (***chamber complex***) of P whose vertices are the centers of the nonempty faces in Φ. Then C is a fundamental region for $S(P)$ in P and $S(P)$ is generated by the reflections $R_0, \ldots, R_{d-1}$ in the walls of C that contain the center of P, where R_i is the reflection in the wall opposite to the vertex of C corresponding to the i-face in Φ. If $P = \{p_1, \ldots, p_{d-1}\}$, then

$$\begin{cases} R_i^2 = (R_j R_k)^2 = 1 & (0 \leq i, j, k \leq d-1,\ |j-k| \geq 2) \\ (R_{i-1} R_i)^{p_i} = 1 & (1 \leq i \leq d-1) \end{cases}$$

is a presentation for $S(P)$ in terms of these generators. In particular, $S(P)$ is a finite (spherical) Coxeter group with string diagram



(see Section 16.6).

If T is a regular tessellation of $\mathbb{E}^d$, pick Φ and C as before. Now $S(P)$ is generated by the $d+1$ reflections in all walls of C giving $R_0, \ldots, R_d$ (as above). The presentation for $S(T)$ carries over, but now $S(T)$ is an infinite (Euclidean) Coxeter group.

16.2 REGULAR STAR-POLYTOPES

The regular star-polyhedra and star-polytopes are obtained by allowing the faces or vertex figures to be *starry* (star-like). This leads to very beautiful figures that are closely related to the regular convex polytopes. See Coxeter [Cox73] for a comprehensive account; see also McMullen and Schulte [MS]. In defining star-polytopes, we shall combine the approach of [Cox73] and McMullen [McM68] and introduce them via the associated starry polytope-configuration.

GLOSSARY

d-polytope-configuration: A finite family Π of affine subspaces, called ***elements***, of Euclidean d-space $\mathbb{E}^d$, ordered by inclusion, such that the following conditions are satisfied. Π contains the empty set $\emptyset$ and $\mathbb{E}^d$ as (*improper*) elements. The dimensions of the other (*proper*) elements can take the values $0, 1, \ldots, d-1$, and the affine hull of their union is $\mathbb{E}^d$. As a partially ordered set, Π is a ranked lattice. For $F, G \in \Pi$ with $F \subseteq G$ call $G/F := \{H \in \Pi | F \subseteq H \subseteq G\}$ the ***subconfiguration*** of Π defined by F and G; this will itself be a $(dim(G) - dim(F) - 1)$-polytope-configuration. As further conditions, each G/F contains at least 2 proper elements if $dim(G) - dim(F) = 2$, and as a partially ordered set, each G/F (including Π itself) is connected if $dim(G) - dim(F) \geq 3$. (See the definition of an abstract polytope in Section 16.8.) It can be proved that in $\mathbb{E}^d$ every Π satisfies the stronger condition that each G/F contains exactly 2 proper elements if $dim(G) - dim(F) = 2$.

Regular polytope-configuration: A polytope-configuration Π whose symmetry group $S(\Pi)$ is flag-transitive. (A flag is a maximal totally ordered subset of Π.)

Regular star-polygon: For positive integers n and k with $(n, k) = 1$ and $1 < k < \frac{n}{2}$, up to similarity the regular star-polygon $\{\frac{n}{k}\}$ is the connected plane polygon whose consecutive vertices are $(\cos(\frac{2\pi kj}{n}), \sin(\frac{2\pi kj}{n}))$ for $j = 0, 1, \ldots, n-1$. If $k = 1$, the same plane polygon is a (nonstarry) convex n-gon with Schläfli symbol $\{n\}$ $(= \{\frac{n}{1}\})$. With each regular (convex or star-) polygon $\{\frac{n}{k}\}$ is associated a regular 2-polytope-configuration obtained by replacing each edge by its affine hull.

Star-polytope-configuration: A d-polytope-configuration Π is ***nonstarry*** if it is the family of affine hulls of the faces of a convex d-polytope. It is ***starry***, or a ***star-polytope-configuration***, if it is not nonstarry. For instance, among the 2-polytope-configurations that are associated with a regular (convex or star-) polygon $\{\frac{n}{k}\}$ for a given n, the one with $k = 1$ is nonstarry and those for $k > 1$ are starry. In the first case the corresponding n-gon is convex, and in the second case it is genuinely star-like. In general, the starry polytope configurations are those that belong to genuinely star-like polytopes (that is, star-polytopes).

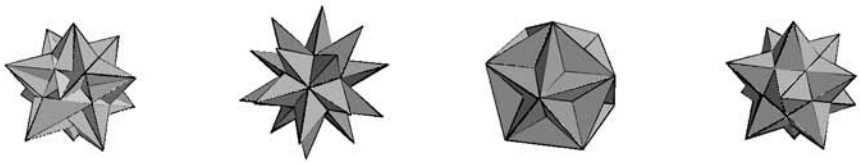
Regular star-polytope: If $d = 2$, a regular star-polytope is a regular star-polygon. Defined inductively, if $d \geq 3$, a regular d-star-polytope P is a finite family of regular convex $(d-1)$-polytopes or regular $(d-1)$-star-polytopes such that the family consisting of their affine hulls as well as the affine hulls of their "faces" is a regular d-star-polytope-configuration $\Pi = \Pi(P)$. Here, the faces of the polytopes can be defined in such a way that they correspond to the elements in the associated polytope-configuration. The symmetry groups of P and Π are the same.

ENUMERATION AND CONSTRUCTION

Regular star-polytopes P can only exist for $d = 2$, 3, or 4. As regular convex polytopes, they are also uniquely determined by the Schläfli symbol $\{p_1, \ldots, p_{d-1}\}$, but now at least one entry is not integral. Again the symbols for the facets and vertex figures, when superposed, give the original. If $d = 2$, $P = \{\frac{n}{k}\}$ for some k with $(n, k) = 1$ and $1 < k < \frac{n}{2}$, and $S(P) = D_n$. For $d = 3$ and 4 the star-polytopes are listed in Table 16.2.1 together with the numbers f_0 and f_{d-1} of vertices and facets, respectively.

TABLE 16.2.1 The regular star-polytopes in $\mathbb{E}^d$ $(d \geq 3)$.

DIMENSION	SCHLÄFLI SYMBOL	f_0	f_{d-1}
$d = 3$	$\{3, \frac{5}{2}\}$	12	20
	$\{\frac{5}{2}, 3\}$	20	12
	$\{5, \frac{5}{2}\}$	12	12
	$\{\frac{5}{2}, 5\}$	12	12
$d = 4$	$\{3, 3, \frac{5}{2}\}$	120	600
	$\{\frac{5}{2}, 3, 3\}$	600	120
	$\{3, 5, \frac{5}{2}\}$	120	120
	$\{\frac{5}{2}, 5, 3\}$	120	120
	$\{3, \frac{5}{2}, 5\}$	120	120
	$\{5, \frac{5}{2}, 3\}$	120	120
	$\{5, 3, \frac{5}{2}\}$	120	120
	$\{\frac{5}{2}, 3, 5\}$	120	120
	$\{5, \frac{5}{2}, 5\}$	120	120
	$\{\frac{5}{2}, 5, \frac{5}{2}\}$	120	120



Great icosahedron | Great stellated dodecahedron | Great dodecahedron | Small stellated dodecahedron

FIGURE 16.2.1 *The four Kepler-Poinsot polyhedra.*

Every regular d-star-polytope has the same vertices and symmetry group as a regular convex d-polytope. The four regular star-polyhedra (3-star-polytopes) are also known as the ***Kepler-Poinsot polyhedra*** (Figure 16.2.1). They can be constructed from the icosahedron $\{3,5\}$ or dodecahedon $\{5,3\}$ by two kinds of operations, ***stellation*** or ***faceting*** [Cox73]. Loosely speaking, in the former operation one extends the faces of a polyhedron symmetrically until they again form a polyhedron, while in the latter operation the vertices of a polyhedron are redistributed in classes that are then the vertex sets for the faces of a new polyhedron. Regarded as regular maps on surfaces (Section 16.3), the polyhedra $\{3,\frac{5}{2}\}$ (***great icosahedron***) and $\{\frac{5}{2},3\}$ (***great stellated dodecahedron***) are of genus 0, while $\{5,\frac{5}{2}\}$ (***great dodecahedron***) and $\{\frac{5}{2},5\}$ (***small stellated dodecahedron***) are of genus 4.

The ten regular star-polytopes in $\mathbb{E}^4$ all have the same vertices and symmetry groups as the 600-cell $\{3,3,5\}$ or 120-cell $\{5,3,3\}$ and can be derived from these by 4-dimensional stellation or faceting operations [Cox73, McM68]. See also [Cox93] for their names, which describe the various relationships among the polytopes.

The dual of $\{p_1,\ldots,p_{d-1}\}$ (which is obtained by dualizing the associated star-polytope-configuration using reciprocation with respect to a sphere) is $\{p_{d-1},\ldots,p_1\}$. Regarded as abstract polytopes (Section 16.8), the star-polytopes $\{p_1,\ldots,p_{d-1}\}$ and $\{q_1,\ldots,q_{d-1}\}$ are isomorphic if and only if the symbol $\{q_1,\ldots,q_{d-1}\}$ is obtained from $\{p_1,\ldots,p_{d-1}\}$ by replacing each entry 5 by $\frac{5}{2}$ and each $\frac{5}{2}$ by 5.

16.3 REGULAR SKEW POLYHEDRA

Regular skew polyhedra are finite or infinite polyhedra whose vertex figures are skew (antiprismatic) polygons. The standard reference is Coxeter [Cox68]. Topologically, these polyhedra are regular maps on surfaces. For general properties of regular maps see Coxeter and Moser [CM80], McMullen and Schulte [MS], or Chapter 18 of this Handbook.

GLOSSARY

(Right) prism, antiprism (with regular bases): A convex 3-polytope whose vertices are contained in two parallel planes and whose set of 2-faces consists of the two ***bases*** (contained in the parallel planes) and the 2-faces in the ***mantle*** that connects the bases. The bases are congruent regular polygons. For a (right) prism, each base is a translate of the other by a vector perpendicular to its affine hull, and the mantle 2-faces are rectangles. For a (right) antiprism, each base is a translate of a reciprocal (dual) of the other by a vector perpendicular to its affine hull, and the mantle 2-faces are isosceles triangles. (The prism or antiprism is ***semi-regular*** if its mantle 2-faces are squares or equilateral triangles, respectively; see Section 16.5.)

Map on a surface: A decomposition (tessellation) P of a closed surface S into nonoverlapping simply connected regions, the 2-***faces*** of P, by arcs, the ***edges*** of P, joining pairs of points, the ***vertices*** of P, such that two conditions are satisfied. First, each edge belongs to exactly two 2-faces. Second, if two distinct edges intersect, they meet in one vertex or in two vertices.

Regular map: A map P on S whose combinatorial automorphism group $A(P)$ is transitive on the flags (incident triples consisting of a vertex, an edge, and a 2-face).

Polyhedron: A map P on a closed surface S embedded (without self-intersections) into a Euclidean space, such that two conditions are satisfied. Each 2-face of P is a convex plane polygon, and any two adjacent 2-faces do not lie in the same plane. See also the more general definition in the next section.

Skew polyhedron: A polyhedron P such that for at least one vertex x, the vertex figure of P at x is not a plane polygon; the ***vertex figure*** at x is the polygon whose vertices are the vertices of P adjacent to x and whose edges join consecutive vertices as one goes around x.

Regular polyhedron: A polyhedron P whose symmetry group $S(P)$ is flag-transitive. (For a regular skew polyhedron P in $\mathbb{E}^3$ or $\mathbb{E}^4$, each vertex figure must be a 3-dimensional antiprismatic polygon, meaning that it contains all edges of an antiprism that are not edges of a base. See also Section 16.4.)

ENUMERATION

In $\mathbb{E}^3$ all, and in $\mathbb{E}^4$ all finite, regular skew polyhedra are known [Cox68]. In these cases the polyhedron P is completely determined by the extended Schläfli symbol $\{p, q|r\}$, where the 2-faces of P are convex p-gons such that q meet at each vertex, and r is the number of edges in each edge path of P that leaves, at each vertex, exactly two 2-faces of P on the right. The group $S(P)$ is isomorphic to $A(P)$ and has the presentation

$$\rho_0{}^2 = \rho_1{}^2 = \rho_2{}^2 = (\rho_0\rho_1)^p = (\rho_1\rho_2)^q = (\rho_0\rho_2)^2 = (\rho_0\rho_1\rho_2\rho_1)^r = 1$$

(but the generators ρ_i are not all hyperplane reflections). The polyhedra $\{p, q|r\}$ and $\{q, p|r\}$ are duals, and the vertices of one can be obtained as the centers of the 2-faces of the other.

In $\mathbb{E}^3$ there are just three regular skew polyhedra: $\{4, 6|4\}$, $\{6, 4|4\}$, and $\{6, 6|3\}$. These are the (infinite) *Petrie-Coxeter polyhedra*. For example, $\{4, 6|4\}$ consists of half the square faces of the cubical tessellation $\{4, 3, 4\}$ in $\mathbb{E}^3$.

TABLE 16.3.1 The finite regular skew polyhedra in $\mathbb{E}^4$.

SCHLÄFLI SYMBOL	f_0	f_2	GROUP ORDER	GENUS
$\{4, 4\|r\}$	r^2	r^2	$8r^2$	1
$\{4, 6\|3\}$	20	30	240	6
$\{6, 4\|3\}$	30	20	240	6
$\{4, 8\|3\}$	144	288	2304	73
$\{8, 4\|3\}$	288	144	2304	73

The finite regular skew polyhedra in $\mathbb{E}^4$ (or equivalently, in spherical 3-space) are listed in Table 16.3.1. There is an infinite sequence of toroidal polyhedra as

well as two pairs of duals related to the (self-dual) 4-simplex $\{3,3,3\}$ and 24-cell $\{3,4,3\}$. For drawings of projections of these polyhedra into 3-space see [SW91]; Figure 16.3.1 represents $\{4,8|3\}$.

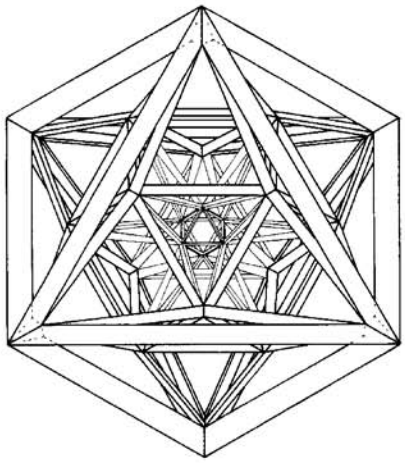


FIGURE 16.3.1
A projection of $\{4,8|3\}$ *into* $\mathbb{R}^3$.

These projections are examples of *combinatorially regular polyhedra* in ordinary 3-space; see Chapter 18 in this Handbook. For a discussion of regular skew polyhedra in hyperbolic 3-space, see [Gar67].

16.4 THE GRÜNBAUM-DRESS POLYHEDRA

A new impetus to the study of regular figures came from Grünbaum [Grü77b], who generalized the regular skew polyhedra by allowing skew polygons as faces as well as vertex figures. This restored the symmetry in the definition of polyhedra. For the classification of these "new" regular polyhedra in $\mathbb{E}^3$, see [Grü77b], [Dre85], and [MS]. The proper setting for this subject is, strictly speaking, in the context of realizations of abstract regular polytopes (see Section 16.8).

GLOSSARY

Polygon: A figure P in Euclidean space $\mathbb{E}^d$ consisting of a (finite or infinite) sequence of distinct points, called the ***vertices*** of P, joined in successive pairs, and closed cyclicly if finite, by line segments, called the ***edges*** of P, such that each compact set in $\mathbb{E}^d$ meets only finitely many edges.

Zigzag polygon: A (zigzag-shaped) infinite plane polygon P whose vertices alternately lie on two parallel lines and whose edges are all of the same length.

Antiprismatic polygon: A closed polygon P in 3-space whose vertices are alternately vertices of each of the two (regular convex) bases of a (right) antiprism Q (Section 16.3), such that the orthogonal projection of P onto the plane of a

base gives a regular star-polygon (Section 16.2). This star-polygon (and thus P) has twice as many vertices as each base, and is a convex polygon if and only if the edges of P are just those edges of Q that are not edges of a base.

Prismatic polygon: A closed polygon P in 3-space whose vertices are alternately vertices of each of the two (regular convex) bases of a (right) prism Q (Section 16.3), such that the orthogonal projection of P onto the plane of a base traverses twice a regular star-polygon in that plane (Section 16.2). Each base of Q (and thus the star-polygon) is assumed to have an odd number of vertices. The star-polygon is a convex polygon if and only if each edge of P is a diagonal in a rectangular 2-face in the mantle of Q.

Helical polygon: An infinite polygon in 3-space whose vertices lie on a helix given parametrically by $(a\cos\beta t, a\sin\beta t, bt)$, where $a, b \neq 0$ and $0 < \beta < \pi$, and are obtained as t ranges over the integers. Successive integers correspond to successive vertices.

Polyhedron: A (finite or infinite) family P of polygons in $\mathbb{E}^d$, called the 2-***faces*** of P, such that three conditions are satisfied. First, each edge of one of the 2-faces is an edge of exactly one other 2-face. Second, for any two edges F and F' of (2-faces of) P there exist chains $F = G_0, G_1, \ldots, G_n = F'$ of edges and $H_1, \ldots, H_n$ of 2-faces such that each H_i is incident with G_{i-1} and G_i. Third, each compact set in $\mathbb{E}^d$ meets only finitely many 2-faces.

Regular: A polygon or polyhedron P is regular if its symmetry group $S(P)$ is transitive on the flags.

ENUMERATION

For a systematic discussion of regular polygons in arbitrary Euclidean spaces see [Cox93] (or [MS]). In light of the geometric classification scheme for the new regular polyhedra in $\mathbb{E}^3$ proposed in [Grü77b], it is useful to classify the regular polygons in $\mathbb{E}^3$ into seven groups: convex polygons, plane star-polygons (Section 16.2), apeirogons (Section 16.2), zigzag polygons, antiprismatic polygons, prismatic polygons, and helical polygons. These correspond to the four kinds of isometries in $\mathbb{E}^3$: rotation, rotatory reflection (a reflection followed by a rotation in the reflection plane), glide reflection, and twist.

The 2-faces and vertex figures of a regular polyhedron P in $\mathbb{E}^3$ are regular polygons of the above kind. (The vertex figure at a vertex x is the polygon whose vertices are the vertices of P adjacent to x and whose edges join two such vertices y and z if and only if $\{y, x\}$ and $\{x, z\}$ are edges of a 2-face in P. For a regular P, this is a single polygon.) It is convenient to group the regular polyhedra in $\mathbb{E}^3$ into 8 classes. The first four are the traditional regular polyhedra: the five Platonic solids; the three planar tessellations; the four regular star-polyhedra (Kepler-Poinsot polyhedra); and the three infinite regular skew polyhedra (Petrie-Coxeter polyhedra). The four other classes and their polyhedra can be described as follows: the class of nine finite polyhedra with finite skew (antiprismatic) polygons as faces; the class of infinite polyhedra with finite skew (prismatic or antiprismatic) polygons as faces, which includes three infinite families as well as three individual polyhedra; the class of polyhedra with zigzag polygons as faces, which contains six infinite families; and the class of polyhedra with helical polygons as faces, which has three infinite families and six individual polyhedra. Each of these polyhedra can be described by a

generalized Schläfli symbol, which encodes the geometric structure of the polygonal faces and vertex figures. For more details see [Grü77b], [Dre85], [Joh], and [MS].

16.5 SEMI-REGULAR AND UNIFORM CONVEX POLYTOPES

The very stringent requirements in the definition of regularity of polytopes can be relaxed in many different ways, yielding a great variety of weaker regularity notions. We shall only consider polytopes and polyhedra that are convex. See Johnson [Joh] for a detailed discussion, or Martini [Mar93] for a survey.

GLOSSARY

Semi-regular: A convex d-polytope P is semi-regular if its facets are regular and its symmetry group $S(P)$ is transitive on the vertices of P.

Uniform: A convex polygon is uniform if it is regular. Recursively, if $d \geq 3$, a convex d-polytope P is uniform if its facets are uniform and its symmetry group $S(P)$ is transitive on the vertices of P.

Regular-faced: P is regular-faced if all its facets (and lower-dimensional faces) are regular.

ENUMERATION

Each regular polytope is semi-regular, and each semi-regular polytope is uniform. Also, by definition each uniform 3-polytope is semi-regular. For $d = 3$ the family of semi-regular (uniform) convex polyhedra consists of the Platonic solids, two infinite classes of prisms and antiprisms, as well as the thirteen polyhedra known as Archimedean solids [Fej64]. The seven semi-regular polyhedra whose symmetry group is edge-transitive are also called the ***quasi-regular*** polyhedra.

Besides the regular polytopes, there are only seven semi-regular polytopes in higher dimensions: three for $d = 4$, and one for each of $d = 5, 6, 7, 8$. However, there are many more uniform polytopes but a complete list is known only for $d = 4$ [Joh]. Except for the regular 4-polytopes and the prisms over uniform 3-polytopes, there are exactly 40 uniform 4-polytopes.

For $d = 3$ all, for $d = 4$ all save one, and for $d \geq 5$ many, uniform polytopes can be obtained by a method called ***Wythoff's construction***. This method proceeds from a finite Euclidean reflection group W in $\mathbb{E}^d$, or the even (rotation) subgroup W^+ of W, and constructs the polytopes as the convex hull of the orbit under W or W^+ of a point, the initial vertex, in the fundamental region of the group, which is a d-simplex (chamber) or the union of two d-simplices in the corresponding Coxeter complex of W, respectively; see Section 16.6.

The regular-faced polytopes have also been described for each dimension. In general, such a polytope can have different kinds of facets (and vertex figures). For $d = 3$ the complete list contains exactly 92 regular-faced convex polyhedra and includes all semi-regular polyhedra. For each $d \geq 5$, there are only two regular-faced d-polytopes that are not semi-regular. Except for $d = 4$, each regular-faced d-polytope has a nontrivial symmetry group.

There are many further generalizations of the notion of regularity [Mar93]. However, in most cases complete lists of the corresponding polytopes are either not known or available only for $d = 3$. The variants that have been considered include: ***isogonal*** polytopes (requiring vertex-transitivity of $S(P)$), or ***isohedral*** polytopes, the reciprocals of the isogonal polytopes, with a facet-transitive group $S(P)$; more generally, ***k-face-transitive*** polytopes (requiring transitivity of $S(P)$ on the k-faces), for a single value or several values of k; ***congruent-faceted***, or ***monohedral***, polytopes (requiring congruence of the facets); and ***equifaceted*** polytopes (requiring combinatorial isomorphism of the facets). Similar problems have also been considered for nonconvex polytopes or polyhedra, and for tilings [GS87].

16.6 REFLECTION GROUPS

Symmetry properties of geometric figures are closely tied to the algebraic structure of their symmetry groups, which are often subgroups of finite or infinite reflection groups. A classical reference for reflection groups is Coxeter [Cox73]. A more recent text is Humphreys [Hum90].

GLOSSARY

Reflection group: A group generated by (hyperplane) reflections in a finite-dimensional space V. The space can be a real or complex vector space (or affine space). A ***reflection*** is a linear (or affine) transformation whose eigenvalues, save one, are all equal to 1, while the remaining eigenvalue is a primitive k-th root of unity for some $k \geq 2$; in the real case, it is -1. If the space is equipped with further structure, the reflections are assumed to preserve it. For example, if V is real Euclidean, the reflections are Euclidean reflections.

Coxeter group: A group W, finite or infinite, that is generated by finitely many generators $\sigma_1, \ldots, \sigma_n$ and has a presentation of the form $(\sigma_i \sigma_j)^{m_{ij}} = 1$ $(i, j = 1, \ldots, n)$, where the m_{ij} are positive integers or ∞ such that $m_{ii} = 1$ and $m_{ij} = m_{ji} \geq 2$ $(i \neq j)$. The matrix $(m_{ij})_{ij}$ is the ***Coxeter matrix*** of W.

Coxeter diagram: A labeled graph $\mathcal{D}$ that represents a Coxeter group W as follows. The nodes of $\mathcal{D}$ represent the generators σ_i of W. The i-th and j-th node are joined by a (single) branch if and only if $m_{ij} > 2$. In this case, the branch is labeled m_{ij} if $m_{ij} \neq 3$ (and remains unlabeled if $m_{ij} = 3$).

Irreducible Coxeter group: A Coxeter group W whose Coxeter diagram is connected. (Each Coxeter group W is the direct product of irreducible Coxeter groups, with each factor corresponding to a connected component of the diagram of W.)

GENERAL PROPERTIES

Every Coxeter group $W = \langle \sigma_1, \ldots, \sigma_n \rangle$ admits a faithful representation as a re-

flection group in the real vector space R^n. This is obtained as follows. If W has Coxeter matrix $M = (m_{ij})_{ij}$ and $e_1, \ldots, e_n$ is the standard basis of $\mathbb{R}^n$, define the symmetric bilinear form $\langle\ ,\ \rangle_M$ by

$$\langle e_i, e_j \rangle_M := -\cos(\pi/m_{ij}) \quad (i, j = 1, \ldots, n),$$

with appropriate interpretation if $m_{ij} = \infty$. For $i = 1, \ldots, n$ the linear transformation $S_i : \mathbb{R}^n \mapsto \mathbb{R}^n$ given by

$$xS_i := x - 2\langle e_i, x \rangle_M e_i \qquad (x \in \mathbb{R}^n)$$

is the orthogonal reflection in the hyperplane orthogonal to e_i. Let $O(M)$ denote the orthogonal group corresponding to $\langle\ ,\ \rangle_M$. Then $\sigma_i \mapsto S_i$ $(i = 1, \ldots, n)$ defines a faithful representation $\rho : W \mapsto GL(\mathbb{R}^n)$, called the ***canonical representation***, such that $W\rho \subseteq O(M)$.

The group W is finite if and only if the associated form $\langle\ ,\ \rangle_M$ is positive definite; in this case, $\langle\ ,\ \rangle_M$ determines a Euclidean geometry on $\mathbb{R}^n$. In other words, each finite Coxeter group is a finite Euclidean reflection group. Conversely, every finite Euclidean reflection group is a Coxeter group. The finite Coxeter groups have been completely classified by Coxeter and are usually listed in terms of their Coxeter diagrams.

The finite irreducible Coxeter groups with string diagrams are precisely the symmetry groups of the convex regular polytopes, with a pair of dual polytopes corresponding to a pair of groups that are related by reversing the order of the generators. See Section 16.1 for an explanation about how the generators act on the polytopes. Table 16.1.1 also lists the names for the corresponding Coxeter diagrams.

For $p_1, \ldots, p_{n-1} \geq 2$ write $[p_1, \ldots, p_{n-1}]$ for the Coxeter group with string diagram $\bullet\overset{p_1}{\text{---}}\bullet\overset{p_2}{\text{---}}\bullet\cdots\cdots\bullet\overset{p_{n-2}}{\text{---}}\bullet\overset{p_{n-1}}{\text{---}}\bullet$. Then $[p_1, \ldots, p_{n-1}]$ is the automorphism group of the universal abstract regular n-polytope $\{p_1, \ldots, p_{n-1}\}$; see Section 16.8. The regular honeycombs $\{p_1, \ldots, p_{n-1}\}$ on the sphere (convex regular polytopes) or in Euclidean or hyperbolic space are examples of such universal polytopes. The spherical honeycombs are exactly the finite universal polytopes (with $p_i > 2$ for all i). The Euclidean honeycombs arise exactly when $p_i > 2$ for all i and the bilinear form $\langle\ ,\ \rangle_M$ for $[p_1, \ldots, p_{n-1}]$ is positive semi-definite (but not positive definite). Similarly, the hyperbolic honeycombs correspond exactly to the groups $[p_1, \ldots, p_{n-1}]$ that are Coxeter groups of "hyperbolic type" [MS].

There are exactly two sources of finite Coxeter groups, to some extent overlapping: the symmetry groups of convex regular polytopes, and the Weyl groups of root systems, which are important in Lie Theory. The latter occur also in the context of sphere packings; see Conway and Sloane [CS88].

16.7 COMPLEX REGULAR POLYTOPES

Complex regular polytopes are subspace configurations in unitary complex space that share many properties with regular polytopes in real spaces. For a detailed account see Coxeter [Cox93].

GLOSSARY

Complex d-polytope: A d-polytope-configuration as defined in Section 16.2, but now the elements, or ***faces***, are subspaces in unitary complex d-space $\mathbb{C}^d$. However, unlike in real space, the subconfigurations G/F with $dim(G) - dim(F) = 2$ can contain more than 2 proper elements. A ***complex polygon*** is a complex 2-polytope.

Regular complex polytope: A complex polytope P whose (unitary) symmetry group $S(P)$ is transitive on the flags (the maximal sets of mutually incident faces).

ENUMERATION AND GROUPS

The regular complex d-polytopes P are completely known for each d. Every d-polytope can be uniquely described by a ***generalized Schläfli symbol***

$$p_0\{q_1\}p_1\{q_2\}p_2 \cdots p_{d-2}\{q_{d-1}\}p_{d-1},$$

which we explain below. For $d = 1$, the regular polytopes are precisely the point sets on the complex line, which in corresponding real 2-space are the vertex sets of regular convex polygons; the Schläfli symbol is simply p if the real polygon is a p-gon. In general, the entry p_i is the Schläfli symbol for the complex 1-polytope that occurs as the 1-dimensional subconfiguration G/F of P, where F is an $(i-1)$-face and G an $(i+1)$-face of P such that $F \subseteq G$. As is further explained below, the p_i i-faces in this subconfiguration are cyclicly permuted by a hyperplane reflection that leaves the whole polytope invariant. Note that, unlike in real Euclidean space, a hyperplane reflection in unitary complex space need not have period 2 but can have any finite period greater than 1. The meaning of the entries q_i is also given below.

The regular complex polytopes P with $d \geq 2$ are summarized in Table 16.7.1, which includes the numbers f_0 and f_{d-1} of vertices and facets ($(d-1)$-faces) and the group order. Listed are only the nonreal polytopes as well as only one polytope from each pair of duals. A complex polytope is ***real*** if, up to an affine transformation of $\mathbb{C}^d$, all its faces are subspaces that can be described by linear equations over the reals. In particular, $p_0\{q_1\}p_1 \ldots p_{d-2}\{q_{d-1}\}p_{d-1}$ is real if and only if $p_i = 2$ for each i; in this case, $\{q_1, \ldots, q_{d-1}\}$ is the Schläfli symbol for the related regular polytope in real space. As in real space, each polytope $p_0\{q_1\}p_1 \ldots p_{d-2}\{q_{d-1}\}p_{d-1}$ has a dual (reciprocal) and its Schläfli symbol is $p_{d-1}\{q_{d-1}\}p_{d-2} \ldots p_1\{q_1\}p_0$; the symmetry groups are the same and the numbers of vertices and facets are interchanged. The polytope $p\{4\}2\{3\}2 \ldots 2\{3\}2$ is the ***generalized complex d-cube***, and its dual $2\{3\}2 \ldots 2\{3\}2\{4\}p$ the ***generalized complex d-cross-polytope***; if $p = 2$, these are the real d-cubes and d-cross-polytopes, respectively.

The symmetry group $S(P)$ of a complex regular d-polytope P is a finite unitary reflection group in $\mathbb{C}^d$; if $P = p_0\{q_1\}p_1 \ldots p_{d-2}\{q_{d-1}\}p_{d-1}$, then the notation for the group $S(P)$ is $p_0[q_1]p_1 \ldots p_{d-2}[q_{d-1}]p_{d-1}$. If $\Phi = \{\emptyset = F_{-1}, F_0, \ldots, F_{d-1}, F_d = \mathbb{C}^d\}$ is a flag of P, then for each $i = 0, 1, \ldots, d-1$ there is a unitary reflection R_i that fixes F_j for $j \neq i$ and cyclicly permutes the p_i i-faces in the subconfiguration

TABLE 16.7.1 The nonreal complex regular polytopes (up to duality).

DIMENSION	POLYTOPE	f_0	f_{d-1}	$\lvert S(P)\rvert$
$d \geq 1$	$p\{4\}2\{3\}2\ldots2\{3\}2$	p^d	pd	$p^d d!$
$d = 2$	$3\{3\}3$	8	8	24
	$3\{6\}2$	24	16	48
	$3\{4\}3$	24	24	72
	$4\{3\}4$	24	24	96
	$3\{8\}2$	72	48	144
	$4\{6\}2$	96	48	192
	$4\{4\}3$	96	72	288
	$3\{5\}3$	120	120	360
	$5\{3\}5$	120	120	600
	$3\{10\}2$	360	240	720
	$5\{6\}2$	600	240	1200
	$5\{4\}3$	600	360	1800
$d = 3$	$3\{3\}3\{3\}3$	27	27	648
	$3\{3\}3\{4\}2$	72	54	1296
$d = 4$	$3\{3\}3\{3\}3\{3\}3$	240	240	155 520

F_{i+1}/F_{i-1} of P. These generators R_i can be chosen in such a way that in terms of $R_0, \ldots, R_{d-1}$, $S(P)$ has a presentation of the form

$$\begin{cases} R_i^{p_i} = 1 & (0 \leq i \leq d-1), \\ R_i R_j = R_j R_i & (0 \leq i < j-1 \leq d-2), \\ R_i R_{i+1} R_i R_{i+1} R_i \ldots = R_{i+1} R_i R_{i+1} R_i R_{i+1} \ldots \\ \text{with } q_{i+1} \text{ generators on each side } (0 \leq i \leq d-2). \end{cases}$$

This explains the entries q_i in the Schläfli symbol. Conversely, any d unitary reflections that satisfy the first two sets of relations, and generate a finite group, can be used to determine a regular complex polytope by a complex analogue of Wythoff's construction (see Section 16.5). If P is real, then $S(P)$ is conjugate, in the general linear group of $\mathbb{C}^d$, to a finite (real) Coxeter group (see Section 16.6). Complex regular polytopes are only one source for finite unitary reflection groups; there are also others [Cox93].

Recently, Cuypers [Cuy95] has announced the classification of quaternionic regular polytopes (polytope-configurations in quaternionic space).

16.8 ABSTRACT REGULAR POLYTOPES

Abstract regular polytopes are combinatorial structures that generalize the familiar regular polytopes. The terminology adopted is patterned after the classical theory. Many symmetric figures discussed in earlier sections could be treated (and their structure clarified) in this more general framework. Much of the research in this area is quite recent. For a comprehensive account see McMullen and Schulte [MS].

GLOSSARY

Abstract d-polytope: A partially ordered set P, with elements called ***faces***, that satisfies the following conditions. P is equipped with a ***rank function*** with range $\{-1, 0, \ldots, d\}$, which associates with a face F its ***rank*** rank F; if rank $F = j$, F is a ***j-face***, or a ***vertex***, an ***edge***, or a ***facet*** if $j = 0, 1$, or $d-1$, respectively. P has a unique minimal element F_{-1} of rank -1 and a unique maximal element F_d of rank d. These two elements are the ***improper*** faces; the others are ***proper***. The ***flags*** (maximal totally ordered subsets) of P all contain exactly $d+2$ faces (including F_{-1} and F_d). If $F < G$ in P, then $G/F := \{H \in P | F \leq H \leq G\}$ is said to be a ***section*** of P. All sections of P (including P itself) are ***connected***, meaning that, given two proper faces H, H' of a section G/F, there is a sequence $H = H_0, H_1, \ldots, H_k = H'$ of proper faces of G/F (for some k) such that H_{i-1} and H_i are incident for each $i = 1, \ldots, k$. (That is, P is ***strongly connected***.) Finally, if $F < G$ with $0 \leq \text{rank } F + 1 = j = \text{rank } G - 1 \leq d-1$, there are exactly two j-faces H such that $F < H < G$. (Note that this last condition is violated for nonreal complex polytopes.)

Faces and co-faces: We can safely identify a face F of P with the section $F/F_{-1} = \{H \in P | H \leq F\}$. The section $F_d/F = \{H \in P | F \leq H\}$ is the ***co-face*** of P, or the ***vertex figure*** if F is a vertex.

Regular polytope: An abstract polytope P whose ***automorphism group*** $A(P)$ (the group of order-preserving permutations of $\mathcal{P}$) is transitive on the flags. (Then $A(P)$ must be simply flag-transitive.)

C-group: A group A generated by involutions $\sigma_1, \ldots, \sigma_m$ (that is, a quotient of a Coxeter group) such that the ***intersection property*** holds:

$$\langle \sigma_i | i \in I \rangle \cap \langle \sigma_i | i \in J \rangle = \langle \sigma_i | i \in I \cap J \rangle \text{ for all } I, J \subset \{1, \ldots, m\}.$$

The letter "C" stands for "Coxeter". (Coxeter groups are C-groups, but not vice versa.)

String C-group: A C-group $A = \langle \sigma_1, \ldots, \sigma_m \rangle$ such that $(\sigma_i \sigma_j)^2 = 1$ if $1 \leq i < j - 1 \leq m - 1$. (That is, A is a quotient of a Coxeter group with a string Coxeter diagram.)

Realization: For a regular d-polytope P with vertex-set $\mathcal{F}_0$, a surjection $\beta : \mathcal{F}_0 \mapsto V$ onto a set V of points in a Euclidean space, such that each automorphism of P induces an isometric permutation of V. Then V is the ***vertex set*** of the realization β.

GENERAL PROPERTIES

Abstract 2-polytopes are isomorphic to ordinary n-gons or apeirogons (Section 16.2). Except for some degenerate cases, the abstract 3-polytopes with finite faces and vertex figures are in one-to-one correspondence with the maps on surfaces (Section 16.3). Accordingly, a finite 4-polytope P has facets and vertex figures that are isomorphic to maps on surfaces.

The group $A(P)$ of every regular d-polytope P is a string C-group. Fix a flag $\Phi := \{F_{-1}, F_0, \ldots, F_d\}$, the ***base flag*** of P. Then $A(P)$ is generated by ***distinguished generators*** $\rho_0, \ldots, \rho_{d-1}$ (*with respect to* Φ), where ρ_i is the unique

automorphism that keeps all but the i-face of Φ fixed. These generators satisfy relations

$$(\rho_i\rho_j)^{p_{ij}} = 1 \quad (i,j = 0,\ldots,d-1),$$

with $p_{ii} = 1$, $p_{ij} = p_{ji} \geq 2$ $(i \neq j)$, and $p_{ij} = 2$ if $|i-j| \geq 2$; in particular, $A(P)$ is a string C-group with generators $\rho_0,\ldots,\rho_{d-1}$. The numbers $p_i := p_{i-1,i}$ determine the (*Schläfli*) ***type*** $\{p_1,\ldots,p_{d-1}\}$ of P. The group $A(P)$ is a quotient of the Coxeter group $[p_1,\ldots,p_{d-1}]$ (Section 16.6), but in general the quotient is proper.

Conversely, if A is a string C-group with generators $\rho_0,\ldots,\rho_{d-1}$, then it is the group of a regular d-polytope P, and $\rho_0,\ldots,\rho_{d-1}$ are the distinguished generators with respect to some base flag of P. The i-faces of P are the right cosets of the subgroup $A_i := \langle \rho_k | k \neq i \rangle$ of A, and in P, $A_i\varphi \leq A_j\psi$ if and only if $i \leq j$ and $A_i\varphi \cap A_j\psi \neq \emptyset$. For any $p_1,\ldots,p_{d-1} \geq 2$, $[p_1,\ldots,p_{d-1}]$ is a string C-group and the corresponding d-polytope is the ***universal*** regular d-polytope $\{p_1,\ldots,p_{d-1}\}$; every other regular d-polytope of the same type $\{p_1,\ldots,p_{d-1}\}$ is derived from it by making identifications. Examples are the regular spherical, Euclidean, and hyperbolic honeycombs. The one-to-one correspondence between string C-groups and the groups of regular polytopes sets up a powerful dialogue between groups on one hand and polytopes on the other. There is also a similar such dialogue for an important class of nearly regular polytopes, called ***chiral*** polytopes; see Schulte and Weiss [SW94].

CLASSIFICATION BY TOPOLOGICAL TYPE

Abstract polytopes are not a priori embedded into an ambient space. Therefore for abstract polytopes, the traditional enumeration of regular polytopes is replaced by the classification by global or local topological type. On the group level, this translates into the enumeration of finite string C-groups with certain kinds of presentations. In this context, the classical regular convex polytopes are precisely the abstract regular polytopes that are locally and globally spherical.

Much work has been done in the toroidal and locally toroidal case [MS, Sch94]. A ***regular toroid*** of rank $d+1$ is the quotient of a regular tessellation $\{p_1,\ldots,p_d\}$ in Euclidean d-space by a lattice that is invariant under all symmetries of the vertex figure of $\{p_1,\ldots,p_d\}$. If $d = 2$, these are the reflexible regular torus maps of [CM80]. For $d \geq 3$ there are three infinite sequences of ***cubical toroids*** of type $\{4,3^{d-2},4\}$, and for $d = 4$ there are two infinite sequences of ***exceptional toroids*** for each of the types $\{3,3,4,3\}$ and $\{3,4,3,3\}$. Their groups are known in terms of generators and relations.

For regular d-polytopes P_1 and P_2, let $\langle P_1,P_2\rangle$ denote the class of all regular $(d+1)$-polytopes with facets isomorphic to P_1 and vertex figures isomorphic to P_2. Each nonempty class $\langle P_1,P_2\rangle$ contains a ***universal polytope*** denoted by $\{P_1,P_2\}$, which "covers" all other polytopes in its class. Classification by local topological type means enumeration of all finite universal polytopes $\{P_1,P_2\}$ where P_1 and P_2 are of the prescribed (global) topological type. There are variants of this definition. A polytope Q in $\langle P_1,P_2\rangle$ is ***locally toroidal*** if P_1 and P_2 are regular convex polytopes (spheres) or regular toroids, with at least one of the latter kind.

Locally toroidal regular polytopes can only exist in ranks 4, 5, and 6. The enumeration is complete for rank 5, and nearly complete for rank 4. In rank 6, a list of finite polytopes is known that is conjectured to be complete.

REALIZATIONS

A good number of the geometric figures discussed in the earlier sections could be described in the general context of realizations of abstract regular polytopes. For an account of realizations see McMullen [McM94] (or [MS]).

Let $\beta : \mathcal{F}_0 \mapsto V$ be a realization of a regular d-polytope P, and let $\mathcal{F}_j$ denote the set of j-faces of P ($j = -1, 0, \ldots, d$). With $\beta_0 := \beta$, $V_0 := V$, then for $j = 1, \ldots, d$, β recursively induces a surjection $\beta_j : \mathcal{F}_j \mapsto V_j$, with $V_j \subset 2^{V_{j-1}}$, given by

$$F\beta_j := \{G\beta_{j-1} | G \in \mathcal{F}_{j-1}, G \leq F\}$$

for each $F \in \mathcal{F}_j$. It is convenient to identify β and $\{\beta_j\}_{j=0}^{d}$ and also call the latter a realization of $\mathcal{P}$. The realization is ***faithful*** if each β_j is a bijection; otherwise, it is ***degenerate***. Its ***dimension*** is the dimension of the affine hull of V.

The traditional approach in the study of regular figures starts from a Euclidean (or other) space and describes all figures of a specified kind that are regular according to some geometric definition of regularity.

Instead, a rather new approach proceeds from a given abstract regular polytope P and describes all the realizations of P. For a finite P, each realization β is uniquely determined by its ***diagonal vector*** Δ, whose components are the squared lengths of the diagonals (pairs of vertices) in the diagonal classes of P modulo $A(P)$. Each orthogonal representation of $A(P)$ yields one or more (possibly degenerate) realizations of P. Then taking the sum of two representations of $A(P)$ is equivalent to an operation for the related realizations called a ***blend***, which in turn amounts to adding the corresponding diagonal vectors. If we identify the realizations with their diagonal vectors, then the space of all realizations of P becomes a closed convex cone $C(P)$, the ***realization cone of*** P, whose finer structure is given by the irreducible representations of $A(P)$. The extreme rays of $C(P)$ correspond to the ***pure*** (unblended) realizations, which are given by the irreducible representations of $A(P)$. Each realization of P is a blend of pure realizations.

For instance, a regular n-gon P has $\lfloor \frac{1}{2}n \rfloor$ diagonal classes, and for each $k = 1, \ldots, \lfloor \frac{1}{2}n \rfloor$, there is a planar regular star-polygon $\{\frac{n}{k}\}$ if $(n, k) = 1$ (Section 16.2), or a "degenerate star-polygon $\{\frac{n}{k}\}$" if $(n, k) > 1$; the latter is a degenerate realization of P, which reduces to a line segment if $n = 2k$. For the regular icosahedron P there are 3 pure realizations. Apart from the usual icosahedron $\{3, 5\}$ itself, there is another 3-dimensional pure realization, namely the great icosahedron $\{3, \frac{5}{2}\}$ (Section 16.2). The final pure realization is induced by its covering of $\{3, 5\}/2$, the ***hemi-icosahedron*** (obtained from P by identifying antipodal vertices), all of whose diagonals are edges; thus its vertices must be those of a 5-simplex. For other polytopes see [MS].

16.9 SOURCES AND RELATED MATERIAL

SURVEYS

[CS88]: A monograph on sphere packings and related topics.

[Cox73]: A monograph on the traditional regular polytopes, regular tessellations, and reflection groups.

[Cox93]: A monograph on complex regular polytopes and complex reflection groups.

[CM80]: A monograph on discrete groups and their presentations.

[Fej64]: A monograph on regular figures, mainly in 3 dimensions.

[Grü67]: A monograph on convex polytopes.

[GS87]: A monograph on plane tilings and patterns.

[Hum90]: A monograph on Coxeter groups and reflection groups.

[Joh]: A monograph on uniform polytopes and semi-regular figures.

[Mar93]: A survey on symmetric convex polytopes and a hierarchical classification by symmetry.

[McM94]: A survey on abstract regular polytopes with emphasis on geometric realizations.

[MS]: A monograph on abstract regular polytopes and their groups.

[Sch94]: A survey on toroidal regular polytopes and their classification.

[SF88]: A text on interdisciplinary aspects of polyhedra and their symmetries.

RELATED CHAPTERS

Chapter 3: Tilings
Chapter 13: Basic properties of convex polytopes
Chapter 18: Polyhedral maps
Chapter 50: Sphere packing and coding theory
Chapter 51: Crystals and quasicrystals

REFERENCES

[CS88] J.H. Conway and N.J.A. Sloane. *Sphere Packings, Lattices and Groups*. Springer-Verlag, New York, 1988.

[Cox68] H.S.M. Coxeter. Regular skew polyhedra in 3 and 4 dimensions and their topological analogues. In *Twelve Geometric Essays*, pages 75–105. Southern Illinois University Press, Carbondale, 1968.

[Cox73] H.S.M. Coxeter. *Regular Polytopes* (3rd edition). Dover, New York, 1973.

[Cox93] H.S.M. Coxeter. *Regular Complex Polytopes* (2nd edition). Cambridge University Press, 1993.

[CM80] H.S.M. Coxeter and W.O.J. Moser. *Generators and Relations for Discrete Groups* (4th edition). Springer-Verlag, Berlin, 1980.

[Cuy95] H. Cuypers. Regular quaternionic polytopes. *Linear Algebra Appl.*, 226–228:311–329, 1995.

[Dre85] A.W.M. Dress. A combinatorial theory of Grünbaum's new regular polyhedra. Part II: Complete enumeration. *Aequationes Math.*, 29:222–243, 1985.

[Fej64] L. Fejes Tóth. *Regular Figures*. Macmillan, New York, 1964.

[Gar67] C.W.L. Garner. Regular skew polyhedra in hyperbolic three-space. *J. Canad. Math. Soc.*, 19:1179–1186, 1967.

[Grü67] B. Grünbaum. *Convex Polytopes*. Interscience, London, 1967; revised edition (V. Klee and P. Kleinschmidt, editors), *Graduate Texts in Math.*, Springer-Verlag, in preparation.

[Grü77a] B. Grünbaum. Regularity of graphs, complexes and designs. In *Problèmes combinatoires et théorie des graphes*, pages 191–197. Number 260 of *Colloq. Int. CNRS*, Orsay, 1977.

[Grü77b] B. Grünbaum. Regular polyhedra – old and new. *Aequationes Math.*, 16:1–20, 1977.

[GS87] B. Grünbaum and G.C. Shephard. *Tilings and Patterns*. Freeman, New York, 1987.

[Hum90] J.E. Humphreys. *Reflection Groups and Coxeter Groups*. Cambridge University Press, 1990.

[Joh] N.W. Johnson. *Uniform Polytopes*. To appear.

[Mar93] H. Martini. A hierarchical classification of Euclidean polytopes with regularity properties. In T. Bisztriczky, P. McMullen, R. Schneider, and A. Ivić Weiss, editors, *Polytopes: Abstract, Convex and Computational*, volume 440 of NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci., pages 71–96. Kluwer, Dordrecht, 1994.

[McM68] P. McMullen. Regular star-polytopes, and a theorem of Hess. *Proc. London Math. Soc.* (3), 18:577–596, 1968.

[McM94] P. McMullen. Modern developments in regular polytopes. In T. Bisztriczky, P. McMullen, R. Schneider, and A. Ivić Weiss, editors, *Polytopes: Abstract, Convex and Computational*, volume 440 of NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci., pages 97–124. Kluwer, Dordrecht, 1994.

[MS] P. McMullen and E. Schulte. *Abstract Regular Polytopes*. Monograph in preparation.

[Sch94] E. Schulte. Classification of locally toroidal regular polytopes. In T. Bisztriczky, P. McMullen, R. Schneider, and A. Ivić Weiss, editors, *Polytopes: Abstract, Convex and Computational*, volume 440 of NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci., pages 125–154. Kluwer, Dordrecht, 1994.

[SF88] M. Senechal and G. Fleck. *Shaping Space*. Birkhäuser, Boston, 1988.

[SW91] E. Schulte and J.M. Wills. Combinatorially regular polyhedra in three-space. In K.H. Hofmann and R. Wille, editors, *Symmetry of Discrete Mathematical Structures and Their Symmetry Groups*, pages 49–88. Heldermann Verlag, Berlin, 1991.

[SW94] E. Schulte and A.I. Weiss. Chirality and projective linear groups. *Discrete Math.*, 131:221–261, 1994.

17 POLYTOPE SKELETONS AND PATHS

Gil Kalai

INTRODUCTION

The k-dimensional ***skeleton*** of a d-polytope P is the set of all faces of the polytope of dimension at most k. The 1-skeleton of P is called the ***graph*** of P and denoted by $G(P)$. $G(P)$ can be regarded as an abstract graph whose vertices are the vertices of P, with two vertices adjacent if they form the endpoints of an edge of P.

In this chapter, we will describe results and problems concerning graphs and skeletons of polytopes. In Section 17.1 we briefly describe the situation for 3-polytopes. In Section 17.2 we consider general properties of polytopal graphs—subgraphs and induced subgraphs, connectivity and separation, expansion, and other properties. In Section 17.3 we discuss problems related to diameters of polytopal graphs in connection with the simplex algorithm and the Hirsch conjecture, and to directed graphs obtained by directing graphs of polytopes via a linear functional. Section 17.4 is devoted to skeletons of polytopes, connectivity, collapsibility and shellability, empty faces and polytopes with "few vertices," and the reconstruction of polytopes from their low-dimensional skeletons; finally we consider what can be said about the collections of all k-faces of a d-polytope, first for $k = d - 1$ and then when k is fixed and d is large compared to k.

17.1 THREE-DIMENSIONAL POLYTOPES

GLOSSARY

Convex polytopes and their *faces* (and, in particular their *vertices*, *edges*, and *facets*) are defined in Chapter 13 of this Handbook.

A graph is ***d-polytopal*** if it is the graph of some d-polytope.

The following standard graph-theoretic concepts are used: *subgraphs*, *induced subgraphs*, the *complete graph* K_n on n vertices, *cycles*, *trees*, a *spanning tree* of a graph, *valence* (or *degree*) of a vertex in a graph, *planar* graphs, *d-connected* graphs, *coloring* of a graph, *subdivision* of a graph, and *Hamiltonian* graphs.

We briefly discuss results on 3-polytopes. Some of the following theorems are the starting points of much research, sometimes of an entire theory. Only in a few cases are there high-dimensional analogues, and this remains an interesting goal for further research.

THEOREM 17.1.1 *Whitney* (1931)

Let G be the graph of a 3-polytope P. Then the graphs of faces of P are precisely the induced cycles in G that do not separate G.

THEOREM 17.1.2 *Steinitz* (1916)

A graph G is a graph of a 3-polytope if and only if G is planar and 3-connected.

Steinitz's theorem is the first of several theorems that describe the tame behavior of 3-polytopes. These theorems fail already in dimension four; see Chapter 13.

The theory of planar graphs is a wide and rich theory. Let us quote here the fundamental theorem of Kuratowski.

THEOREM 17.1.3 *Kuratowski* (1930)

A graph G is planar if and only if G does not contain a subdivision of K_5 or $K_{3,3}$.

THEOREM 17.1.4 *Lipton and Tarjan* (1979), *in a stronger form given by Miller* (1986)

The graph of every 3-polytope with n vertices can be separated, by $2\sqrt{2n}$ vertices forming a circuit in the graph, into connected components of size at most $2n/3$.

It is worth mentioning that the Koebe circle packing theorem gives a new approach to both the Steinitz and Lipton-Tarjan theorems. (See [Zie95, PA95]).

Euler's formula $V - E + F = 2$ has many applications concerning graphs of 3-polytopes; in higher dimensions, our knowledge of face numbers of polytopes (see Chapter 15) applies to the study of their graphs and skeletons. Simple applications of Euler's theorem are:

THEOREM 17.1.5

Every 3-polytopal graph has a vertex of valence at most 5. (Equivalently, every 3-polytope has a face with at most five sides.)

THEOREM 17.1.6

Every 3-polytope has either a trivalent vertex or a triangular face.

A deeper application of Euler's theorem is:

THEOREM 17.1.7 *Kotzig* (1955)

Every 3-polytope has two adjacent vertices the sum of whose valences is at most 13.

For a simple 3-polytope P, let $p_k = p_k(P)$ be the number of k-sized faces of P.

THEOREM 17.1.8 *Eberhard* (1891)

For every finite sequence (p_k) of nonnegative integers with $\sum_{k\geq 3}(6-k)p_k = 12$, there exists a simple 3-polytope P with $p_k(P) = p_k$ for every $k \neq 6$.

Eberhard's theorem is the starting point of a large number of results and problems. While no high-dimensional direct analogues are known or even conjectured, the results and problems on facet-forming polytopes and nonfacets mentioned below seem related.

THEOREM 17.1.9 *Motzkin* (1964)

The graph of a simple 3-polytope whose facets have 0 (mod 3) vertices has, all together, an even number of vertices.

THEOREM 17.1.10 *Barnette* (1966)

Every 3-polytopal graph contains a spanning tree of maximal valence 3.

We will now describe some results and a conjecture on colorability and Hamiltonian circuits.

THEOREM 17.1.11 *Four Color Theorem: Appel and Haken* (1977)
The graph of every 3*-polytope is* 4*-colorable.*

THEOREM 17.1.12 *Tutte* (1956)
4*-connected planar graphs are Hamiltonian.*

Tait conjectured in 1880, and Tutte disproved in 1946, that the graph of every simple 3-polytope is Hamiltonian. This started a rich theory of trivalent planar graphs without large paths.

CONJECTURE 17.1.13 *Barnette*
Every graph of a simple 3*-polytope whose facets have an even number of vertices is Hamiltonian.*

Finally, there are several exact and asymptotic formulas for the numbers of distinct graphs of 3-polytopes. A remarkable enumeration theory was developed by Tutte and was further developed by several authors. We will quote one result.

THEOREM 17.1.14 *Tutte* (1962)
The number of rooted simplicial 3*-polytopes with* v *vertices is*

$$\frac{2(4v-11)!}{(3v-7)!(v-2)!}.$$

17.2 GRAPHS OF d-POLYTOPES—GENERALITIES

GLOSSARY

For a graph G, TG denotes any ***subdivision*** of G, i.e., any graph obtained from G by replacing the edges of G by paths with disjoint interiors.

A d-polytope P is ***simplicial*** if all its proper faces are simplices. P is ***simple*** if every vertex belongs to d edges or, equivalently, if the polar of P is simplicial. P is ***cubical*** if all its proper faces are cubes.

A simplicial polytope P is ***stacked*** if it is obtained by the repeated operation of gluing a simplex along a facet.

For the definition of the *cyclic polytope* $C(d,n)$, see Chapter 13.

For two graphs G and H (considered as having disjoint sets V and V' of vertices), $G+H$ denotes the graph on $V \cup V'$ that contains all edges of G and H together with all edges of the form $\{v,v'\}$ for $v \in V$ and $v' \in V'$.

A graph G is ***d-connected*** if G remains connected after the deletion of any set of at most $d-1$ vertices.

An ***empty simplex*** of a polytope P is a set S of vertices such that S does not form a face but every proper subset of S forms a face.

A graph G whose vertices are embedded in $\mathbb{R}^d$ is ***rigid*** if every small perturbation of the vertices of G that does not change the distance of adjacent vertices in G is induced by an affine rigid motion of $\mathbb{R}^d$. G is ***generically d-rigid*** if it is rigid with respect to "almost all" embeddings of its vertices into $\mathbb{R}^d$. (Generic rigidity is thus a graph theoretic property, but no description of it in pure combinatorial terms is known for $d > 2$; cf. Chapter 49.)

A set A of vertices of a graph G is ***totally separated*** by a set B of vertices, if A and B are disjoint and every path between two distinct vertices in A meets B.

A graph G is an ***ϵ-expander*** if, for every set A of at most half the vertices of G, there are at least $\epsilon \cdot |A|$ vertices not in A that are adjacent to vertices in A.

Neighborly polytopes and $(0,1)$*-polytopes* are defined in Chapter 13.

The *polar dual* P^Δ of a polytope P is defined in Chapter 13.

SUBGRAPHS AND INDUCED SUBGRAPHS

THEOREM 17.2.1 *Grünbaum* (1965)

Every d-polytopal graph contains a TK_{d+1}.

THEOREM 17.2.2 *Kalai* (1987)

The graph of a simplicial d-polytope P *contains a* TK_{d+2} *if and only if* P *is not stacked.*

One important difference between the situation for $d = 3$ and for $d > 3$ is that K_n, for every $n > 4$, is the graph of a 4-dimensional polytope (e.g., a cyclic polytope). Simple manipulations on the cyclic 4-polytope with n vertices show:

PROPOSITION 17.2.3 *Perles*

(i) *Every graph* G *is a spanning subgraph of the graph of a 4-polytope.*

(ii) *For every graph* G, $G + K_n$ *is a d-polytopal graph for some* n *and some* d.

This proposition extends easily to higher-dimensional skeletons in place of graphs. It is not known what the minimal dimension is for which $G + K_n$ is d-polytopal, nor even whether $G + K_n$ (for some $n = n(G)$) can be realized in some bounded dimension uniformly for all graphs G.

CONNECTIVITY AND SEPARATION

THEOREM 17.2.4 *Balinski* (1961)

The graph of a d-polytope is d-connected.

A set S of d vertices that separates P must form an empty simplex; in this case, P can be obtained by gluing two polytopes along a simplex facet of each.

THEOREM 17.2.5 *Larman* (1970)

Let G *be the graph of a d-polytope. Let* $e = \lfloor (d+1)/3 \rfloor$. *Then for every two disjoint sequences* $(v_1, v_2, \ldots, v_e)$ *and* $(w_1, w_2, \ldots w_e)$ *of vertices of* G, *there are* e *vertex-disjoint paths connecting* v_i *to* w_i, $i = 1, 2, \ldots, e$.

PROBLEM 17.2.6 *Larman*

Is the last theorem true for $e = \lfloor d/2 \rfloor$*?*

THEOREM 17.2.7 *Cauchy, Dehn, Alexandrov, Whiteley, ...*

(i) *If* P *is a simplicial* d*-polytope,* $d \geq 3$*, then* $G(P)$ *(with its embedding in* $\mathbb{R}^d$*) is rigid.*

(ii) *For a general* d*-polytope* P*, let* G' *be a graph (embedded in* $\mathbb{R}^d$*) obtained from* $G(P)$ *by triangulating the* 2*-faces of* P *without introducing new vertices. Then* G' *is rigid.*

COROLLARY 17.2.8

For a simplicial d*-polytope* P*,* $G(P)$ *is generically* d*-rigid. For a general* d*-polytope* P *and a graph* G' *(considered as an abstract graph) as in the previous theorem,* G' *is generically* d*-rigid.*

The main combinatorial application of the above theorem is the Lower Bound Theorem (see Chapter 15) and its extension to general polytopes. Note that Corollary 17.2.8 can be regarded also as a strong form of Balinski's theorem. It is easy to see that generic d-rigidity implies d-connectivity. If the graph G of a general d-polytope P can be separated into two parts (say a red part and a blue part) by deleting $d-1$ vertices, then it is possible to triangulate the 2-faces of P without introducing a blue-red edge, and hence the resulting triangulation is not $(d-1)$-connected and therefore not generically d-rigid.

Let $\mu(n,d) = f_{d-1}(C(d,n))$ be the number of facets of a cyclic d-polytope with n vertices, which, by the Upper Bound Theorem, is the maximal number of facets possible for a d-polytope with n vertices.

THEOREM 17.2.9 *Klee* (1964)

The number of vertices of a d*-polytope that can be totally separated by* n *vertices is at most* $\mu(n,d)$*.*

EXPANSION

Expansion properties for the graph of the d-dimensional cube are known and important in various areas of combinatorics. By direct combinatorial methods, one can obtain expansion properties of duals to cyclic polytopes. There are a few positive results and several interesting conjectures on expansion properties of graphs of large families of polytopes.

THEOREM 17.2.10 *Kalai* (1992)

Graphs of duals to neighborly d*-polytopes with* n *facets are* ϵ*-expanders for* $\epsilon = O(n^{-4})$*.*

CONJECTURE 17.2.11 *Mihail and Vazirani*

Graphs of $(0,1)$*-polytopes* P *have the following expansion property: For every set* A *of at most half the vertices of* P*, the number of edges joining vertices in* A *to vertices not in* A *is at least* $|A|$*.*

It is also conjectured that graphs of polytopes cannot have very good expansion properties:

CONJECTURE 17.2.12 *Graphs of polytopes are not very good expanders*
Let d be fixed and set $r = \lfloor d/2 \rfloor$. The graph of every simple d-polytope with n vertices can be separated into two parts, each having at least $n/3$ vertices, by removing $O(n^{(r-1)/r})$ vertices.

CONJECTURE 17.2.13 *Expansion properties of random polytopes*
A random simple d-polytope with n facets is an $O(1/(n-d))$-expander.

CONJECTURE 17.2.14 *There are only a "few" graphs of polytopes*
The number of distinct (isomorphism types) of graphs of simple d-polytopes with n vertices is at most C_d^n, where C_d is a constant depending on d.

It is even possible that the same constant applies for all dimensions and that the conjecture holds even for graphs of general polytopes.

OTHER PROPERTIES

CONJECTURE 17.2.15 *Barnette*
Every graph of a simple d-polytope, $d \geq 4$, is Hamiltonian.

THEOREM 17.2.16 *Goodman and Onishi* (1984)
For a simple d-polytope P, $G(P)$ is 2-colorable if and only if $G(P^\Delta)$ is d-colorable.

17.3 DIAMETERS OF POLYTOPAL GRAPHS

GLOSSARY

A ***d-polyhedron*** is the intersection of a finite number of halfspaces in $\mathbb{R}^d$.

$\Delta(d,n)$ denotes the maximal diameter of the graphs of d-dimensional polyhedra P with n facets.

$\Delta_{\mathrm{b}}(d,n)$ denotes the maximal diameter of the graphs of d-polytopes with n vertices.

Given a d-polyhedron P and a linear functional ϕ on $\mathbb{R}^d$, we denote by $G^{\rightarrow}(P)$ the directed graph obtained from $G(P)$ by directing an edge $\{v,u\}$ from v to u if $\phi(v) \leq \phi(u)$. $v \in P$ is a ***top vertex*** if ϕ attains its maximum value in P on v.

Let $H(d,n)$ be the maximum over all d-polyhedra with n facets and all linear functionals on $\mathbb{R}^d$ of the maximum length of a minimal monotone path from any vertex to a top vertex.

Let $M(d,n)$ be the maximal number of vertices in a monotone path over all d-polyhedra with n facets and all linear functionals on $\mathbb{R}^d$.

For the notions of *simplicial complex*, *polyhedral complex*, *pure simplicial complex*, and the *boundary complex* of a polytope, see Chapter 15.

Given a pure $(d-1)$-dimensional simplicial (or polyhedral) complex K, the ***dual graph*** $G^{\Delta}(K)$ of K is the graph whose vertices are the facets ($(d-1)$-faces) of K, with two facets F, F' adjacent if $\dim(F \cap F') = d-2$.

A pure simplicial complex K is ***vertex-decomposable*** if there is a vertex v of K such that $\mathrm{lk}(v) = \{S \backslash \{v\} \mid S \in K, v \in S\}$ and $\mathrm{ast}(v) = \{S \mid S \in K, v \notin S\}$ are both vertex-decomposable. (The complex $K = \{\emptyset\}$ consisting of the empty face alone is vertex-decomposable.)

It is a long-outstanding open problem to determine the behavior of the function $\Delta(d,n)$. In 1957, Hirsch conjectured that $\Delta(d,n) \leq n-d$. Klee and Walkup showed that the Hirsch conjecture is false for unbounded polyhedra. The Hirsch conjecture for bounded polyhedra is still open. The special case asserting that $\Delta_{\mathrm{b}}(d,2d) = d$ is called the ***d-step conjecture***, and it was shown by Klee and Walkup to imply that $\Delta_{\mathrm{b}}(d,n) \leq n-d$. Another equivalent formulation is that between any pair of vertices v and w of a polytope P there is a ***nonrevisiting path***, i.e., a path $v = v_1, v_2, \ldots, v_m = w$ such that for every facet F of P, if $v_i, v_j \in F$ for $i < j$ then $v_k \in F$ for every k, $i \leq k \leq j$.

THEOREM 17.3.1 *Klee and Walkup* (1967)

$$\Delta(d,n) \geq n - d + \min\{\lfloor d/4 \rfloor, \lfloor (n-d)/4 \rfloor\}.$$

THEOREM 17.3.2 *Adler* (1974)

$$\Delta_{\mathrm{b}}(d,n) > \lfloor n - d - (n-d)/\lfloor 5d/4 \rfloor \rfloor.$$

THEOREM 17.3.3 *Larman* (1970)

$$\Delta(d,n) \leq n 2^{d-3}.$$

THEOREM 17.3.4 *Kalai and Kleitman* (1992)

$$\Delta(d,n) \leq n \cdot \binom{\log n + d}{d} \leq n^{\log d + 1}.$$

THEOREM 17.3.5 *Provan and Billera* (1980)

Let G be the dual graph that corresponds to a vertex-decomposable $(d-1)$-dimensional simplicial complex with n vertices. Then the diameter of G is at most $n-d$.

It is known that this theorem does not imply the Hirsch conjecture (for polytopes) since there are simplicial polytopes whose boundary complexes are not vertex-decomposable. Yet, such examples are not so easy to come by.

Some special classes of polytopes are known to satisfy the Hirsch bound or to have upper bounds for their diameters that are polynomial in d and n.

THEOREM 17.3.6 *Naddef* (1989)

The graph of every $(0,1)$ d-polytope has diameter at most d.

Balinski proved the Hirsch bound for dual transportation polytopes, Dyer and Frieze showed a polynomial upper bound for unimodular polyhedra, Kalai observed that if the ratio between the number of facets and the dimension is bounded above

for the polytope and all its faces then the diameter is bounded above by a polynomial in the dimension, Kleinschmidt and Onn proved extensions of Naddef's results to integral polytopes, and Deza and Onn found upper bounds for the diameter in terms of lattice points in the polytope.

The value of $\Delta(d,n)$ is a lower bound for the number of iterations needed for Dantzig's simplex algorithm for linear programming with any pivot rule. However, it is still an open problem to find pivot rules where each pivot step can be computed with a polynomial number of arithmetic operations in d and n such that the number of pivot steps needed comes close to the upper bounds for $\Delta(d,n)$ given above. It is worth noting, however, that by using linear programming it is possible to find a path in a polytope P that obeys the upper bounds given above such that the number of arithmetic operations is bounded by the size of the path times a polynomial in the input size.

The upper bounds for $\Delta(d,n)$ mentioned above apply even to $H(d,n)$. Klee and Minty considered a certain geometric realization of the d-cube to show that

THEOREM 17.3.7 *Klee and Minty* (1972)
$M(d,2d) \geq 2^d$.

Recent far-reaching extensions of the Klee-Minty construction were found by Amenta and Ziegler.

Objective functions allow us to direct the graph of the polytope so that every edge is directed toward the vertex with the higher value of the objective function. Let us consider objective functions that give different values to adjacent vertices, and call the digraph resulting from such an objective function a ***polytopal digraph.*** An important property of such directed graphs is that there is always one sink (and one source). This property is inherited for induced subgraphs on vertices of any face of the polytope.

The h-vector of a simplicial polytope P has a simple and important interpretation in terms of the directed graph that corresponds to the polar of P. The number $h_k(P)$ is the number of vertices v of P^Δ of outdegree k. (Recall that every vertex in a simple polytope has exactly d neighboring vertices.) Switching from ϕ to $-\phi$, one gets the Dehn-Sommerville relations $h_k = h_{d-k}$ (including the Euler relation for $k = 0$); see Chapter 15.

17.4 SKELETONS OF POLYTOPES

GLOSSARY

A pure polyhedral complex K is ***strongly connected*** if its dual graph is connected.

A ***shelling order*** of the facets of a polyhedral $(d-1)$-dimensional sphere is an ordering of the set of facets $F_1, F_2, \ldots, F_n$ so that the simplicial complex K_i spanned by $F_1 \cup F_2 \cup \cdots \cup F_i$ is a simplicial ball for every $i < n$. A polyhedral complex is ***shellable*** if there exists a shelling order of its facets.

A simplicial polytope is ***extendably shellable*** if any way to start a shelling can be continued to a shelling.

An ***elementary collapse*** on a simplicial complex is the deletion of two faces F and G so that F is maximal and G is a codimension-1 face of F that is not included in any other maximal face. A polyhedral complex is ***collapsible*** if it can be reduced to the void complex by repeated applications of elementary collapses.

A d-dimensional polytope P is ***facet-forming*** if there is a $(d+1)$-dimensional polytope Q such that all facets of Q are combinatorially isomorphic to P. If no such Q exists, P is called a ***nonfacet***.

A ***rational polytope*** is a polytope whose vertices have rational coordinates. (Not every polytope is combinatorially isomorphic to a rational polytope; see Chapter 13.)

A d-polytope P is ***k-simplicial*** if all its faces of dimension at most k are simplices. P is ***k-simple*** if its polar dual P^{Δ} is k-simplicial.

Zonotopes are defined in Chapters 13 and 15.

Let K be a polyhedral complex. An ***empty simplex*** S of K is a minimal nonface of K, i.e., a subset S of the vertices of K with S itself not in K, but every proper subset of S in K.

Let K be a polyhedral complex and let U be a subset of its vertices. The ***induced subcomplex*** of K on U, denoted by $K[U]$, is the set of all faces in K whose vertices belong to U. An ***empty face*** of K is an induced polyhedral subcomplex of K that is homeomorphic to a polyhedral sphere. An empty 2-dimensional face is called an ***empty polygon***. An ***empty pyramid*** of K is an induced subcomplex of K that consists of all the proper faces of a pyramid over a face of K.

CONNECTIVITY AND SUBCOMPLEXES

THEOREM 17.4.1 *Grünbaum* (1965)

The i-skeleton of every d-polytope contains a subdivision of $\mathrm{skel}_i(\Delta^d)$*, the i-skeleton of a d-simplex.*

THEOREM 17.4.2 *Folklore*

(i) *For $i > 0$,* $\mathrm{skel}_i(P)$ *is strongly connected.*

(ii) *For every face F, let $U_i(F)$ be the set of all i-faces of P containing F. Then if $i > \dim F$, $U_i(F)$ is strongly connected.*

Part (ii) follows at once from the fact that the faces of P containing F correspond to faces of the quotient polytope P/F. However, properties (i) and (ii) together are surprisingly strong, and all the known upper bounds for diameters of graphs of polytopes rely only on properties (i) and (ii) for the dual polytope.

THEOREM 17.4.3 *van Kampen and Flores* (1935)

For $i \geq \lfloor d/2 \rfloor$, $\mathrm{skel}_i(\Delta^{d+1})$ *is not embeddable in S^{d-1} (and hence not in the boundary complex of any d-polytope).*

(This extends the fact that K_5 is not planar.)

CONJECTURE 17.4.4 *Lockeberg*

For every partition of $d = d_1 + d_2 + \cdots d_k$ and two vertices v and w of P, there are k disjoint paths between v and w such that the i th path is a path of d_i-faces in which any two consecutive faces have (d_i-1)-dimensional intersection.

SHELLABILITY AND COLLAPSIBILITY

THEOREM 17.4.5 *Bruggesser and Mani* (1970)

Boundary complexes of polytopes are shellable.

The proof of Bruggesser and Mani is based on starting with a point near the center of a facet and moving from this point to infinity, and back from the other direction, keeping track of the order in which facets are seen. This proves a stronger form of shellability, in which each K_i is the complex spanned by all the facets that can be seen from a particular point in $\mathbb{R}^d$. It follows from shellability that

THEOREM 17.4.6

Polytopes are collapsible.

THEOREM 17.4.7 *Ziegler* (1992)

There are d-polytopes, $d \geq 4$, whose boundary complexes are not extendably shellable.

FACET-FORMING POLYTOPES AND SMALL LOW-DIMENSIONAL FACES

THEOREM 17.4.8 *Perles and Shephard* (1967)

Let P be a d-polytope such that the maximum number of k-faces of P on any $(d-2)$-sphere in the skeleton of P is at most $(d-1-k)/(d+1-k)f_k(P)$. Then P is a nonfacet.

THEOREM 17.4.9 *Schulte* (1985)

The cuboctahedron and the icosidodecahedron are nonfacets.

PROBLEM 17.4.10

Is the icosahedron facet-forming?

For all other regular polytopes the situation is known. The simplices and cubes in any dimension and the 3-dimensional octahedron are facet-forming. All other regular polytopes with the exception of the icosahedron are known to be nonfacets.

Next, we try to understand if it is possible for all the k-faces of a d-polytope to be isomorphic to a given polytope P. The following conjecture asserts that if d is large with respect to k, this can happen only if P is either a simplex or a cube.

CONJECTURE 17.4.11 *Kalai*

For every k there is a $d(k)$ such that every d-polytope with $d > d(k)$ has a k-face that is either a simplex or combinatorially isomorphic to a k-dimensional cube.

For simple polytopes, it follows from the next theorem that if $d > ck^2$ then every d-polytope has a k-face F such that $f_r(F) \leq f_r(C_k)$. (Here, C_k denotes the k-dimensional cube.)

THEOREM 17.4.12 *Nikulin* (1981)

The average number of r-dimensional faces of a k-dimensional face of a simple d-dimensional polytope is at most

$$\binom{d-r}{d-k} \cdot \left(\left(\binom{\lfloor d/2 \rfloor}{r} + \binom{\lfloor (d+1)/2 \rfloor}{r} \right) \Big/ \left(\binom{\lfloor d/2 \rfloor}{k} + \binom{\lfloor (d+1)/2 \rfloor}{k} \right) \right).$$

THEOREM 17.4.13 *Kalai* (1989)

Every d-polytope for $d \geq 5$ has a 2-face with at most 4 vertices.

THEOREM 17.4.14 *Meisinger* (1994)

Every rational d-polytope for $d \geq 9$ has a 3-face with at most 150 vertices.

The last two theorems and the next one are proved using the linear inequalities for flag numbers that are known via intersection homology of toric varieties; see Chapter 15. One can also study, in a similar fashion, quotients of polytopes.

CONJECTURE 17.4.15 *Perles*

For every k there is a $d'(k)$ such that every d-polytope with $d > d'(k)$ has a k-dimensional quotient that is a simplex.

As was mentioned in the first section, $d'(2) = 3$. The 24-cell, which is a regular 4-polytope all whose faces are octahedra, shows that $d'(3) > 4$.

THEOREM 17.4.16 *Meisinger* (1994)

Every d-polytope with $d \geq 9$ has a 3-dimensional quotient that is a simplex.

PROBLEM 17.4.17

For which values of k and r are there d-polytopes other than the d-simplex that are both k-simplicial and r-simple?

It is known that this can happen only when $k+r \leq d$. There are infinite families of $(d-2)$-simplicial and 2-simple polytopes, and some examples of $(d-3)$-simplicial and 3-simple d-polytopes.

RECONSTRUCTION

THEOREM 17.4.18 *Extension of Whitney's theorem*

d-polytopes are determined by their $(d-2)$-skeletons.

THEOREM 17.4.19 *Perles* (1973)

Simplicial d-polytopes are determined by their $\lfloor d/2 \rfloor$-skeletons.

This follows from the following theorem (here, $\mathrm{ast}(F, P)$ is the complex formed by the faces of P that are disjoint to all vertices in F).

THEOREM 17.4.20 *Perles* (1973)

Let P be a simplicial d-polytope.

(i) *If F is a k-face of P, then* $\mathrm{skel}_{d-k-2}(\mathrm{ast}(F,P))$ *is contractible in* $\mathrm{skel}_{d-k-1}(\mathrm{ast}(F,P))$.

(ii) *If F is an empty k-simplex, then* $\mathrm{ast}(F,P)$ *is homotopically equivalent to S^{d-k}; hence,* $\mathrm{skel}_{d-k-2}(\mathrm{ast}(F,P))$ *is not contractible in* $\mathrm{skel}_{d-k-1}(\mathrm{ast}(F,P))$.

THEOREM 17.4.21 *Blind and Mani* (1987)

Simple polytopes are determined by their graphs.

THEOREM 17.4.22 *Kalai and Perles* (1988)

Simplicial d-polytopes are determined by the incidence relations between i- and $(i+1)$-faces for every $i > \lfloor d/2 \rfloor$.

THEOREM 17.4.23 *Bjorner, Edelman, and Ziegler* (1990)

Zonotopes are determined by their graphs.

In all instances of the above theorems except the single case of the Blind-Mani theorem, the proofs give reconstruction algorithms that are polynomial in the data. It is an open question if a polynomial algorithm exists to determine a simple polytope from its graph.

PROBLEM 17.4.24

Is there an e-dimensional polytope other than the d-cube with the same graph as the d-cube?

If we go beyond the class of polytopes and consider polyhedral spheres, then:

THEOREM 17.4.25 *Babson, Billera, and Chan*

For every $e \geq 4$ there is an e-dimensional cubical sphere with 2^d vertices whose $\lfloor e/2 \rfloor$-skeleton is combinatorially isomorphic to the $\lfloor e/2 \rfloor$-skeleton of a d-dimensional cube.

Another issue of reconstruction for polytopes that was studied extensively is the following: In which cases does the combinatorial structure of a polytope determine its geometric structure (up to projective transformations)? Such polytopes are called ***projectively unique***, and the major unsolved problem is:

PROBLEM 17.4.26

Are there only finitely many projectively unique polytopes in each dimension?

McMullen constructed projectively unique d-polytopes with $3^{d/3}$ vertices.

EMPTY FACES AND POLYTOPES WITH FEW VERTICES

THEOREM 17.4.27 *Perles* (1970)

Let $f(d,k,b)$ be the number of combinatorial types of k-skeletons of d-polytopes with $d+b+1$ vertices. Then, for fixed b and k, $f(d,k,b)$ is bounded.

This follows from

THEOREM 17.4.28 *Perles* (1970)

The number of empty i-pyramids for d-polytopes with $d+b$ vertices is bounded by a function of i and b.

For a d-polytope P, let $e_i(P)$ denote the number of empty i-simplices of P.

PROBLEM 17.4.29

Characterize the sequence of numbers $(e_1(P), e_2(P), \ldots, e_d(P))$ arising from simplicial d-polytopes and from general d-polytopes.

CONJECTURE 17.4.30 *Kalai, Kleinschmidt, and Lee*

For all simplicial d-polytopes with prescribed h-vector $h = (h_0, h_1, \ldots, h_d)$, the number of i-dimensional empty simplices is maximized by the Billera-Lee polytopes $P_{BL}(h)$.

$P_{BL}(h)$ is the polytope constructed by Billera and Lee (see Chapter 15) in their proof of the sufficiency part of the g-theorem. It is quite possible that the conjecture applies also to general polytopes.

17.5 CONCLUDING REMARKS AND EXTENSIONS TO MORE GENERAL OBJECTS

The reader who compares this chapter with other chapters on convex polytopes may notice the sporadic nature of the results and problems described here. Indeed, it seems that our main limits in understanding the combinatorial structure of polytopes still lie in our ability to raise the right questions. Another feature that comes to mind (and is not unique to this area) is the lack of examples, methods of constructing them, and means of classifying them.

We have considered mainly properties of general polytopes and of simple or simplicial polytopes. There are many classes of polytopes that are either of intrinsic interest from the combinatorial theory of polytopes, or that arise in various other fields, for which the problems described in this chapter are interesting.

Most of the results of this chapter extend to much more general objects than convex polytopes. Finding combinatorial settings for which these results hold is an interesting and fruitful area. On the other hand, the results described here are not sufficient to distinguish polytopes from larger classes of polyhedral spheres, and finding delicate combinatorial properties that distinguish polytopes is an important area of research. Few of the results on skeletons of polytopes extend to skeletons of other convex bodies, and relating the combinatorial theory of polytopes with other aspects of convexity is a great challenge.

17.6 SOURCES AND RELATED MATERIAL

FURTHER READING

Grünbaum [Grü75] is a survey on polytopal graphs with many references (most

references for the results in Section 17.1 and related results can be found there). More material on the topic of this chapter can also be found in [Grü67], [Zie95], [KK95], and [BL93]. The references in Ziegler's book [Zie95] and their electronic updates cover most of the results described in this chapter. For a survey on the Hirsch conjecture and its relatives, see [KK87]. More recent results can be found in Kleinschmidt's chapter in [BMSW94] and in [Zie95, Section 3.3]. Related results on simplex algorithms are mentioned in Chapter 38 of this Handbook.

Martini's chapter in [BMSW94] is on the regularity properties of polytopes (a topic not covered here; cf. Chapter 16), and contains references on facet-forming polytopes and nonfacets. The original papers on facet-forming polytopes and nonfacets contain many more results, and describe relations to questions on tiling spaces with polyhedra. Other chapters of [BMSW94] are also relevant.

RELATED CHAPTERS

Chapter 13: Basic properties of convex polytopes
Chapter 15: Face numbers of polytopes and complexes
Chapter 38: Linear programming in low dimensions
Chapter 39: Mathematical programming

REFERENCES

[BL93] M.M. Bayer and C.W. Lee. Combinatorial aspects of convex polytopes. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 485–534. North-Holland, Amsterdam, 1993.

[BMSW94] T. Bisztriczky, P. McMullen, R. Schneider, and A. Ivić Weiss, editors. *Polytopes: Abstract, Convex and Computational.* Volume 440 of *NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci.* Kluwer, Dordrecht, 1994.

[Grü67] B. Grünbaum. *Convex Polytopes.* Interscience, London, 1967; revised edition (V. Klee and P. Kleinschmidt, editors), *Graduate Texts in Math.*, Springer-Verlag, in preparation.

[Grü75] B. Grünbaum. Polytopal graphs. In D.R. Fulkerson, editor, *Studies in Graph Theory*, pages 201–224. Math. Assoc. America, Washington, 1975.

[KK87] V. Klee and P. Kleinschmidt. The d-step conjecture and its relatives. *Math. Oper. Res.*, 12:718–755, 1987.

[KK95] V. Klee and P. Kleinschmidt. Polyhedral complexes and their relatives. In R. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, pages 875–917. North-Holland, Amsterdam, 1995.

[PA95] J. Pach and P.K. Agarwal. *Combinatorial Geometry.* Wiley-Interscience, New York, 1995.

[Zie95] G.M. Ziegler. *Lectures on Polytopes.* Volume 152 of *Graduate Texts in Math.*, Springer-Verlag, New York, 1995.

18 POLYHEDRAL MAPS

Ulrich Brehm and Egon Schulte

INTRODUCTION

Historically, polyhedral maps on surfaces made their first appearance as convex polyhedra. The famous Kepler-Poinsot (star) polyhedra marked the first occurrence of maps on orientable surfaces of higher genus (namely 4), and started the branch of topology dealing with regular maps. Further impetus to the subject came from the theory of automorphic functions and from the Four-Color-Problem (Coxeter and Moser [CM80], Barnette [Bar83]).

A more systematic investigation of general polyhedral maps and nonconvex polyhedra began only around 1970, and was inspired by Grünbaum's book "Convex Polytopes" [Grü67]. Since then, the subject has grown into an active field of research on the interfaces of convex and discrete geometry, graph theory, and combinatorial topology. The underlying topology is mainly elementary, and many basic concepts and constructions are inspired by convex polytope theory.

18.1 POLYHEDRA

Tessellations on surfaces are natural objects of study in topology that generalize convex polyhedra and plane tessellations. For general properties of convex polyhedra, polytopes, and tessellations, see Grünbaum [Grü67], Coxeter [Cox73], and Grünbaum and Shephard [GS86], or Chapters 3, 13, 14, 15, and 16 of this Handbook. For a survey on polyhedral manifolds see Brehm and Wills [BW93], which also has an extensive list of references. The long list of definitions that follows places polyhedral maps in the general context of topological and geometric complexes. For an account of 2- and 3-dimensional geometric topology, see Moise [Moi77].

GLOSSARY

Polyhedral complex: A finite set Γ of convex polytopes, the ***faces*** of Γ, in real n-space $\mathbb{R}^n$, such that two conditions are satisfied. First, if $Q \in \Gamma$ and F is a face of Q, then $F \in \Gamma$. Second, if $Q_1, Q_2 \in \Gamma$, then $Q_1 \cap Q_2$ is a face of Q_1 and Q_2 (possibly the empty face $\emptyset$). The subset $||\Gamma|| := \bigcup_{Q \in \Gamma} Q$ of $\mathbb{R}^n$, equipped with the induced topology, is called the ***underlying space*** of Γ. The ***dimension*** $d := \dim \Gamma$ of Γ is the maximum of the dimensions (of the affine hulls) of the elements in Γ. We also call Γ a ***polyhedral d-complex***. A face of Γ of dimension 0, 1, or i is a ***vertex***, an ***edge***, or an ***i-face*** of Γ. A face that is maximal (with respect to inclusion) is called a ***facet*** of Γ. (In our applications, the facets are just the d-faces of Γ.)

Face poset: The set $P(\Gamma)$ of all faces of Γ, partially ordered by inclusion. As a partially ordered set, $P(\Gamma) \cup \{||\Gamma||\}$ is a ranked lattice.

(Geometric) simplicial complex: A polyhedral complex Γ all of whose nonempty faces are simplices. An ***abstract simplicial complex*** Δ is a family of subsets of a finite set V, the ***vertex set*** of Δ, such that $\{x\} \in \Delta$ for all $x \in V$, and such that $F \subseteq G \in \Delta$ implies $F \in \Delta$. Each abstract simplicial complex Δ is isomorphic (as a poset ordered by inclusion) to the face poset of a geometric simplicial complex Γ. Once such an isomorphism is fixed, we set $||\Delta|| := ||\Gamma||$, and the terminology introduced for Γ carries over to Δ. (One often omits the qualifications "geometric" or "abstract.")

Link: The link of a vertex x in a simplicial complex Γ is the subcomplex consisting of the faces that do not contain x of all the faces of Γ containing x.

Polyhedron: A subset P of $\mathbb{R}^n$ such that $P = ||\Gamma||$ for some polyhedral complex Γ. In general, given P, there is no canonical way to associate with it the complex Γ. However, once Γ is specified, the terminology for Γ regarding $P(\Gamma)$ is also carried over to P.

Subdivision: If Γ_1 and Γ_2 are polyhedral complexes, Γ_1 is a subdivision of Γ_2 if $||\Gamma_1|| = ||\Gamma_2||$ and each face of Γ_1 is a subset of a face of Γ_2. If Γ_1 is a simplicial complex, this is a ***simplicial subdivision.***

Combinatorial d-manifold: For $d = 1$, this is a simplicial 1-complex Δ such that $||\Delta||$ is a 1-sphere. Inductively, if $d \geq 2$, it is a simplicial d-complex Δ such that $||\Delta||$ is a topological d-manifold (without boundary) and each vertex link is a ***combinatorial $(d-1)$-sphere*** (that is, a combinatorial $(d-1)$-manifold whose underlying space is a $(d-1)$-sphere).

Polyhedral d-manifold: A polyhedral d-complex Γ having a simplicial subdivision that is a combinatorial d-manifold. If $d = 2$, this is simply a polyhedral 2-complex Γ for which $||\Gamma||$ is a compact 2-manifold (without boundary).

Triangulation: A triangulation (simplicial decomposition) of a topological space X is a simplicial complex Γ such that X and $||\Gamma||$ are homeomorphic.

Ball complex: A finite family $\mathcal{C}$ of topological balls (homeomorphic images of Euclidean unit balls) in a Hausdorff space, the ***underlying space*** $||\mathcal{C}||$ of $\mathcal{C}$, whose relative interiors partition $||\mathcal{C}||$ in such a way that the boundary of each ball in $\mathcal{C}$ is the union of other balls in $\mathcal{C}$. The ***dimension*** of $\mathcal{C}$ is the maximum of the dimensions of the balls in $\mathcal{C}$.

Embedding: For a ball complex $\mathcal{C}$, a continuous mapping $\gamma : ||\mathcal{C}|| \to \mathbb{R}^n$ that is a homeomorphism of $||\mathcal{C}||$ onto its image. $\mathcal{C}$ is said to be ***embedded*** in $\mathbb{R}^n$.

Polyhedral embedding: For a ball complex $\mathcal{C}$, an embedding γ that maps each ball in $\mathcal{C}$ onto a convex polytope.

Immersion: For a ball complex $\mathcal{C}$, a continuous mapping $\gamma : ||\mathcal{C}|| \to \mathbb{R}^n$ that is locally injective (hence the image may have self-intersections). $\mathcal{C}$ is said to be ***immersed*** in $\mathbb{R}^n$.

Polyhedral immersion: For a ball complex $\mathcal{C}$, an immersion γ that maps each ball in $\mathcal{C}$ onto a convex polytope.

Map on a surface: An embedded finite graph M (without loops or multiple edges) on a compact 2-manifold (surface) S such that two conditions are satisfied: The closures of the connected components of $S \setminus M$, the ***faces*** of M, are closed

2-cells (closed topological disks), and each vertex of M has valency at least 3. (Note that some authors use a broader definition of maps; e.g., see [CM80].)

Polyhedral map: A map M on S such that the intersection of any two distinct faces is either empty, a common vertex, or a common edge.

Figure 18.1.1 shows a polyhedral map on a surface of genus 3, known as ***Dyck's regular map***. We will discuss this map further in Sections 18.4 and 18.5.

Type: A map M on S is of type $\{p, q\}$ if all its faces are topological p-gons such that q meet at each vertex. The symbol $\{p, q\}$ is the ***Schläfli symbol*** for M.

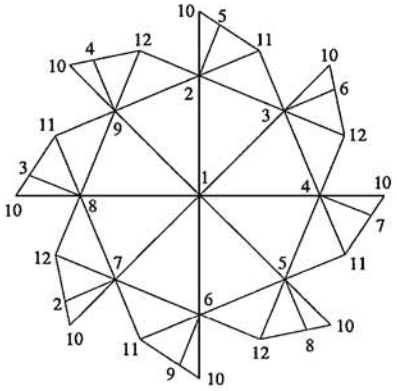


FIGURE 18.1.1
Dyck's regular map, of type $\{3, 8\}$.
Vertices with the same label are identified.

BASIC RESULTS

Simplicial complexes are important in topology, geometry, and combinatorics. Each abstract simplicial d-complex Δ with n vertices is isomorphic to the face poset of a geometric simplicial complex Γ in $\mathbb{R}^{2d+1}$ that is the image under a projection (Schlegel diagram—see Chapter 13) of a simplicial d-subcomplex in the boundary complex of the cyclic convex $(2d+2)$-polytope $C(n, 2d + 2)$ with n vertices; see [Grü67] or Chapter 13 of this Handbook.

Let C be a ball complex and $P(C)$ the associated poset (i.e., C ordered by inclusion). Let $\Delta(P(C))$ denote the ***order complex*** of $P(C)$; that is, the simplicial complex whose vertex set is C and whose k-faces are the k-chains $x_0 < x_1 < \ldots < x_k$ in $P(C)$. Then $||C||$ and $||\Delta(P(C))||$ are homeomorphic. This means that the poset $P(C)$ already carries complete topological information about $||C||$. See [Bjö95] or [BW93], as well as Chapter 15, for further information.

Each polyhedral d-complex is a d-dimensional ball complex. The set C of vertices, edges, and faces of a map M on a 2-manifold S is a 2-dimensional ball complex. In particular, a map M is a polyhedral map if and only if the intersection of any two elements of C is empty or an element of C. A map is usually identified with its poset of vertices, edges, and faces, ordered by inclusion. If M is a polyhedral map, then this poset is a lattice when augmented by $\emptyset$ and S as smallest and largest elements. The dual lattice (obtained by reversing the order) again gives a polyhedral map, the ***dual map***, on the same 2-manifold S. Note that in the context of polyhedral maps, the qualification "polyhedral" does not mean that it can be realized as a polyhedral complex. However, a polyhedral 2-manifold can always be regarded as a polyhedral map.

An important problem is the following:

PROBLEM 18.1.1 *General Embeddability Problem*

When is a given finite poset isomorphic to the face poset of some polyhedral complex in a given space $\mathbb{R}^n$? When can a ball complex be polyhedrally embedded or polyhedrally immersed in $\mathbb{R}^n$?

These questions are different from the embeddability problems that are discussed in piecewise-linear topology, because simplicial subdivisions are excluded. A complete answer is available only for the face posets of spherical maps:

THEOREM 18.1.2 *Steinitz's Theorem*

Each polyhedral map M on the 2-sphere is isomorphic to the boundary complex of a convex 3-polytope. Equivalently, a finite graph is the edge graph of a convex 3-polytope if and only if it is planar and 3-connected (it has at least 4 vertices and the removal of any 2 vertices leaves a connected graph).

Very little is known about polyhedral embeddings of orientable polyhedral maps of positive genus g. For example, for each $g > 1$ it is an open question whether or not each simplicial polyhedral map on the orientable closed surface of genus g can be polyhedrally embedded in 3-space $\mathbb{R}^3$. However, there are some general necessary combinatorial conditions for the existence of polyhedral embeddings of polyhedral maps in n-space $\mathbb{R}^n$ [BGH91]. Each nonorientable closed surface can be immersed but not embedded in $\mathbb{R}^3$. However, the Möbius strip and therefore each nonorientable surface can be triangulated in such a way that the resulting simplicial polyhedral map cannot be polyhedrally immersed in $\mathbb{R}^3$ [BW93]. On the other hand, each triangulation of the torus or the real projective plane $\mathbb{R}\mathrm{P}^2$ can be polyhedrally embedded in $\mathbb{R}^4$ [BS95].

Another important type of problem asks for topological properties of the space of all polyhedral embeddings, or of all convex d-polytopes, with a given face lattice. This is the ***realization space*** for this lattice. Every convex 3-polytope has an open ball as its realization space. However, the realization spaces of convex 4-polytopes can be arbitrarily complicated; see the "Universality Theorem" by Richter-Gebert [Ric96] in Chapter 13 of this Handbook.

For further embeddability results in higher dimensions, as well as for a discussion of some related problems such as the polytopality problems and isotopy problems, see [Zie95, BLS+93, BW93]. For a computational approach to the embeddability problem in terms of oriented matroids, see Bokowski and Sturmfels [BS89], as well as Chapter 6 of this Handbook. We shall revisit the embeddability problem in Sections 18.2 and 18.5 for interesting special classes of polyhedral maps.

Many interesting maps M on compact surfaces S have a Schläfli symbol $\{p, q\}$; for examples, see Section 18.4. These maps can then be obtained from the regular tessellation $\{p, q\}$ of the 2-sphere, the Euclidean plane, or the hyperbolic plane by making identifications. Trivially, $qf_0 = 2f_1 = pf_2$. Also, if the Euler characteristic χ of S is negative and m denotes the number of flags (incident triples consisting of a vertex, an edge, and a face) of M, then

$$\chi = f_0 - f_1 + f_2 = \frac{m}{2}\left(\frac{1}{q} - \frac{1}{2} + \frac{1}{p}\right) \leq -\frac{m}{84}\,, \tag{18.1.1}$$

and equality holds on the right-hand side if and only if M is of type $\{3, 7\}$ or $\{7, 3\}$.

18.2 EXTREMAL PROPERTIES

There is a natural interest in polyhedral maps and polyhedra defined by certain minimality properties. For relations with the famous Map Color Theorem, which gives the minimum genus of a surface on which the complete graph K_n can be embedded, see Ringel [Rin74] and Barnette [Bar83]. See also Brehm and Wills [BW93].

GLOSSARY

f-vector: For a map M, the vector $f(M) = (f_0, f_1, f_2)$, where f_0, f_1, f_2 are the numbers of vertices, edges, and faces of M, respectively.

Weakly neighborly: A polyhedral map is weakly neighborly (a ***wnp map***) if any two vertices lie in a common face.

Neighborly: A map is neighborly if any two vertices are joined by an edge.

Nonconvex vertex: A vertex x of a polyhedral 2-manifold M in $\mathbb{R}^3$ is a ***convex vertex*** if at least one of the two components into which M divides a small convex neighborhood of x in $\mathbb{R}^3$ is convex; otherwise, x is nonconvex.

Tight polyhedral 2-manifold: A polyhedral 2-manifold M embedded in $\mathbb{R}^3$ such that every hyperplane strictly supporting M locally at a point supports M globally.

BASIC RESULTS

THEOREM 18.2.1

Let M be a polyhedral map of Euler characteristic χ with f-vector (f_0, f_1, f_2). Then

$$f_0 \geq \lceil (7 + \sqrt{49 - 24\chi})/2 \rceil. \tag{18.2.1}$$

Here, $\lceil t \rceil$ denotes the smallest integer greater than or equal to t. This lower bound is known as the ***Heawood bound*** and is an easy consequence of Euler's formula $f_0 - f_1 + f_2 = \chi$ $(= 2 - 2g$ if M is orientable of genus $g)$.

THEOREM 18.2.2

Except for the nonorientable 2-manifolds with $\chi = 0$ (Klein bottle) or $\chi = -1$ and the orientable 2-manifold of genus $g = 2$ ($\chi = -2$), each 2-manifold admits a triangulation for which the lower bound (18.2.1) *is attained.*

This is closely related to the Map Color Theorem. The same lower bound (18.2.1) holds for the number f_2 of faces of M, since the dual of M is a polyhedral map with the same Euler characteristic and with f-vector (f_2, f_1, f_0).

The exact minimum for the number f_1 of edges of a polyhedral map is known for only some manifolds. For the orientable 2-manifolds of genus $g = 0$, 1, 5, or 14, this minimal number is 6, 18, 40, or 78, respectively; and for the nonorientable 2-manifolds of Euler characteristic $\chi = 1$, 0, -1, or -2, it is 15, 18, 23, or 26,

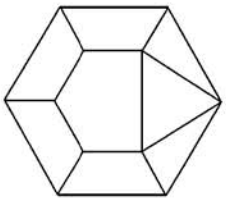


FIGURE 18.2.1
A self-dual polyhedral map on $\mathbb{R}\mathrm{P}^2$ *with the minimum number* (15) *of edges.*

respectively. For a map on $\mathbb{R}\mathrm{P}^2$ with 15 edges, see Figure 18.2.1. A general bound is given by

THEOREM 18.2.3 [Bre90]

$$f_1 \geq -\chi + \min\{y \in \mathbb{N} \mid y(\sqrt{2y} - 6) \geq -8\chi \text{ and } y \geq 8\}\,,$$

where $\mathbb{N}$ *is the set of natural numbers.*

If M is a polyhedral map on a surface S, then a new polyhedral map M' on S can be obtained from M by the following operation, called ***face splitting***. A new edge xy is added across a face of M, where x and y are points on edges of M that are not contained in a common edge. The new vertices x and y of M' may be vertices of M, or one or both may be relative interior points of edges of M. The dual operation is called ***vertex splitting***. On the sphere S^2, the (boundary complex of the) tetrahedron is the only polyhedral map that is minimal with respect to face splitting. On the real projective plane $\mathbb{R}\mathrm{P}^2$, there are exactly 16 polyhedral maps that are minimal with respect to face splitting [Bar91], and exactly 7 that are minimal with respect to both face splitting and vertex splitting. These are exactly the polyhedral maps on $\mathbb{R}\mathrm{P}^2$ with 15 edges, which is the minimum number of edges for $\mathbb{R}\mathrm{P}^2$. For an example, see Figure 18.2.1.

For neighborly polyhedral maps we always have equality in (18.2.1). Weakly neighborly polyhedral maps (wnp maps) are a generalization of neighborly polyhedral maps. On the 2-sphere, the only wnp maps are the (boundary complexes of the) pyramids and the triangular prism. Every other 2-manifold admits only finitely many combinatorially distinct wnp maps. Moreover,

$$\limsup_{\chi \to \infty} V_{\max}(\chi) \cdot |2\chi|^{-2/3} \leq 1\,,$$

where $V_{\max}(\chi)$ denotes the maximum number of vertices of a wnp map of Euler characteristic χ; see [BA86], which also discusses further equalities and inequalities for general polyhedral maps. For several 2-manifolds, all wnp maps have been determined. For example, on the torus there are exactly five wnp maps, and three of them are geometrically realizable as polyhedra in $\mathbb{R}^3$.

In some instances, the combinatorial lower bound (18.2.1) can also be attained geometrically by (necessarily orientable) polyhedra in $\mathbb{R}^3$. Trivially, the tetrahedron minimizes f_0 $(= 4)$ for $g = 0$. For $g = 1$ there is a polyhedron with $f_0 = 7$ known as the ***Császár torus***; see Figure 18.2.2. A pair of congruent copies of the torus shown in Figure 18.2.2b can be linked (if the coordinates orthogonal to the plane of projection are sufficiently small). Polyhedra that have the minimum number of vertices have also been found for $g = 2$ (the exceptional case), 3, or 4, with 10, 10, or 11 vertices, respectively.

The minimum number of vertices for polyhedral maps that admit polyhedral immersions in $\mathbb{R}^3$ is 9 for both the real projective plane $\mathbb{R}\mathrm{P}^2$ and the Klein bottle.

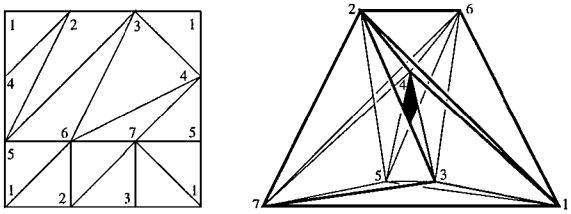


FIGURE 18.2.2
(a) *The unique 7-vertex triangulation of the torus and* (b) *a symmetric realization as a polyhedron.*

The lower bound for $\mathbb{R}P^2$ follows directly from the fact that each immersion of $\mathbb{R}P^2$ in $\mathbb{R}^3$ has (generically) a triple point (like the classical Boy surface).

There are also some surprising results for higher genus. For example, for each $q \geq 3$ there exists a polyhedral map M_q of type $\{4, q\}$ with $f_0 = 2^q$ and $g = 2^{q-3}(q-4) + 1$ such that M_q and its dual have polyhedral embeddings in $\mathbb{R}^3$. These polyhedra are combinatorially regular in the sense of Section 18.5. Note that $f_0 = O(g/\log g)$. Thus for sufficiently large genus, M_q has more handles than vertices, and its dual has more handles than faces.

Every polyhedral 2-manifold in $\mathbb{R}^3$ of genus $g \geq 1$ contains at least 5 nonconvex vertices. This bound is attained for each $g \geq 1$. For tight polyhedral 2-manifolds, the lower bound for the number of nonconvex vertices is larger and depends on g. For a survey on tight polyhedral submanifolds see [Küh95].

18.3 EBERHARD'S THEOREM AND RELATED RESULTS

Eberhard's theorem is one of the oldest nontrivial results about convex polyhedra. The standard reference is Grünbaum [Grü67, Grü70]. For recent developments see also Jendrol [Jen93].

GLOSSARY

***p*-sequence:** For a polyhedral map M, the sequence $p(M) = (p_k(M))_{k\geq 3}$, where $p_k = p_k(M)$ is the number of k-gonal faces of M.

***v*-sequence:** For a polyhedral map M, the sequence $v(M) = (v_k(M))_{k\geq 3}$, where $v_k = v_k(M)$ is the number of vertices of M of degree k.

EBERHARD-TYPE RESULTS

Significant results are known for the general problem of determining what kind of polygons, and how many of each kind, may be combined to form the faces of a polyhedral map M on an orientable surface of genus g. These refine results (for $d = 3$) about the boundary complex and the number of i-dimensional faces ($i = 0, \ldots, d-1$) of a convex d-polytope [Grü67, Zie95]; see Chapter 15.

If M is a polyhedral map of genus g with f-vector (f_0, f_1, f_2), then

$$\sum_{k\geq 3} p_k = f_2, \quad \sum_{k\geq 3} v_k = f_0, \quad \sum_{k\geq 3} kp_k = 2f_1 = \sum_{k\geq 3} kv_k\,. \tag{18.3.1}$$

Further, Euler's formula $f_0 - f_1 + f_2 = 2(1-g)$ implies the equations

$$\sum_{k\geq 3}(6-k)p_k + 2\sum_{k\geq 3}(3-k)v_k = 12(1-g) \tag{18.3.2}$$

and

$$\sum_{k\geq 3}(4-k)(p_k+v_k) = 8(1-g)\,. \tag{18.3.3}$$

These equations contain no information about p_6, v_3 and p_4, v_4, respectively.

Eberhard-type results deal with the problem of determining which pairs $(p_k)_{k\geq 3}$ and $(v_k)_{k\geq 3}$ of sequences of nonnegative integers can occur as p-sequences $p(M)$ and v-sequences $v(M)$ of polyhedral maps M of a given genus g. The above equations yield simple necessary conditions. As a consequence of Steinitz's theorem (Section 18.1), the problem for $g = 0$ is equivalent to a similar such problem for convex 3-polytopes [Grü67, Grü70]. The classical theorem of Eberhard says the following:

THEOREM 18.3.1 *Eberhard's Theorem*

For each sequence $(p_k \mid 3 \leq k \neq 6)$ of nonnegative integers satisfying

$$\sum_{k\geq 3}(6-k)p_k = 12,$$

there exist values of p_6 such that the sequence $(p_k)_{k\geq 3}$ is the p-sequence of a spherical polyhedral map all of whose vertices have degree 3, or, equivalently, of a convex 3-polytope that is simple (has vertices only of degree 3).

This is the case $g = 0$ and $v_3 = f_0$, $v_k = 0$ $(k \geq 4)$.

More general results have been established [Jen93]. Given two sequences $p' = (p_k \mid 3 \leq k \neq 6)$ and $v' = (v_k \mid k > 3)$ of nonnegative integers such that the equation (18.3.2) is satisfied for a given genus g, let $E(p', v'; g)$ denote the set of integers $p_6 \geq 0$ such that $(p_k)_{k\geq 3}$ and $(v_k)_{k\geq 3}$, with $v_3 := (\sum_{k\geq 3} kp_k - \sum_{k\geq 4} kv_k)/3$ determined by (18.3.1), are the p-sequences and v-sequences, respectively, of a polyhedral map of genus g. For all but two admissible triples (p', v', g), the set $E(p', v'; g)$ is known up to a finite number of elements. For example, for $g = 0$, the set $E(p', v'; 0)$ is nonempty if and only if $\sum_{k/6\equiv(\mathrm{mod}\,3)} v_k \neq 1$ or $p_k \neq 0$ for at least one odd k. In particular, for each such nonempty set, there exists a constant c depending on (p', v') such that $E(p', v'; 0) = \{j \mid c \leq j\}$, $\{j \mid c \leq j \equiv 0\,(\mathrm{mod}\,2)\}$, or $\{j \mid c \leq j \equiv 1\,(\mathrm{mod}\,2)\}$. Similarly, for each triple with $g \geq 2$, there is a constant c depending on (p', v', g) such that $E(p', v'; g) = \{j \mid c \leq j\}$. There are analogous results for sequences $(p_k \mid 3 \leq k \neq 4)$ and $(v_k \mid 3 \leq k \neq 4)$ that satisfy the equation (18.3.3) or other related equations.

For $g = 1$ there is also a more geometric Eberhard-type result available, which requires the polyhedral map M to be polyhedrally embedded in $\mathbb{R}^3$:

THEOREM 18.3.2 [Gri83]

Let s, p_k $(k \geq 3, k \neq 6)$ be nonnegative integers. Then there exists a toroidal polyhedral 2-manifold M in $\mathbb{R}^3$ with $p_k(M) = p_k$ $(k \neq 6)$ and $\sum_{k\geq 3}(k-3)v_k(M) = s$ if and only if $\sum_{k\geq 3}(6-k)p_k = 2s$ and $s \geq 6$.

Also, for toroidal polyhedral 2-manifolds in $\mathbb{R}^3$ (as well as for convex 3-polytopes), the exact range of possible f-vectors is known [Grü67, BW93].

THEOREM 18.3.3

A polyhedral embedding of the torus in $\mathbb{R}^3$ *with* f*-vector* (f_0, f_1, f_2) *exists if and only if* $f_0 - f_1 + f_2 = 0$, $f_2(11 - f_2)/2 \leq f_0 \leq 2f_2$, $f_0(11 - f_0)/2 \leq f_2 \leq 2f_0$, *and* $2f_1 - 3f_0 \geq 6$.

For generalizations of Eberhard's theorem to tilings of the Euclidean plane, see also [GS86].

18.4 REGULAR MAPS

Regular maps are topological analogues of the ordinary regular polyhedra and star-polyhedra on surfaces. Historically they became important in the context of transformations of algebraic equations and representations of algebraic curves in homogeneous complex variables. There is a large body of literature on regular maps and their groups. The classical text is Coxeter and Moser [CM80]. For a recent text see McMullen and Schulte [MS].

GLOSSARY

(Combinatorial) automorphism: An incidence-preserving bijection (of the set of vertices, edges, and faces) of a map M on a surface S to itself. The (***combinatorial automorphism***) ***group*** $A(M)$ of M is the group of all such bijections. It can be "realized" by a group of homeomorphisms of S.

Regular map: A map M on S whose group $A(M)$ is transitive on the flags (incident triples consisting of a vertex, an edge, and a face) of M.

GENERAL RESULTS

Each regular map M is of type $\{p, q\}$ for some finite p and q. Its group $A(M)$ is transitive on the vertices, the edges, and the faces of M. In general, the Schläfli symbol $\{p, q\}$ does not determine M uniquely. The group $A(M)$ is generated by involutions ρ_0, ρ_1, ρ_2 such that the ***standard relations***

$$\rho_0{}^2 = \rho_1{}^2 = \rho_2{}^2 = (\rho_0\rho_1)^p = (\rho_1\rho_2)^q = (\rho_0\rho_2)^2 = 1$$

hold, but in general there are also further independent relations. Any triangle in the "barycentric subdivision" (order complex) of M is a fundamental region for $A(M)$ on the underlying surface S; see Section 18.1. For any fixed such triangle, we can take for ρ_i the "combinatorial reflection" in its side opposite to the vertex that corresponds to an i-dimensional element of M. The set of standard relations gives a presentation for the symmetry group of the regular tessellation $\{p, q\}$ on the 2-sphere, in the Euclidean plane, or in the hyperbolic plane, whichever is the universal covering of M. See Figure 18.5.1 (a) for a conformal (hyperbolic) drawing of the Dyck map (shown also in Figure 18.1.1) with a fundamental region shaded. The identifications on the boundary of the drawing are indicated by letters.

For orientable surfaces S, the regular maps are known for genus $g \leq 6$. Up to isomorphism, if $g = 0$, there are just the Platonic solids (or regular spherical

tessellations) $\{3,3\}$, $\{3,4\}$, $\{4,3\}$, $\{3,5\}$, and $\{5,3\}$. For $g = 1$, there are three infinite families of torus maps of type $\{3,6\}$, $\{6,3\}$, and $\{4,4\}$, which are quotients of the corresponding Euclidean universal covering tessellations $\{3,6\}$, $\{6,3\}$, or $\{4,4\}$, respectively. For $g \geq 2$, the universal covering tessellation $\{p,q\}$ is hyperbolic and there are only finitely many regular maps on a surface of genus g. The latter follows from the ***Hurwitz formula*** $|A(M)| \leq 84\,|\chi|$ (or from the inequality 18.1.1), where χ is the Euler characteristic of S. Each regular map on a nonorientable surface is doubly covered by a regular map of the same type on an orientable surface, and this covering map is unique [Wil78].

Generally speaking, given M, the topology of S is reflected in the relations that have to be added to the standard relations to obtain a presentation for $A(M)$. Conversely, many interesting regular maps can be constructed by adding certain kinds of extra relations for the group. Two examples are the regular maps $\{p,q\}_r$ and $\{p,q|r\}$ obtained by adding the extra relations $(\rho_0\rho_1\rho_2)^r = 1$ or $(\rho_0\rho_1\rho_2\rho_1)^r = 1$, respectively. Often these are "infinite maps" on noncompact surfaces, but there are also many (finite) maps on compact surfaces. The Dyck map $\{3,8\}_6$ and the famous Klein map $\{3,7\}_8$ (with group $PGL(2,7)$) are both of genus 3 and of the first kind, while the so-called regular skew polyhedra in Euclidean 3-space or 4-space are of the second kind. For more details and further interesting classes of regular maps, see [CM80, MS] and Chapter 16 of this Handbook. In Section 18.5 we shall discuss polyhedral embeddings of regular maps in ordinary 3-space.

The rotation subgroup (orientation preserving subgroup) of the group of an orientable regular map (of type $\{3,7\}$ or $\{7,3\}$) that achieves equality in the Hurwitz formula is also called a ***Hurwitz group***. The Klein map is the regular map of smallest genus whose rotation subgroup is a Hurwitz group [Con90].

18.5 SYMMETRIC POLYHEDRA

Traditionally, much of the appeal of polyhedral 2-manifolds comes from their combinatorial or geometric symmetry properties. For surveys on symmetric polyhedra in $\mathbb{R}^3$ see Schulte and Wills [SW91], Bokowski and Wills [BW88], and Brehm and Wills [BW93].

GLOSSARY

Combinatorially regular: A polyhedral 2-manifold (or polyhedron) P is combinatorially regular if its combinatorial automorphism group $A(P)$ is flag-transitive (or, equivalently, if the underlying polyhedral map is a regular map).

Equivelar: A polyhedral 2-manifold (or polyhedron) P is equivelar of type $\{p,q\}$ if all its 2-faces are convex p-gons and all its vertices are q-valent.

GENERAL RESULTS

See Section 18.4 for results about regular maps. Up to isomorphism, the Platonic solids are the only combinatorially regular polyhedra of genus 0. For the torus, each regular map that is a polyhedral map also admits an embedding in $\mathbb{R}^3$ as a

combinatorially regular polyhedron. Much less is known for maps of genus $g \geq 2$. Two infinite sequences of combinatorially regular polyhedra have been discovered, one consisting of polyhedra of type $\{4, q\}$ ($q \geq 3$) and the other of their duals of type $\{q, 4\}$. These are polyhedral embeddings of the maps M_q and their duals mentioned in Section 18.2. Several famous regular maps have also been realized as polyhedra, including Klein's $\{3, 7\}_8$, Dyck's $\{3, 8\}_6$, and Coxeter's $\{4, 6|3\}$, $\{6, 4|3\}$, $\{4, 8|3\}$, and $\{8, 4|3\}$ [SW91, BS89]. However, a complete classification of combinatorially regular polyhedra is not within reach at present. See Figure 18.5.1 for an illustration of a polyhedral realization of Dyck's regular map $\{3, 8\}_6$ shown in Figure 18.1.1. (a) shows a conformal drawing of the Dyck map, with a fundamental region shaded, while (b) shows a maximally symmetric polyhedral realization.

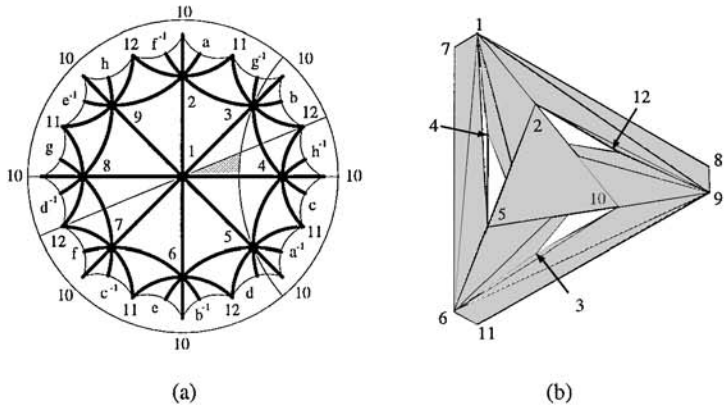


FIGURE 18.5.1
Dyck's regular map:
(a) *a conformal drawing, with fundamental region shaded;*
(b) *a symmetric polyhedral realization.*

For a more general concept of polyhedra in $\mathbb{R}^3$ or higher-dimensional spaces, as well as an enumeration of the corresponding regular polyhedra, see Chapter 16 of this Handbook. The latter also contains a depiction of the polyhedral realization of $\{4, 8|3\}$.

Equivelarity is a local regularity condition. Each combinatorially regular polyhedron in $\mathbb{R}^3$ is equivelar. However, there are many other equivelar polyhedra. For sufficiently large genus g, for example, there are equivelar polyhedra for each of the types $\{3, q\}$ with $q = 7, 8, 9$; $\{4, q\}$ with $q = 5, 6$; and $\{q, 4\}$ with $q = 5, 6$ [BW93].

The symmetry group of a polyhedron can be much smaller than the combinatorial automorphism group of the underlying polyhedral map. In particular, the five Platonic solids are the only polyhedra in $\mathbb{R}^3$ with a flag-transitive symmetry group. However, even for higher genus (namely, for $g = 1, 3, 5, 7, 11$, and 19), polyhedra with a vertex-transitive symmetry group are known. Such a polyhedral torus is shown in Figure 18.5.2.

Finally, if we relax the requirement that a polyhedron be free of self-intersections and allow more general "polyhedral realizations" of maps (for instance, polyhedral immersions), then there is much more flexibility in the construction of "polyhedra" with high symmetry properties. The most famous examples are the Kepler-Poinsot star-polyhedra, but there are also many others. For more details see [SW91, BW88, BW93, MS] and Chapter 16 of this Handbook.

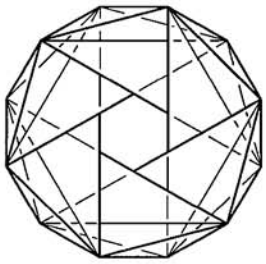


FIGURE 18.5.2
A vertex-transitive polyhedral torus.

18.6 SOURCES AND RELATED MATERIAL

SURVEYS

[Bar83]: A text about colorings of maps and polyhedra.

[Bjö95]: A survey on topological methods in combinatorics.

[BLS+93]: A monograph on oriented matroids.

[BS89]: A text about computational aspects of geometric realizability.

[BW93]: A survey on polyhedral manifolds in 2 and higher dimensions.

[Con90]: A survey on Hurwitz groups.

[Cox73]: A monograph on regular polytopes, regular tessellations, and reflection groups.

[CM80]: A monograph on discrete groups and their presentations.

[Grü67]: A monograph on convex polytopes.

[Grü70]: A survey on convex polytopes complementing the exposition in [Grü67].

[GS86]: A monograph on plane tilings and patterns.

[Küh95]: A survey on tight polyhedral manifolds.

[Moi77]: A text about geometric topology in low dimensions.

[MS]: A monograph on abstract regular polytopes and their groups.

[Rin74]: A text about maps on surfaces and the Map Color Theorem.

[SW91]: A survey on combinatorially regular polyhedra in 3-space.

[Zie95]: A text about convex polytopes also stressing some more recent developments.

RELATED CHAPTERS

Chapter 3: Tilings
Chapter 6: Oriented matroids
Chapter 13: Basic properties of convex polytopes

Chapter 14: Subdivisions and triangulations of polytopes
Chapter 15: Face numbers of polytopes and complexes
Chapter 16: Symmetry of polytopes and polyhedra

REFERENCES

[Bar83] D.W. Barnette. *Map Coloring, Polyhedra, and the Four-Color Problem.* Math. Assoc. America, Washington, 1983.

[Bar91] D.W. Barnette. The minimal projective plane polyhedral maps. In P. Gritzmann and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics—The Victor Klee Festschrift*, pages 63–70, volume 4 of *DIMACS Series in Discrete Math. and Theor. Comput. Sci.* Amer. Math. Soc., Providence, 1991.

[BGH91] D.W. Barnette, P. Gritzmann, and R. Höhne. On valences of polyhedra. *J. Combin. Theory Ser. A*, 58:279–300, 1991.

[Bjö95] A. Björner. Topological methods. In: R.L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, pages 1819–1872. Elsevier Science Publ., Amsterdam, 1995.

[BLS+93] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G.M. Ziegler. *Oriented Matroids.* Volume 46 of *Encyclopedia Math. Appl.*. Cambridge University Press, 1993.

[BS89] J. Bokowski and B. Sturmfels. *Computational Synthetic Geometry.* Volume 1355 of *Lecture Notes in Math.* Springer-Verlag, Heidelberg, 1989.

[BW88] J. Bokowki and J.M. Wills. Regular polyhedra with hidden symmetries. *Math. Intelligencer*, 10:27–32, 1988.

[Bre90] U. Brehm. Polyhedral maps with few edges. In R. Bodendiek and R. Henn, editors, *Topics in Combinatorics and Graph Theory*, pages 153–162. Physica Verlag, Heidelberg, 1990.

[BA86] U. Brehm and A. Altshuler. On weakly neighborly polyhedral maps of arbitrary genus. *Israel J. Math.*, 53:137–157, 1986.

[BS95] U. Brehm and G. Schild. Realizability of the torus and the projective plane in R^4. *Israel J. Math.*, 91:249–251, 1995.

[BW93] U. Brehm and J.M.Wills. Polyhedral manifolds. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, Volume A, pages 535–554. North-Holland, Amsterdam, 1993.

[Con90] M. Conder. Hurwitz groups: A brief survey. *Bull. Amer. Math. Soc.*, 23:359–370, 1990.

[Cox73] H.S.M. Coxeter. *Regular Polytopes* (3rd edition). Dover, New York, 1973.

[CM80] H.S.M. Coxeter and W.O.J. Moser. *Generators and Relations for Discrete Groups* (4th edition). Springer-Verlag, Berlin, 1980.

[Gri83] P. Gritzmann. The toroidal analogue of Eberhard's theorem. *Mathematika*, 30:274–290, 1983.

[Grü67] B. Grünbaum. *Convex Polytopes.* Interscience, London, 1967; revised edition (V. Klee and P. Kleinschmidt, editors), *Graduate Texts in Math.*, Springer-Verlag, in preparation.

[Grü70] B. Grünbaum. Polytopes, graphs, and complexes. *Bull. Amer. Math. Soc.*, 76:1131–1201, 1970.

[GS86] B. Grünbaum and G.C. Shephard. *Tilings and Patterns.* Freeman, New York, 1987.

[Jen93] S. Jendrol. On face-vectors and vertex-vectors of polyhedral maps on orientable 2-manifolds. *Math. Slovaca*, 43:393–416, 1993.

[Küh95] W. Kühnel. *Tight Polyhedral Submanifolds and Tight Triangulations.* Volume 1612 of *Lecture Notes in Math.* Springer-Verlag, New York, 1995.

[Moi77] E.E. Moise. *Geometric Topology in Dimensions 2 and 3.* Volume 47 of *Graduate Texts in Math.*, Springer-Verlag, New York, 1977.

[MS] P. McMullen and E. Schulte. *Abstract Regular Polytopes*. Monograph in preparation.

[Ric96] J. Richter-Gebert. *Realization Spaces of Polytopes.* Volume 1643 of *Lecture Notes in Math.*, Springer-Verlag, Berlin/Heidelberg, 1996.

[Rin74] G. Ringel. *Map Color Theorem.* Springer-Verlag, Berlin, 1974.

[SW91] E. Schulte and J.M. Wills. Combinatorially regular polyhedra in three-space. In K.H. Hofmann and R. Wille, editors, *Symmetry of Discrete Mathematical Structures and Their Symmetry Groups*, pages 49–88. Heldermann Verlag, Berlin, 1991.

[Wil78] S.E. Wilson. Non-orientable regular maps. *Ars Combin.*, 5:213–218, 1978.

[Zie95] G.M. Ziegler. *Lectures on Polytopes.* Volume 152 of *Graduate Texts in Math.*, Springer-Verlag, New York, 1995.

ALGORITHMS AND COMPLEXITY OF FUNDAMENTAL GEOMETRIC OBJECTS

19 CONVEX HULL COMPUTATIONS

Raimund Seidel

INTRODUCTION

The "convex hull problem" is a catch-all phrase for computing various descriptions of a polytope that is either specified as the convex hull of a finite point set in $\mathbb{R}^d$ or as the intersection of a finite number of halfspaces. We first define the various problems and discuss their mutual relationships (Section 19.1). We discuss the very special case of the irredundancy problem in Section 19.2. We consider general dimension d in Section 19.3 and describe the most common general algorithmic approaches along with the best run-time bounds achieved so far. In Section 19.4 we consider separately the case of small dimensions $d = 2, 3, 4, 5$. Finally, Section 19.5 addresses various issues related to the convex hull problem.

19.1 DESCRIBING CONVEX POLYTOPES AND POLYHEDRA

"Computing the convex hull" is a phrase whose meaning varies with the context. Consequently there has been confusion regarding the applicability and efficiency of various "convex hull algorithms." We therefore first discuss the different versions of the "convex hull problem" along with versions of the "halfspace intersection problem" and how they are related via polarity.

CONVEX HULLS

The generic convex hull problem can be stated as follows: Given a finite set $S \subset \mathbb{R}^d$, compute a description of $P = \text{conv}S$, the ***polytope*** formed by the convex hull of S.

A convex polytope P can be described in many ways. In our context the most important descriptions are those listed below.

GLOSSARY

(See Chapter 13 for basic concepts and results of polytope theory.)

Vertex description: The set of all vertices of P (specified by their coordinates).

Facet description: The set of all facets of P (specified by their defining linear inequalities).

Double description: The set of vertices of P, the set of facets of P, and the incidence relation between the vertices and the facets (specified by an incidence matrix).

Lattice description: The face lattice of P (specified by its *Hasse diagram* (cf.

below), with vertex and facet nodes augmented by coordinates and defining linear inequalities, respectively).

Boundary description: A triangulation of the boundary of P (specified by a simplicial complex, with vertices and maximal simplices augmented by coordinates and defining normalized linear inequalities, respectively).

Hasse diagram: A directed graph of an order relation that joins nodes a to b iff $a \leq b$ and there are no elements between a and b in the sense that if $a \leq c \leq b$ then either $c = a$ or $c = b$. For the face lattice the order relation is containment.

The five descriptions above assume that P is full-dimensional. If it is not, then a specification of the smallest affine subspace containing P has to be added to all but the vertex description.

These five descriptions make explicit to varying degrees the geometric information carried by polytope P and the combinatorial information of its facial structure. The vertex description and the facet description each carry only rudimentary geometric information about P. We therefore call them ***purely geometric descriptions***. The other three descriptions we call ***combinatorial*** since they also carry more or less complete combinatorial information about the face structure of P. As a matter of fact, these three descriptions are equivalent in the sense that one can be computed from the other by purely combinatorial means, i.e., without the use of arithmetic operations on real numbers.

Which description is to be computed depends on the application at hand. It is important keep in mind, however, that these descriptions can differ drastically in terms of their sizes (see Section 19.3).

INTERSECTION OF HALFSPACES

Closely related to the convex hull problem is the ***halfspace intersection*** problem: Given a finite set H of halfspaces in $\mathbb{R}^d$, compute a description of the *polyhedron* $Q = \bigcap H$.

Convex polyhedra are more general objects than convex polytopes in that they need not be bounded. Consequently their descriptions are slightly more complicated. Every polyhedron Q admits a unique "factorization" $Q = L + C + R$, where L is a linear subspace orthogonal to C and R, the set C is a convex cone, and R is a convex polytope. The "vertex description" of Q then consists of a minimal set of vectors spanning L, the set of extreme rays of C, and the set of vertices of R. Our other four description methods for convex polytopes have to be adjusted accordingly in order to apply to polyhedra. Also, the triangulations appearing in the boundary description need to allow for unbounded simplices (this concept makes sense if one views a k-simplex as an intersection of $k + 1$ halfspaces).

Because polyhedra are more general than polytopes, all statements about the size differences among the various descriptions of the latter apply also to the former.

POLARITY

The relationship between computing convex hulls and computing the intersection of halfspaces arises because of ***polarity*** (Section 13.1.2). Let S be a finite set in $\mathbb{R}^d$ and let H_S be the set of halfspaces $\{h_p \mid p \in S\}$, with $h_p = \{x \mid \langle x, p \rangle \leq 1\}$. Let

$P = \text{conv}S$ and let $Q = \bigcap H_S$. Polarity yields a 1-1 correspondence between the k-faces of Q and the $(d-k)$-faces of P that admit supporting hyperplanes having P and the origin strictly on the same side. In particular, if the origin is contained in the relative interior of P, then the face lattices of P and Q are anti-isomorphic.

It is thus easy to reduce a convex hull problem to a halfspace intersection problem: First translate S by $-\sum_{p \in S} p/|S|$ to insure that the origin is contained in the relative interior of P, and compute $Q = \bigcap H_S$ for the resulting H_S. The polytope Q is then the polar P^Δ of P, and, assuming that P is full-dimensional, we have straightforward correspondences between the vertex description of Q and the facet description of P, between the facet description of Q and the vertex description of P, between the double descriptions of Q and of P (reverse the roles of vertices and facets), and between the lattice descriptions of Q and P (reverse the order of the lattice). Note that there is *no* correspondence between the boundary descriptions. If P has dimension $l < d$ then $Q = Q' \times L$, where polytope Q' has dimension l and L is a linear subspace of dimension $d - l$. The indicated correspondences then hold between P and Q'.

Reducing a halfspace intersection problem to a convex hull problem is more difficult. Polarity assumes all halfspaces to be describable as $\{x \mid \langle a, x \rangle \leq 1\}$, which means they must strictly contain the origin. In general not all halfspaces in a set H will be of such a form. In order to achieve this form the origin must be translated to a point r that is contained in the interior of $Q = \bigcap H$. Determining such a point r requires solving a linear program. Moreover, such an r does not exist if Q is empty, in which case the halfspace intersection problem has a trivial solution, or if Q is not full-dimensional, in which case one has to perform some sort of dimension reduction.

In general, halfspace intersection appears to be a slightly more general and versatile problem, especially in a homogenized formulation, which very elegantly avoids various special cases (see, e.g., [MRTT53]). Nevertheless, we will concentrate exclusively on the convex hull problem. The stated results can be translated *mutatis mutandis* to the halfspace intersection problem. In many cases the algorithms can be "dualized" to apply directly to the halfspace intersection problem, or the algorithms were originally stated for the halfspace intersection problem and were "dualized" to the convex hull problem.

19.2 THE IRREDUNDANCY PROBLEM

GLOSSARY

Irredundancy problem: Given a set S of n points in $\mathbb{R}^d$, compute the vertex description of $P = \text{conv}S$.

$\lambda(n,d)$: The time to solve a linear programming problem in d variables with n constraints. $O(n)$ for fixed d.

This problem seeks to compute all points in S that are irredundant, in the sense that they cannot be represented as a convex combination of the remaining points in S. The equivalent polar formulation requires computation of the facet

description of $Q = \bigcap H$, given a set H of n halfspaces in $\mathbb{R}^d$. We will follow the primal formulation.

The flavor of this version of the convex hull problem is very different from the other versions. Testing whether a point $p \in S$ is irredundant amounts to solving a linear programming problem in d variables with $n-1$ constraints. The straightforward method of successively testing points for irredundancy results in an algorithm with running time $O(n\lambda(n-1,d))$, which for fixed dimension d is $O(n^2)$.

Clarkson [Cla94] has ingeniously improved this method so that every linear program involves only at most V constraints, where V is the number of vertices of P, i.e., the output size. The resulting running time is $O(n\lambda(V,d))$, which for fixed d is $O(nV)$.

In each of these two methods the n linear programs that occur are closely related to each other. This can be exploited, at least theoretically, by using data structures for so-called linear programming queries [Mat93]. This was first done by Matoušek for the naive method [Mat93], and then by Chan for the improved method [Cha95], resulting for fixed $d > 3$ in an asymptotic time bound of

$$O(n \log^{d+2} V + (nV)^{1-1/(\lfloor d/2 \rfloor + 1)} \log^{O(1)} n) .$$

Finally, note that for the small-dimensional case $d = 2, 3$ there are even algorithms with running time $O(n \log V)$ (see Chapter 38), which can be shown to be asymptotically worst-case optimal [KS86].

19.3 COMPUTING COMBINATORIAL DESCRIPTIONS

GLOSSARY

Facet enumeration problem: Compute the facet description of $P = \text{conv}S$, given S.

Vertex enumeration problem: Compute the vertex description of $Q = \bigcap H$, given H.

The facet and vertex enumeration problems are classical and were already considered as early as 1824 by Fourier (see [Sch86, pp. 209–225] for a survey). Interestingly, no *efficient* algorithm is known that solves these enumeration problems without also computing, besides the desired purely geometric description, some combinatorial description of the polyhedron involved. Consequently we now concentrate on computing combinatorial descriptions.

THE SIZES OF COMBINATORIAL DESCRIPTIONS

It is important to understand how the three combinatorial descriptions differ in terms of their sizes. Let S be a set of n points in $\mathbb{R}^d$ and let $P = \text{conv}S$. Assume that P is a d-polytope and that it has m facets. As a consequence of McMullen's Upper Bound Theorem (Chapter 15) and of polarity, the following inequalities hold

between n and m and are tight:

$$n \leq \mu(d,m) \qquad \text{and} \qquad m \leq \mu(d,n)\,,$$

where

$$\mu(d,x) = f_{d-1}(C_d(x)) = \binom{x - \lceil d/2 \rceil}{\lfloor d/2 \rfloor} + \binom{x - 1 - \lceil (d-1)/2 \rceil}{\lfloor (d-1)/2 \rfloor}\,,$$

which is $\Theta(x^{\lfloor d/2 \rfloor})$ for fixed d.

For the sake of definiteness let us define the sizes of the various descriptions as follows. For the double description of P it is the number of vertex-facet incidences, for the lattice description it is the total number of faces (of all dimensions) of P, and for the boundary description it is the number of $(d-1)$-simplices in the boundary triangulation.

Note that for the double and the lattice descriptions the sizes are completely determined by P, whereas the size of a boundary description depends on the boundary triangulation that is actually used. The sizes of those triangulations for a given P can vary quite drastically, even if, as we assume from now on, all vertices of the triangulation must be from S.

These size measures are only crude approximations of the space required to store such descriptions in memory (in particular, in case of the lattice description the edges of the Hasse diagram are completely ignored). However, these approximations suffice to convey the possible similarities and differences between the sizes of the different descriptions.

For such a comparison between the description sizes of $P = \text{conv}S$ consider Table 19.3.1, whose columns deal with three cases. The first column lists worst-case upper bounds in terms of n and d. The second columns lists upper bounds in terms of m and d under the assumption that S is in nondegenerate position, i.e., no $d+1$ points in S lie in a common hyperplane, which means that P must be simplicial. Note that in this case there is a unique boundary description. Finally, the third column lists asymptotic bounds (d fixed) for *products of cyclic polytopes* $CC_d(n)$, a certain class of highly degenerate polytopes described in [ABS97]. (See Section 13.1.4 for a discussion of cyclic polytopes.) In this third table column, $\delta = \lfloor \sqrt{d/2} \rfloor$.

TABLE 19.3.1 Polytope description sizes.

DESCRIPTION	WORST CASE	NONDEGENERATE	DEGENERATE CLASS $CC_d(n)$
Double	$d \cdot \mu(d,n)$	$d \cdot m$	$\Theta(n \cdot m^{1-1/\delta})$
Lattice	$2^d \cdot \mu(d,n)$	$2^d \cdot m$	$\Theta((n+m)^\delta)$
Boundary	$\mu(d,n)$	m	$\Omega((n+m)^\delta)$

The bounds in the table are based on the fact that all description sizes are maximized when P is a cyclic polytope, that each facet of a simplicial d-polytope contains 2^d faces, and that the Upper Bound Theorem also applies to simplicial spheres. The lower bound on the size of the boundary description of $CC_d(n)$ applies no matter which triangulation of the boundary is actually used.

The implication of this table is that in the worst case and also in the nondegenerate case all three combinatorial descriptions of P have approximately the same size. If d is considered constant, then the sizes are $\Theta(n^{\lfloor d/2 \rfloor})$ in the worst case, where n is the number of points in S (i.e., n is the input size), and the description sizes are $\Theta(m)$ in the nondegenerate case, where m is the number of facets of P (in a way the output size). The third column of the table, however, shows that in the general case the double description of a polytope P may be substantially more compact than the lattice description or the boundary description.

MAIN RESULTS AND OPEN PROBLEMS

The main positive results are that in the sense of asymptotic worst case complexity the convex hull problem has been solved completely, and that in the case of nondegenerate input, each of the three combinatorial descriptions can be found in time polynomial in the size of the input and the size of the output. In the case of general input this has only been shown for the lattice and for a boundary description, whereas it is unknown whether this is also possible for the double description.

In the following let $P = \text{conv}S$ be a d-polytope, and $|S| = n$.

THEOREM 19.3.1 *Chazelle* [Cha93]

If the dimension d is considered constant, then given S, each of the three combinatorial descriptions of $P = \text{conv}S$ can be computed in time $O(n \log n + n^{\lfloor d/2 \rfloor})$ using space $O(n^{\lfloor d/2 \rfloor})$. This is asymptotically worst-case optimal.

THEOREM 19.3.2 *Avis-Fukuda* [AF92]

Given S, a boundary description of $P = \text{conv}S$ can be computed in time $O(dnM)$ using space $O(dn)$, where M is the size of the boundary description produced.

If S is nondegenerate, then each of the three combinatorial descriptions of P can be computed in time $O(d^{O(1)}nM)$, where M is the size of the respective description.

THEOREM 19.3.3 *Swart* [Swa85] *and Chand-Kapur* [CK70]

Given S, the lattice description of $P = \text{conv}S$ can be computed in time and space polynomial in d, n, and the size of the output.

OPEN PROBLEM 19.3.4

Is there an algorithm that, given S, computes the double description of $P = \text{conv}S$ in time polynomial in d, n, and the size of the double description?

The algorithm in Chazelle's theorem appears to be of theoretical interest only. The algorithm of Avis-Fukuda is quite practical, the algorithms of Swart and of Chand and Kapur are less so because of the potentially large space requirements. (See Chapter 52 for descriptions of available code.) The running times of the last two algorithms admit some theoretical improvements, as will be discussed in the following sections.

All algorithms that have been published for solving the different versions of the convex hull problem and the halfspace intersection problem appear to be variations of three general methods: incremental, graph traversal, and divide-and-conquer. We discuss the incremental and the graph traversal methods in the next two subsections. Divide-and-conquer has proven useful only for very small dimension, and

we will discuss it in that context in Section 19.4.

19.3.1 THE INCREMENTAL METHOD

The incremental method puts the points in S in some order $p_1, \ldots, p_n$ and then successively computes a description of $P_i = \text{conv}S_i$ from the description of P_{i-1} and p_i, where $S_i = \{p_1, \ldots, p_i\}$.

Before discussing details it should be noted that no matter how the incremental method is implemented, it has a serious shortcoming in that the intermediate polytopes P_i may have many more facets than the final $P_n = P$ (see, e.g., [ABS97]). Thus the description sizes of the intermediate polytopes may be much larger than the size of the description of the final result, and hence this method cannot have running time that depends reasonably on the output size.

This is not necessarily just the result of an unfortunate choice of the insertion order. Recently, Bremner [Bre96] has shown that if S is the vertex set of the aforementioned product of cyclic polytopes $CC_d(n)$, then P_{n-1} has $\Omega(m^{\lfloor\sqrt{d/2}\rfloor-1})$ facets no matter which insertion order is used, where m is the number of facets of $P_n = P$.

We first present a selection of algorithms implementing the incremental method and list their asymptotic worst-case or expected running times for fixed d (Table 19.3.2). All these algorithms compute boundary descriptions, except for [Sei81] (see also [Ede87, Section 8.4]), which can also be made to compute a lattice description, and [MRTT53], which computes a double description.

TABLE 19.3.2 Sample of incremental algorithms.

ALGORITHM	TIME	BOUND TYPE
Kallay [PS85, Section 3.4.2]	$n^{\lfloor d/2\rfloor+1}$	worst-case
Seidel [Sei81]	$n \log n + n^{\lceil d/2\rceil}$	worst-case
Chazelle [Cha93]	$n \log n + n^{\lfloor d/2\rfloor}$	worst-case
Clarkson-Shor [CS89]	$n \log n + n^{\lfloor d/2\rfloor}$	expected
Clarkson et al. [CMS93]	$n \log n + n^{\lfloor d/2\rfloor}$	expected
Motzkin et al. [MRTT53]	$n^{3\lfloor d/2\rfloor+1}$	worst-case

We now concentrate on how P_{i-1} and P_i differ. For the sake of simplicity we will first assume that S is nondegenerate and hence all involved polytopes are simplicial. Moreover we will ignore how the insertion method starts and assume that P_{i-1} and P_i are full-dimensional. We say that a facet of P_{i-1} is ***visible*** (from p_i) if its supporting hyperplane separates P_{i-1} and p_i. Otherwise the facet is ***obscured***.

The facet set of P_i consists of "old facets," namely all obscured facets of P_{i-1}, and "new facets," namely facets of the form $\text{conv}(R \cup \{p_i\})$, where R is a "horizon" ridge of P_{i-1}, i.e., R is contained in a visible and in an obscured facet of P_{i-1}.

Updating P_{i-1} to P_i thus requires solving three subproblems: finding (and deleting) all visible facets of P_{i-1}; finding all horizon ridges; forming all new facets. The various incremental algorithms only differ in how they solve those subproblems, and they differ in the type of insertion order used.

Visible facets. The simplest way of finding the visible facets is simply to check each facet of P_{i-1}. This is done in Kallay's "beneath-beyond" method [PS85, Section 3.4.2] and in the "double description method" of Motzkin et al. [MRTT53]. Since P_i may have $\Theta(i^{\lfloor d/2 \rfloor})$ facets such an approach automatically leads to a suboptimal overall running time of $\Omega(n^{\lfloor d/2 \rfloor + 1})$ in the worst case.

Another way is to maintain "conflict lists" between facets and not yet inserted points. In the worst case this is no better than the previous method. However, if the insertion order is a random permutation of the points in S, then in expectation this method works in $O(n^{\lfloor d/2 \rfloor})$ time [CS89].

The last method requires the maintenance of a ***facet graph***, whose nodes are the facets and whose arcs connect facets if they share a common ridge. The visible facets form a connected subgraph of this facet graph. Thus they can be determined by graph search, such as depth-first search. This takes time proportional to the number of visible facets, which means that in the amortized sense this takes no time since all those visible facets will be deleted. This graph search requires that one starting visible facet be known. Such an initial visible facet can be determined relatively efficiently by a special choice of the insertion order, as in [Sei81], by maintaining "canonical visible facets," as in [CS89] and [CMS93], or by linear programming, as in [Sei91].

Horizon ridges. Determining the horizon ridges is trivial if the facet graph is used, since those ridges correspond to arcs connecting visible and obscured facets. Otherwise one has to use data structuring techniques to determine which of the ridges incident to the visible facets are incident to exactly one visible facet.

New facets. After the horizon ridges are determined, the new facets are easily constructed in time proportional to their number. Keeping this number small is one of the main difficulties of making the insertion method efficient. In the worst case there may be as many as $\mu(d-1, i-1) = \Theta(i^{\lfloor (d-1)/2 \rfloor})$ such new facets. For even d this is $\Theta(i^{\lfloor d/2 \rfloor - 1})$, which is the main reason why it was relatively easy to obtain an asymptotically worst-case optimal running time of $O(n^{\lfloor d/2 \rfloor})$ for even d [Sei81]. For general d, using a random insertion order [CS89, CMS93, Sei91] appears to be the only known way to keep this number low, at least in terms of expectation. Chazelle's celebrated deterministic algorithm [Cha93] applies derandomization and thus in effect "simulates" random insertion order so that the number of new facets is not only small in the expected sense but also in the worst case.

Finally, if a facet graph is used, then the arcs corresponding to the ridges between the new facets need to be generated, which can be done via data structuring techniques, as in [Sei91], or by graph traversal techniques, as in [CS89, CMS93]. We should mention that if we remove the nondegeneracy assumption this problem of determining the new ridges seems to become very difficult.

Degenerate input. So far we have assumed that the input set S be nondegenerate. If this is not the case, then this can be simulated using perturbation techniques [Sei94]. This way the algorithms produce a boundary description from which a lattice description or a double description could be computed in $O(n^{\lfloor d/2 \rfloor})$ worst-case time.

The algorithm of Seidel [Sei81] (see also [Ede87, Section 8.4]) also works with degenerate input and then produces a lattice description. Most interesting, though, in the case of degeneracy is the so-called double description algorithm of Motzkin et al. [MRTT53].

THE DOUBLE DESCRIPTION METHOD

Although it is one of the oldest published incremental algorithms, this method has received little attention in the computational geometry community. This method maintains only the double descriptions of the polytopes P_i. It makes no assumptions about nondegeneracy. In fact, despite its poor worst-case complexity, empirically this method works well for degenerate inputs, where all other methods seem to fail, running out of time or space.

The algorithm determines the visible facets by simply checking all facets of P_{i-1}. The interesting point is how it determines the horizon ridges, from which the new facets are then constructed. In contrast to the other methods it does not maintain ridges, since, as we already mentioned, determining the new ridges created during an insertion is difficult. The double description method simply considers each pair of visible and obscured facets of P_{i-1} and checks whether their intersection A forms a horizon ridge. This is achieved by testing whether the vertex set in A is contained in some other facet of P_{i-1}. If it is, then A is not a ridge and hence not a horizon ridge.

A straightforward implementation of this idea will require $\Theta(i^{3\lfloor d/2 \rfloor})$ time in the worst case to discover all horizon ridges of P_{i-1}, resulting in a high worst-case overall running time. Although a number of heuristics have been proposed to speed up this process (see [Zie94, p. 48]), experiments show that this method is unbearably slow in the nondegenerate case when compared to other algorithms. However, in the case of degenerate input it still appears to be the method of choice.

Finally, we should mention that convex hull algorithms based on so-called Fourier-Motzkin elimination are nothing but incremental algorithms dressed up in an algebraic formulation.

19.3.2 THE GRAPH TRAVERSAL METHOD

This method attempts to traverse the facet graph of polytope $P = \text{conv}S$ in an organized fashion. The basic step is: given a facet F of P and a ridge R contained in F, find the other facet F' of P that also contains R. Geometrically this amounts to determining the point $p \in S$ such that the hyperplane spanned by R and p maximizes the angle to F. In analogy to a 3-dimensional physical realization this operation is therefore known as a "gift-wrapping step," and these algorithms are known as ***gift-wrapping algorithms***. In the polar context of intersecting half-spaces, this step corresponds to moving along an edge from one vertex to another and is equivalent to a pivoting step of the simplex algorithm for linear programming. Thus these algorithms are also known as ***pivoting algorithms***.

The basic outline of the graph traversal method is as follows: Find some initial facet of $P = \text{conv}S$ and the ridges that it contains. As long as there is an ***open ridge*** R, i.e., one for which only one containing facet F is known, perform a gift-wrapping step to discover the other facet F' containing R and determine the ridges that F' contains.

This general method faces three problems:

(a) How does one maintain the set of open ridges?

(b) How can the ridges of the new facet F' be quickly discovered?

(c) How can an individual gift-wrapping step be performed quickly?

THE NONDEGENERATE CASE

Let us again first assume that the input set S is in nondegenerate position. This trivializes problem (b) since every facet is a $(d-1)$-simplex and each of the d subsets with $d-1$ of its d vertices will span a ridge.

The most straightforward way to deal with problem (a) is to use some sort of dictionary data structure to store the set of open ridges. The most straightforward way to deal with (c) is to scan through all the points in S to find the best candidate, leading to work proportional to n per discovered facet. This straightforward method has been proposed many times (see [Sch86, p. 224] and [Chv83, p. 282] for references) and has running time $O(d^2nM)$ using $O(d(M+n))$ space, where M is the number of facets of P.

The gift-wrapping steps can be performed faster if a special data structure (for the dual of ray-shooting queries) is used. This was developed by Chan [Cha95], who achieved for fixed $d > 3$ an asymptotic time bound of

$$O(n \log M + (nM)^{1-1/(\lfloor d/2 \rfloor + 1)} \log^{O(1)} n) .$$

Avis and Fukuda [AF92] proposed an ingenious way to deal with problem (a) so that no storage space is needed. They pointed out that there is a way of defining a canonical spanning tree T of the facet graph of polytope P so that the arcs of T can be recognized locally. Gift-wrapping steps are then performed only over ridges corresponding to arcs of T. Doing this in the form of a depth-first search traversal of T avoids the use of any extra storage space. Facets can be output as soon as they are discovered. Their algorithm is eminently practical and has a running time of $O(dnM)$ using only $O(dn)$ space.

In theory the gift-wrapping step improvement of Chan also could be applied to the algorithm of Avis and Fukuda. However, this appears to be of little practical relevance.

A completely different way of simultaneously addressing problems (a) and (c) was suggested by Seidel [Sei86a]. He proposed to try to discover the facets in an order corresponding to a straight-line shelling of P. In many cases gift-wrapping steps over several currently open ridges would yield the same new facet F'. However, in that case the entire vertex set of F' is known already and the expensive scan to solve problem (c) is not necessary. The facets of P for which this trick is not applicable can be discovered in advance by linear programming. This "shelling algorithm" has running time $O(n\lambda(n-1, d-1) + d^3M \log n)$, where $\lambda(n-1, d-1)$ is again the time necessary to solve a linear program with $n-1$ constraints in $d-1$ variables. From the way a shelling proceeds one can prove that the space requirement for storing the open ridges is somewhat lower than in an ordinary gift-wrapping algorithm.

The linear programs that need to be solved are similar to the ones in the irredundancy problem of Section 19.2. Again improvements can be achieved by applying linear programming queries. Thus, as Matoušek [Mat93] has shown, the $n\lambda(n-1, d-1)$ factor can be improved to $n^{2-2/(\lfloor d/2 \rfloor + 1)} \log^{O(1)} n$.

THE GENERAL CASE

There are two ways to approach the general case where P is not simplicial. The first is again to apply perturbations in order to simulate nondegeneracy of S. This way all previously mentioned algorithms still apply, however they now compute a boundary description of P. The parameter M is now the size of the triangulation

that happens to be constructed. Moreover, the perturbed computations slow down the running times by a polynomial factor in d.

The second way to deal with the general case is to generalize the algorithms so that they compute the lattice description of P. The main obstacle that must be overcome in the degenerate case is problem (b), the discovery of the ridges of a new facet F'. The obvious way to address this problem is to view the construction of F' as a recursive subproblem one dimension down. Some care must be taken however that in the many recursions small-dimensional faces are not reconstructed too often. This method was proposed by Chand and Kapur [CK70] and their algorithm was later improved and analyzed by Swart [Swa85] who showed a running time of $O(d^2nK_1 + d^3K_2 \log K_0)$, where K_i is the number of directed $(i+1)$-vertex paths in the Hasse diagram of the face lattice of P.

Rote [Rot92] generalized the algorithm of Avis and Fukuda to produce the lattice description using little storage space. Its running time is $O(dK_{d+1}n)$ and it appears to be not as relevant in practice as the original algorithm.

Finally, Seidel [Sei86b] generalized his shelling algorithm to produce the lattice description in time $O(n\lambda(n-1, d-1) + K_2(d^2 + \log K_0))$. Because of the recursive nature of straight-line shellings, this generalization avoids reconstruction of small-dimensional faces. Again the improvement of Matoušek applies.

19.4 THE CASE OF SMALL DIMENSION

Convex hull computations in very small dimension are special. We have strong geometric intuitions about 2- and 3-dimensional space (and via Schlegel diagrams even about 4-polytopes). Moreover the situation is simpler in the case $d = 2, 3$ since our five polytope descriptions cannot differ much in terms of their sizes (they are all within a constant factor of each other), which means there is little need for keeping an exact distinction. Algorithmically, small dimensions are special in that besides the incremental and the graph traversal method, divide-and-conquer methods have also been brought to fruition.

THE 2-DIMENSIONAL CASE

The planar convex hull problem has drawn considerable attention and many different algorithmic paradigms have been tried (see textbooks such as [PS85] or [O'R94]). The graph traversal method was rediscovered and is known in the planar case as the ***Jarvis march*** with running time $O(nM)$, and the incremental method was rediscovered and is known in a rather different guise as the ***Graham scan*** with running time $O(n \log n)$ (as usual n and M are the sizes of the input and output, respectively). It was easy and natural to apply the divide-and-conquer paradigm to obtain further $O(n \log n)$ time algorithms. By giving this paradigm the extra twist of "marriage-before-conquest" it was possible even to obtain an $O(n \log M)$ algorithm, which was also shown to be worst-case optimal in the algebraic computation tree model of computation [KS86]. This algorithm required the use of 2-dimensional linear programming. Much later Chan, Snoeyink, and Yap [CSY95] showed how to avoid this and substantially simplified the algorithm in way that allowed its generalization to higher dimensions. Recently Chan [Cha95] showed

quite surprisingly that by using simple data structures and the method of guessing the output size by repeated squaring, the Jarvis march algorithm can be sped up to also run in time $O(n \log M)$.

THE 3-DIMENSIONAL CASE

In 3 dimensions the output size M is $O(n)$ in the worst case. However, the straightforward implementations of the standard incremental and the graph traversal methods only yield algorithms with worst-case running time $O(n^2)$. In this context the use of the divide-and-conquer paradigm was decisive in obtaining $O(n \log n)$ running time, which was achieved by Preparata and Hong (see [PS85, Section 3.4.4]; for a more detailed account, [Ede87, Section 8.5]). This running time was later matched in the expected sense by the randomized incremental algorithm of Clarkson and Shor [CS89], who also gave another randomized algorithm with expected performance $O(n \log M)$.

The question whether this optimal output-size sensitive bound could also be achieved deterministically was open for a long time. Edelsbrunner and Shi [ES91] first generalized the "marriage-before-conquest" method of [KS86] but achieved only a running time of $O(n \log^2 M)$. Eventually Chazelle and Matoušek [CM92] succeeded in derandomizing the randomized algorithm of Clarkson and Shor and obtained, at least theoretically, this optimal $O(n \log M)$ time bound. Recently Chan [Cha95] showed that there is a relatively simple algorithm for achieving this bound, again by the method of speeding up the gift-wrapping method using data structures and guessing the output size by repeated squaring.

THE CASE $d = 4,5$

In this case the sizes of the combinatorial descriptions may be as large as $\Theta(n^2)$. All the methods and bounds mentioned in Section 19.3 apply. In addition there are recent methods for computing a boundary description based on sophisticated divide-and-conquer and some additional pruning mechanisms. Worst-case time bounds of $O((n+M) \log^{d-2} M)$ were achieved by Chan, Snoeyink, and Yap [CSY95] for $d = 4$, and by Amato and Ramos [AR96] for $d = 4, 5$. The latter paper also states that their bound applies to computing the lattice description in the case $d = 4$.

19.5 RELATED TOPICS

The expected sizes of convex hulls of point sets drawn according to some statistical distribution are typically much smaller than the worst-case sizes. Constructing such convex hulls has been explicitly studied by several authors (see, e.g.,[DT81, Dwy91, BGJR91]). One should also mention in this context the randomized incremental algorithm [CS89]. With input set $S \subset \mathbb{R}^d$ its expected running time for constructing a boundary description is

$$O\left(\sum_{d+1<r\leq n} df_r(S)/r + \sum_{d+1\leq r<n} d^2 n f_r(S)/r^2\right),$$

where $f_r(S)$ is the expected size of the boundary description of the convex hull of a random subset of S of size r. For many distributions f_r is sufficiently sublinear so that this randomized incremental algorithm has $O(n)$ expected running time.

The problem of maintaining convex hulls under insertions and deletions of points has been addressed also. A truly satisfactory solution has only been obtained in the planar case [OvL81, Gow80] based on divide-and-conquer algorithms. In higher dimensions randomized incremental algorithms have been adapted by several authors to process updates [Mul94, Sch91, CMS93]. However, the analyses are all based on some probabilistic model of which updates actually occur.

For some time there was hope that additional input information might help compute convex hulls. Although this is true in the planar case, where having points presorted or having them given along a nonintersecting polygonal line [Mel87] leads to linear-time algorithms, it has been shown [Sei85] that for dimension $d \geq 3$ such additional information does not help. Having a 3-dimensional set S presorted or even knowing a non-self-intersecting polyhedral surface whose vertex set is S does not in general make it easier to find the convex hull of S.

There have been some attempts to generalize the convex hull construction problem so that the input S does not consist of points but of more general objects such as algebraically described regions in the plane [NY94] or balls in $\mathbb{R}^d$ [BCD+92].

Finally, parallel algorithms for the convex hull problem have been developed; see Chapter 36.

19.6 SOURCES AND RELATED MATERIALS

FURTHER READING

[Zie94]: A modern account of polytope theory.

[MR80]: A survey of vertex enumeration methods from the dual standpoint.

RELATED CHAPTERS

Chapter 13: Basic properties of convex polytopes.
Chapter 15: Face numbers of polytopes and complexes
Chapter 20: Voronoi diagrams and Delaunay triangulations.
Chapter 38: Linear programming in low dimensions

REFERENCES

[ABS97] D. Avis, D. Bremner, and R. Seidel. How good are convex hull algorithms? *Comput. Geom. Theory Appl.*, 7, 1997, to appear.

[AF92] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete Comput. Geom.*, 8:295–313, 1992.

[AR96] N.M. Amato and E.A. Ramos. On computing Voronoi diagrams by divide-prune-and-conquer. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 166–175, 1996.

[BCD+92] J.D. Boissonnat, A. Cérézo, O. Devillers, J. Duquesne, and M. Yvinec. An algorithm for constructing the convex hull of a set of spheres in dimension d. In *Proc. 4th Canad. Conf. Comput. Geom.*, pages 269–273, 1992.

[BGJR91] K.H. Borgwardt, N. Gaffke, M. Junger, and G. Reinelt. Computing the convex hull in the Euclidean plane in linear expected time. In P. Gritzmann and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, volume 4 of *DIMACS Series in Discrete Math. and Theoret. Comput. Sci.*, pages 91–107. AMS Press, Providence, 1991.

[Bre96] D. Bremner. Incremental convex hull algorithms are not output sensitive. In *Proc. Internat. Sympos. Algorithms Comput.*, volume 1178 of *Lecture Notes in Comput. Sci.*, pages 26–35. Springer-Verlag, Berlin, 1996. To appear in *Discrete Comput. Geom.*

[Cha93] B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete Comput. Geom.*, 10:377–409, 1993.

[Cha95] T.M. Chan. Output-sensitive results on convex hulls, extreme points, and related problems. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 10–19, 1995.

[Chv83] V. Chvátal. *Linear Programming.* W. H. Freeman, New York, 1983.

[CK70] D.R. Chand and S.S. Kapur. An algorithm for convex polytopes. *J. Assoc. Comput. Mach.*, 17:78–86, 1970.

[Cla94] K.L. Clarkson. More output-sensitive geometric algorithms. In *Proc. 35th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 695–702, 1994.

[CM92] B. Chazelle and J. Matoušek. Derandomizing an output-sensitive convex hull algorithm in three dimensions. Tech. Rep., Dept. Comput. Sci., Princeton Univ., 1992.

[CMS93] K.L. Clarkson, K. Mehlhorn, and R. Seidel. Four results on randomized incremental constructions. *Comput. Geom. Theory Appl.*, 3:185–212, 1993.

[CS89] K.L. Clarkson and P.W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.

[CSY95] T.M. Chan, J. Snoeyink, and C.K. Yap. Output-sensitive construction of polytopes in four dimensions and clipped Voronoi diagrams in three. In *Proc. 6th ACM-SIAM Sympos. Discrete Algorithms (SODA '95)*, pages 282–291, 1995.

[DT81] L.P. Devroye and G.T. Toussaint. A note on linear expected time algorithms for finding convex hulls. *Computing*, 26:361–366, 1981.

[Dwy91] R. Dwyer. Convex hulls of samples from spherically symmetric distributions. *Discrete Appl. Math.*, 31:113–132, 1991.

[Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monogr. Theoret. Comput. Sci.* Springer-Verlag, Heidelberg, 1987.

[ES91] H. Edelsbrunner and W. Shi. An $O(n \log^2 h)$ time algorithm for the three-dimensional convex hull problem. *SIAM J. Comput.*, 20:259–277, 1991.

[Gow80] I.G. Gowda. Dynamic problems in computational geometry. M.Sc. thesis, Dept. Comput. Sci., Univ. British Columbia, Vancouver, 1980.

[KS86] D.G. Kirkpatrick and R. Seidel. The ultimate planar convex hull algorithm? *SIAM J. Comput.*, 15:287–299, 1986.

[Mat93] J. Matoušek. Linear optimization queries. *J. Algorithms*, 14:432–448, 1993.

[Mel87] A. Melkman. On-line construction of the convex hull of a simple polyline. *Inform. Process. Lett.*, 25:11–12, 1987.

[MR80] T.H. Mattheiss and D. Rubin. A survey and comparison of methods for finding all vertices of convex polyhedral sets. *Math. Oper. Res.*, 5:167–185, 1980.

[MRTT53] T.S. Motzkin, H. Raiffa, G.L. Thompson, and R.M. Thrall. The double description method. In H.W. Kuhn and A.W. Tucker, editors, *Contributions to the Theory of Games II*, volume 8 of *Ann. of Math. Stud.*, pages 51–73. Princeton University Press, 1953.

[Mul94] K. Mulmuley. *Computational Geometry: An Introduction through Randomized Algorithms.* Prentice Hall, Englewood Cliffs, 1994.

[NY94] F. Nielsen and M. Yvinec. Output-sensitive convex hull algorithms of planar convex objects. Research Report 2575, INRIA, France, 1994.

[O'R94] J. O'Rourke. *Computational Geometry in C.* Cambridge University Press, 1994.

[OvL81] M.H. Overmars and J. van Leeuwen. Maintenance of configurations in the plane. *J. Comput. Syst. Sci.*, 23:166–204, 1981.

[PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, 1985.

[Rot92] G. Rote. Degenerate convex hulls in high dimensions without extra storage. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 26–32, 1992.

[Sch86] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley-Interscience, Chichester, New York, 1986.

[Sch91] O. Schwarzkopf. Dynamic maintenance of geometric structures made easy. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 197–206, 1991.

[Sei81] R. Seidel. A convex hull algorithm optimal for point sets in even dimensions. M.Sc. thesis, Dept. Comput. Sci., Univ. British Columbia, Vancouver, 1981. Report 81/14.

[Sei85] R. Seidel. A method for proving lower bounds for certain geometric problems. In G.T. Toussaint, editor, *Computational Geometry*, pages 319–334. North-Holland, Amsterdam, 1985.

[Sei86a] R. Seidel. Constructing higher-dimensional convex hulls at logarithmic cost per face. In *Proc. 18th Annu. ACM Sympos. Theory Comput.*, pages 404–413, 1986.

[Sei86b] R. Seidel. *Output-size sensitive algorithms for constructive problems in computational geometry.* Ph.D. thesis, Dept. Comput. Sci., Cornell Univ., Ithaca, 1986. Tech. Rep. TR 86-784.

[Sei91] R. Seidel. Small-dimensional linear programming and convex hulls made easy. *Discrete Comput. Geom.*, 6:423–434, 1991.

[Sei94] R. Seidel. The meaning and nature of perturbations in geometric computing. In *Proc. 11th Sympos. Theoret. Aspects Comput. Sci. (STACS '94)*, pages 1–15, 1994, to appear in *Discrete Comput. Geom.*

[Swa85] G.F. Swart. Finding the convex hull facet by facet. *J. Algorithms*, 6:17–48, 1985.

[Zie94] G.M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Math.* Springer-Verlag, New York, 1994.

20 VORONOI DIAGRAMS AND DELAUNAY TRIANGULATIONS

Steven Fortune

INTRODUCTION

The Voronoi diagram of a set of sites partitions space into regions, one per site; the region for a site s consists of all points closer to s than to any other site. The dual of the Voronoi diagram, the Delaunay triangulation, is the unique triangulation such that the circumsphere of every simplex contains no sites in its interior. Voronoi diagrams and Delaunay triangulations have been rediscovered or applied in many areas of mathematics and the natural sciences; they are central topics in computational geometry, with hundreds of papers discussing algorithms and extensions.

Section 20.1 discusses the definition and basic properties in the usual case of point sites in $\mathbb{R}^d$ with the Euclidean metric, while Section 20.2 gives basic algorithms. Some of the many extensions obtained by varying metric, sites, environment, and constraints are discussed in Section 20.3. Section 20.4 finishes with some interesting and nonobvious structural properties of Voronoi diagrams and Delaunay triangulations.

GLOSSARY

Site: A defining object for a Voronoi diagram or Delaunay triangulation. Also generator, source, Voronoi point.

Voronoi face: The set of points for which a single site is closest (or more generally a set of sites is closest). Also Voronoi region, Voronoi cell.

Voronoi diagram: The set of all Voronoi faces. Also Thiessen diagram, Wigner-Seitz diagram, Blum transform, Dirichlet tessellation.

Delaunay triangulation: The unique triangulation of a set of sites such that the circumsphere of each full-dimensional simplex has no sites in its interior.

20.1 POINT SITES IN THE EUCLIDEAN METRIC

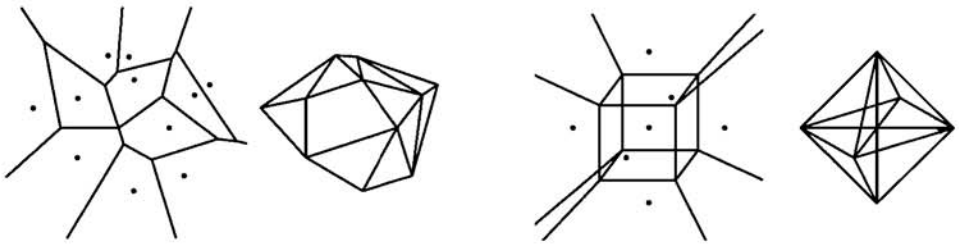
See [Aur91, Ede87, For95] for more details and proofs of material in this section.

GLOSSARY

Sites: Points in a finite set S in $\mathbb{R}^d$.

Voronoi face of a site s: The set of all points of $\mathbb{R}^d$ strictly closer to the

FIGURE 20.1.1
Voronoi diagram and Delaunay triangulation of the same set of sites in two dimensions (a,b) and three dimensions (c,d).



site $s \in S$ than to any other site in S. The Voronoi face of a site is always a nonempty, open, convex, full-dimensional subset of $\mathbb{R}^d$.

Voronoi face V(T) of a subset T: For T a nonempty subset of S, the set of points of $\mathbb{R}^d$ equidistant from all members of T and closer to any member of T than to any member of $S \backslash T$.

Voronoi diagram of S: The collection of all nonempty Voronoi faces $V(T)$, for $T \subseteq S$. The Voronoi diagram forms a cell complex partitioning $\mathbb{R}^d$. In two dimensions (Figure 20.1.1(a)), the Voronoi face of a site is the interior of a convex, possibly infinite polygon; its boundary consists of ***Voronoi edges*** (1-dimensional faces) equidistant from two sites and ***Voronoi vertices*** (0-dimensional faces) equidistant from at least three sites. Figure 20.1.1(c) shows a Voronoi diagram in three dimensions.

Delaunay face D(T) of a subset T: The Delaunay face $D(T)$ is defined for a subset T of S whenever there is a sphere through all the sites of T with all other sites exterior (equivalently, whenever $V(T)$ is not empty). Then $D(T)$ is the (relative) interior of the convex hull of T. For example, in two dimensions (Figure 20.1.1(b)), a ***Delaunay triangle*** is formed by three sites whose circumcircle is empty and a ***Delaunay edge*** connects two sites that have an empty circumcircle (in fact, infinitely many empty circumcircles).

Delaunay triangulation of S: The collection of all Delaunay faces. The Delaunay triangulation forms a cell complex partitioning the convex hull of S.

There is an obvious one-one correspondence between the Voronoi diagram and the Delaunay triangulation; it maps the Voronoi face $V(T)$ to the Delaunay face $D(T)$. This correspondence has the property that the sum of the dimensions of $V(T)$ and $D(T)$ is always d. Thus, in two dimensions, $V(T)$ is a Voronoi vertex iff $D(T)$ is an open polygonal region; $V(T)$ is an edge iff $D(T)$ is; $V(T)$ is an open polygonal region iff $D(T)$ is a vertex, i.e., a site. In fact, the 1–1 correspondence is a duality between cell complexes, reversing face ordering: for subsets $T, T' \subseteq S$, $V(T')$ is a face of $V(T)$ iff $D(T)$ is a face of $D(T')$.

The set of sites $S \subset \mathbb{R}^d$ is in ***general position*** (or is ***nondegenerate***) if no $d+2$ points lie on a common d-sphere and no $k+2$ points lie on a common k-flat, for $k < d$. If S is in general position, then the Delaunay triangulation of S is a simplicial complex, and every vertex of the Voronoi diagram is incident to $d+1$ edges in the Delaunay triangulation. If S is not in general position, then Delaunay faces need

not be simplices; for example the four cocircular sites in Figure 20.1.1(b) form a Delaunay quadrilateral. A ***completion*** of a Delaunay triangulation is obtained by splitting nonsimplicial faces into simplices without adding new vertices.

RELATION TO CONVEXITY

There is an intimate connection between Delaunay triangulations in $\mathbb{R}^d$ and convex hulls in $\mathbb{R}^{d+1}$, and between Voronoi diagrams in $\mathbb{R}^d$ and halfspace intersections in $\mathbb{R}^{d+1}$. To see the connections, consider the special case of $d = 2$. Identify $\mathbb{R}^2$ with the plane spanned by the first two coordinate axes of $\mathbb{R}^3$, and call the third coordinate direction the *vertical* direction.

The ***lifting map*** $\lambda : \mathbb{R}^2 \to \mathbb{R}^3$ is defined by $\lambda(x_1, x_2) = (x_1, x_2, x_1^2 + x_2^2)$; $\Lambda = \lambda(\mathbb{R}^2)$ is a paraboloid of revolution about the vertical axis. See Figure 20.1.2(a). Let H be the convex hull of the lifted sites $\lambda(S)$.

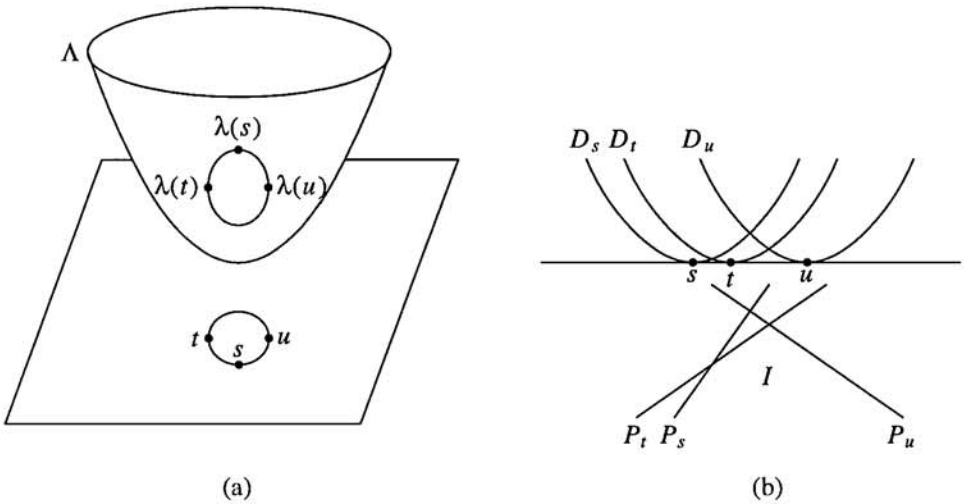
The Delaunay triangulation of S is exactly the orthogonal projection into $\mathbb{R}^2$ of the lower faces of H (a face is ***lower*** if it has a supporting plane with inward normal having positive vertical coordinate). To see this informally, suppose that triangle $\lambda(s)\lambda(t)\lambda(u)$ is a lower facet of H, and that plane P passes through $\lambda(s)\lambda(t)\lambda(u)$. The intersection of P with Λ is an ellipse that projects orthogonally to a circle in $\mathbb{R}^2$ (Figure 20.1.2(a)). Since all other lifted sites are above the plane, all other unlifted sites are outside the circle, and *stu* is a Delaunay triangle. The opposite direction, that a Delaunay triangle is a lower facet, is similar.

For Voronoi diagrams, assign to each site $s = (s_1, s_2)$ the plane

$$P_s = \{(x_1, x_2, x_3) : x_3 = -2x_1 s_1 + s_1^2 - 2x_2 s_2 + s_2^2\}.$$

Let I be the intersection of the lower halfspaces of the planes P_s. The Voronoi diagram is exactly the orthogonal projection into $\mathbb{R}^2$ of the upper faces of I. To

FIGURE 20.1.2
(a) *The intersection of a plane with* Λ *is an ellipse that projects to a circle;* (b) *on any vertical line, the surfaces* $\{D_s\}$ *appear in the same order as the planes* $\{P_s\}$.



see this informally, consider the surfaces

$$D_s = \{(x_1, x_2, x_3) : x_3 = ((x_1 - s_1)^2 + (x_2 - s_2)^2\}$$

(see Figure 20.1.2). Viewed as a function from $\mathbb{R}^2$ into R, D_s gives the squared distance to site s. Furthermore, P_s and D_s differ only by the quadratic term $x_1^2 + x_2^2$, which is independent of s. Hence a point $x \in \mathbb{R}^2$ is in the Voronoi cell of site t iff on the vertical line through x, D_t is lowest among all surfaces $\{D_s\}$. This happens exactly if, on the same line, P_t is lowest among all planes $\{P_s\}$, i.e., x is in the projection of the upper face of I formed by P_t.

COMBINATORIAL COMPLEXITY

In dimension 2, a Delaunay triangulation of $n \geq 3$ sites has at most $2n-5$ triangles and $3n - 6$ edges (and the Voronoi diagram has at most as many vertices and edges, respectively). In dimension $d \geq 3$ the Delaunay triangulation can have $\Theta(n^{\lceil d/2 \rceil})$ faces. For a lower bound example in dimension 3, choose $n/2$ distinct point sites on each of two noncoplanar line segments l and l'. Then there is an empty sphere through each quadruple of sites (a, a', b, b') with a, a' adjacent on l and b, b' adjacent on l'. Since there are $\Omega(n^2)$ such quadruples, there are as many Delaunay tetrahedra. In d dimensions, the maximum number of Voronoi i-faces for n sites is $C_{d-i}(n, d+1) - \delta_{0k}$, where $C_i(n, d+1)$ is the maximum number of i-faces for a convex polytope in $d+1$ dimensions (Section 13.1.4 and Theorem 15.3.4), and δ is the Kronecker delta function.

If point sites are chosen uniformly at random from inside a sphere, then the expected number of faces is linear in the number of sites. In dimension 2, the expected number of triangles is $2n$; in dimension 3, the expected number of tetrahedra is $\sim 6.77n$; in dimension 4, the expected number of 4-simplices is $\sim 31.78n$ [Dwy91]. Similar bounds probably hold for other distributions, but proofs are lacking.

20.2 BASIC ALGORITHMS

Table 20.2.1 lists basic algorithms that compute the Delaunay triangulation of n point sites in $\mathbb{R}^d$ using the Euclidean metric. Using the connection with convexity, any $(d+1)$-dimensional convex hull algorithm can be used to compute a d-dimensional Delaunay triangulation; in fact the divide-and-conquer, incremental, and gift-wrapping algorithms are specialized convex hull algorithms. Running times are given both for worst-case inputs, and for inputs chosen uniformly at random inside a sphere, with expectation taken over input distribution. The Voronoi diagram can be obtained in linear time from the Delaunay triangulation, using the one-one correspondence between their faces. See [Aur91, Ede87, For95, For93] for more citations.

THE RANDOMIZED INCREMENTAL ALGORITHM

The incremental algorithm adds sites one by one, updating the Delaunay triangulation after each addition. The update consists of discovering all Delaunay faces

TABLE 20.2.1 Delaunay Triangulation algorithms in the Euclidean metric for point sites.

ALGORITHM	DIM	WORST CASE	UNIFORM
Flipping	2	$O(n^2)$	
Plane sweep	2	$O(n \log n)$	
Divide-and-conquer	2	$O(n \log n)$	$O(n)$
Randomized incremental	2	$O(n \log n)$	
Randomized incremental	≥ 3	$O(n^{\lceil d/2 \rceil})$	$O(n \log n)$
Gift-wrapping	≥ 2	$O(n^{\lceil d/2 \rceil + 1})$	$O(n)$

whose circumspheres contain the new site. These faces are deleted and the empty region is partitioned into new faces, each of which has the new site as a vertex. See Figure 20.2.1. An efficient algorithm requires a good data structure for finding the faces to be deleted. Then the running time is determined by the total number of face updates, which depends upon site insertion order. The bounds given in Table 20.2.1 are the expected running time of an algorithm that makes a random choice of insertion order, with each insertion permutation equally likely; the bounds for the worst-case insertion order are about a factor of n worse. (For uniform data there is a double expectation, over both insertion order and input distribution.) With additional algorithmic complexity, it is possible to obtain deterministic algorithms with the same worst-case running times [Cha91].

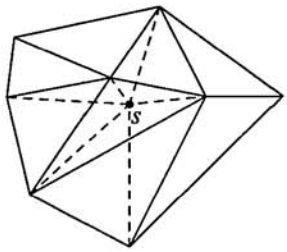


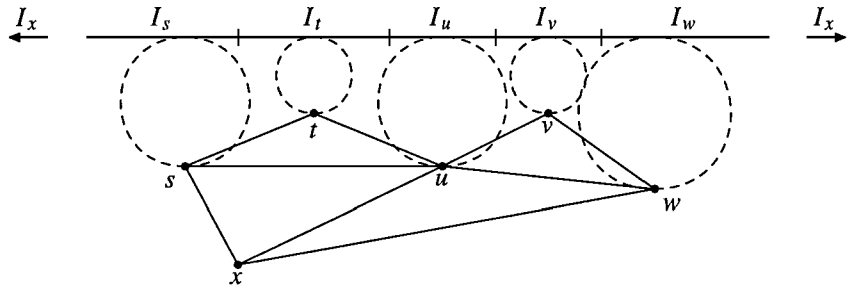
FIGURE 20.2.1
The addition of site s deletes four triangles and adds six (shown dashed).

THE PLANE SWEEP ALGORITHM

The plane sweep algorithm computes a planar Delaunay triangulation using a horizontal line that sweeps upwards across the plane. The algorithm discovers a Delaunay triangle when the sweepline passes through the topmost point of its circumcircle; in Figure 20.2.2, the Delaunay triangles shown have already been discovered. A sweepline data structure stores an ordered list of sites; the entry for site s corresponds to an interval I_s on the sweepline where each maximal empty circle with topmost point in I_s touches site s. The sweepline moves in discrete steps only when the ordered list changes. This happens when a new site is encountered or when a new Delaunay triangle is discovered (at the topmost point of the circumcircle of three sites that are consecutive on the sweepline list). A priority queue is needed to determine the next sweepline move. The running time of the algorithm is $O(n \log n)$

since the sweepline moves $O(n)$ times—once per site and once per triangle—and it costs time $O(\log n)$ per move to maintain the priority queue and sweepline data structure.

FIGURE 20.2.2
The sweepline list is x, s, t, u, v, w, x. The next Delaunay triangle is tuv.



OTHER ALGORITHMS

The divide-and-conquer algorithm uses a splitting line to partition the point set into two equal halves, recursively computes the Delaunay triangulation of each half, and then merges the two subtriangulations in linear time. The gift-wrapping algorithm is a specialization of the convex-hull gift-wrapping algorithm (Chapter 19) to Delaunay triangulations.

Output-sensitive algorithms, with running time approximately proportional to the actual number of Delaunay facets, have remained elusive. See [CSY95] for recent progress.

If the sites form the vertices of a convex polygon, then the Voronoi diagram can be computed in linear time [AGSS89].

IMPLEMENTATIONS

Many of these algorithms have been implemented; see Section 52.2.

20.3 EXTENSIONS

GLOSSARY

Order-k Voronoi diagram: Partitions $\mathbb{R}^d$ on the basis of the first k closest sites (without distinguishing order among them).

Furthest site Voronoi diagram: Partitions $\mathbb{R}^d$ on the basis of the farthest site, or equivalently, the closest $n-1$ of n sites.

Constrained Delaunay triangulation (CDT): The CDT of a set of points

$S \subset \mathbb{R}^2$ using a set of noncrossing ***constraint edges*** E is a triangulation of S extending the edges in E so that the circumcircle of every triangle contains no site that is visible from all three sites defining the triangle.

Power or Laguerre diagram: A Voronoi diagram for sites s_i with weights w_i where the distance from a point x is measured along a tangent to the sphere of radius $\sqrt{w_i}$ centered on s_i.

HIGHER-ORDER VORONOI DIAGRAMS

The order-k Voronoi diagram can be obtained as an appropriate projection of the k-level of an arrangement of hyperplanes (see [Ede87, For93] and Section 21.2 of this Handbook); it can also be obtained as the orthogonal projection of an intersection polytope [AS92]. In dimension 2, the order-k Voronoi diagram has $O(k(n-k))$ faces. In dimensions $d \geq 3$, the sum of the number of faces of the order-j diagrams, $j \leq k$, is $O(n^{\lceil d/2 \rceil} k^{\lfloor d/2 \rfloor + 1})$ [CS89]; finding good bounds for fixed k remains an open problem. See Table 20.3.1 for algorithm bounds.

TABLE 20.3.1 Algorithms for order-k Voronoi diagrams of point sites in the Euclidean metric.

PROBLEM	DIM	TIME
Furthest site	2	$O(n \log n)$
Furthest site	≥ 3	$O(n^{\lceil d/2 \rceil})$
Order-k	2	$O(k(n-k)\log n + n \log^3 n)$
Order-j, $1 \leq j \leq k$	≥ 3	$O(n^{\lceil d/2 \rceil} k^{\lfloor d/2 \rfloor + 1})$

VISIBILITY CONSTRAINTS

Let S be a set of n point sites in $\mathbb{R}^2$ and E a set of noncrossing ***constraint*** edges with endpoints in S. A point $p \in \mathbb{R}^2$ is ***visible*** from a site s if the open segment ps does not intersect any edge of E. A ***constrained Delaunay triangulation*** (CDT) of S using E is a triangulation of S extending the edges in E so that the circumcircle of every triangle contains no site that is visible from all three sites defining the triangle. The CDT is as close as possible to the true Delaunay triangulation, subject to the constraint that the edges in E must be used. See also Section 22.2.

The ***bounded distance*** from a site to a point is Euclidean distance if the point is visible, and infinite otherwise; the ***bounded Voronoi diagram*** of S using E is defined using bounded distance. The bounded Voronoi diagram is dual to a subgraph of the CDT.

Both the CDT and the bounded Voronoi diagram can be computed in time $O(n \log n)$ using either divide-and-conquer or the sweepline paradigm. If the sites and constraint edges are the vertices and edges of a simple polygon, respectively, then the CDT can be computed in linear time [KL93].

There is no obvious generalization of constrained Delaunay triangulations to

dimension $d \geq 3$, since there exist polyhedra in $\mathbb{R}^3$ that cannot be triangulated, at least without using Steiner points.

OTHER DISTANCE MEASURES

Table 20.3.2 lists Voronoi diagram algorithms where "distance" is altered. The distance from a site s_i to a point x can be a function of the Euclidean distance $e(s_i, x)$ and a site-specific real weight w_i.

TABLE 20.3.2 Algorithms for point sites in $\mathbb{R}^2$, other distance measures.

PROBLEM	DISTANCE TO x	TIME
Additive weights	$w_i + e(s_i, x)$	$O(n \log n)$
Multiplicative weights	$w_i e(s_i, x)$	$O(n^2)$
Laguerre or power	$\sqrt{e(s_i, x)^2 - w_i}$	$O(n \log n)$
L_p	$\|\|s_i - x\|\|_p$	$O(n \log n)$
Convex distance function		$O(n \log n)$
Abstract	axiomatic	$O(n \log n)$
Simple polygon	geodesic	$O(n \log^2 n)$
Crystal growth	$w_i \cdot SP(s_i, x)$	$O(n^3 + nS \log S)$

The seemingly peculiar ***power distance*** [Aur87] is the distance from x to the sphere of radius $\sqrt{w_i}$ about s_i along a line tangent to the sphere. Many of the basic Voronoi diagram algorithms extend immediately to the power distance, even in higher dimension.

A ***(polygonal) convex distance function*** [CD85] is defined by a convex polygon C with the origin in its interior. The distance from x to y is the real $r \geq 0$ so that the boundary of $rC + x$ contains y. Polygonal convex distance functions generalize the L_1 and L_∞ metrics (C is a diamond or square, respectively); a polygonal convex distance function is a metric exactly if C is symmetric about the origin.

An ***abstract*** Voronoi diagram [KMM93] is defined by the "bisectors" between pairs of sites, which must satisfy special properties.

The ***geodesic*** distance inside an environment of polygonal obstacles is the length of the shortest path that avoids obstacle interiors. Recent progress using the geodesic metric appears in [HS93].

The ***crystal growth*** Voronoi diagram [SD91] models crystal growth where each crystal has a different growth rate. The distance from a site s_i to a point x in the Voronoi face of s_i is $w_i \cdot SP(s_i, x)$, where w_i is a weight and $SP(s_i, x)$ is the shortest path distance lying entirely within the Voronoi face of s_i. The parameter S in the running time measures the time to approximate bisectors numerically.

OTHER SITES

The Voronoi diagram of a set of n line segment sites can be computed in time $O(n \log n)$ using the sweepline method or the divide-and-conquer method; the me-

dial axis of a polygon or polygonal region can be obtained from the Voronoi diagram of its constituent line segments. The divide-and-conquer algorithm extends to circular-arc segments as well. The Voronoi diagram of a set of circles can be computed using an additively-weighted point-site algorithm.

MOTION PLANNING

The motion planning problem is to find a collision-free path for a robot in an environment filled with obstacles. The Voronoi diagram of the obstacles is quite useful, since it gives a lower-dimensional skeleton of maximal clearance from the obstacles. In many cases the shape of the robot can be used to define an appropriate metric for the Voronoi diagram. See Section 40.2 for more on the use of Voronoi diagrams in motion planning.

20.4 IMPORTANT PROPERTIES

ROUNDNESS

The Delaunay triangulation is "round," that is, skinny simplices are avoided. This can be formalized in two dimensions by Lawson's classic result: over all possible triangulations, the Delaunay triangulation maximizes the minimum angle of any triangle. No generalization using angles is known in higher dimension. However, define the ***enclosing radius*** of a simplex as the minimum radius of an enclosing sphere. In any dimension and over all possible triangulations of a point set, the Delaunay triangulation minimizes the maximum enclosing radius of any simplex [Raj94]. Also see Section 22.4 on mesh generation.

VISIBILITY DEPTH ORDERING

Choose a viewpoint v and a family of disjoint convex objects in $\mathbb{R}^d$. Object A is *in front of* object B from v if there is a ray starting at v that intersects A and then B in that order. Though an arbitrary family can have cycles in the "in front of" relation, the relation is acyclic for the faces of the Delaunay triangulation, for any viewpoint and any dimension [Ede90].

An application comes from computer graphics. The ***painter's algorithm*** renders three-dimensional objects in back to front order, with later objects simply overpainting the image space occupied by earlier objects. A valid rendering order always exists if the "in front of" relation is acyclic, as is the case if the objects are Delaunay tetrahedra, or a subset of a set of Delaunay tetrahedra.

SUBGRAPH RELATIONSHIPS

The edges of a Delaunay triangulation form a graph DT whose vertices are the

sites. In any dimension, the following subgraph relations hold:

$$EMST \subseteq RNG \subseteq GG \subseteq DT$$

where EMST is the Euclidean minimum spanning tree, RNG is the relative neighborhood graph, and GG is the Gabriel graph. See Section 43.2 on pattern recognition.

DILITATION

A geometrically embedded graph G has ***dilatation*** c if for any two vertices, the shortest path distance along the edges of G is at most c times the Euclidean distance between the vertices. In $\mathbb{R}^2$, the edge set of the Delaunay triangulation has dilatation at most ~ 2.42; with an equilateral-triangle convex distance function, the dilatation is at most 2.

INTERPOLATION

Suppose each point site $s_i \in S \subset \mathbb{R}^d$ has an associated function value f_i. For $p \in \mathbb{R}^d$ define $\lambda_i(p)$ as the proportion of the area of s_i's Voronoi cell that would be removed if p were added as a site. Then the ***natural neighbor*** interpolant $f(p) = \sum \lambda_i(p) f_i$ is C^0, and C^1 except at sites. A more complex C^1 interpolant can be obtained as well [Sib81].

Alternatively, for a triangulation of S in $\mathbb{R}^2$, consider the piecewise linear surface defined by linear interpolation over each triangle. Over all possible triangulations, the Delaunay triangulation minimizes the roughness of the resulting surface, where ***roughness*** is the square of the L_2 norm of the gradient of the surface, integrated over the triangulation [Rip90].

20.5 SOURCES AND RELATED MATERIAL

FURTHER READING

[Aur91, For95]: Survey papers that cover many aspects of Delaunay triangulations and Voronoi diagrams.

[OBS92]: A book entirely devoted to Voronoi diagrams, with an extensive discussion of applications.

[Ede87, PS85]: Basic references for geometric algorithms.

RELATED CHAPTERS

Chapter 19: Convex hull computations
Chapter 21: Arrangements

Chapter 22: Triangulations
Chapter 40: Algorithmic motion planning
Chapter 43: Pattern recognition

REFERENCES

[AGSS89] A. Aggarwal, L.J. Guibas, J. Saxe, and P.W. Shor. A linear-time algorithm for computing the Voronoi diagram of a convex polygon. *Discrete Comput. Geom.*, 4:591–604, 1989.

[AS92] F. Aurenhammer and O. Schwarzkopf. A simple randomized incremental algorithm for computing higher order Voronoi diagrams. *Internat. J. Comput. Geom. Appl.*, 2:363–381, 1992.

[Aur87] F. Aurenhammer. Power diagrams: properties, algorithms, and applications. *SIAM J. Comput.*, 16:78–96, 1987.

[Aur91] F. Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23:345–405, 1991.

[CD85] L.P. Chew and R.L. Drysdale. Voronoi diagrams based on convex distance functions. In *Proc. 1st Annu. ACM Sympos. Comput. Geom.*, pages 234–244, 1985.

[Cha91] B. Chazelle. An optimal convex hull algorithm and new results on cuttings. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 29–38, 1991.

[CS89] K.L. Clarkson and P.W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.

[CSY95] T.M.Y. Chan, J. Snoeyink, and C.K. Yap. Output-sensitive construction of polytopes in four dimensions and clipped Voronoi diagrams in three. In *Proc. 6th ACM-SIAM Sympos. Discrete Algorithms*, pages 282–291, 1995.

[Dwy91] R. Dwyer. Higher-dimensional Voronoi diagrams in linear expected time. *Discrete Comput. Geom.*, 6:343–367, 1991.

[Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer-Verlag, Berlin, 1987.

[Ede90] H. Edelsbrunner. An acyclicity theorem for cell complexes in d dimensions. *Combinatorica*, 10:251–260, 1990.

[For93] S. Fortune. Progress in computational geometry. In R. Martin, editor, *Directions in Geometric Computing*, pages 81–128. Information Geometers, Winchester, 1993.

[For95] S. Fortune. Voronoi diagrams and Delaunay triangulations. In F. Hwang and D.Z. Du, editors, *Computing in Euclidean Geometry* (Second Edition), pages 225–265. World Scientific, Singapore, 1995.

[HS93] J. Hershberger and S. Suri. Efficient computation of Euclidean shortest paths in the plane. In *Proc. 34th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 508–517, 1993.

[KL93] R. Klein and A. Lingas. A linear-time randomized algorithm for the bounded Voronoi diagram of a simple polygon. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 124–132, 1993.

[KMM93] R. Klein, K. Mehlhorn, and S. Meiser. Randomized incremental construction of abstract Voronoi diagrams. *Comput. Geom. Theory Appl.*, 3:157–184, 1993.

[OBS92] A. Okabe, B. Boots, and K. Sugihara. *Spatial Tesselations: Concepts and Applications of Voronoi Diagrams.* Wiley, Chichester, 1992.

[PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry.* Springer-Verlag, New York, 1985.

[Raj94] V.T. Rajan. Optimality of the Delaunay triangulation in R^d. *Discrete Comput. Geom.*, 12:189–202, 1994.

[Rip90] S. Rippa. Minimal roughness property of the Delaunay triangulation. *Comput. Aided Design*, 7:489–497, 1990.

[SD91] B. Schaudt and R.L. Drysdale. Multiplicatively weighted crystal growth Voronoi diagram. In *Proc. 7th. Annu. ACM Sympos. Comput. Geom.*, pages 214–223, 1991.

[Sib81] R. Sibson. Natural neighbor interpolation. In V. Barnett, editor, *Interpreting Multivariate Data*, pages 21–36. Wiley, Chichester, 1981.

21 ARRANGEMENTS

Dan Halperin

INTRODUCTION

Given a finite collection $\mathcal{S}$ of geometric objects such as hyperplanes or spheres in $\mathbb{R}^d$, the *arrangement* $\mathcal{A}(\mathcal{S})$ is the decomposition of $\mathbb{R}^d$ into connected open cells of dimensions $0, 1, \ldots, d$ induced by $\mathcal{S}$. Besides being interesting in their own right, arrangements of hyperplanes have served as a unifying structure for many problems in discrete and computational geometry. With the recent advances in the study of arrangements of curved (algebraic) surfaces, arrangements have emerged as the underlying structure of geometric problems in a variety of "physical world" application domains such as robot motion planning and computer vision. This chapter is devoted to arrangements of hyperplanes and of curved surfaces in low-dimensional Euclidean space, with an emphasis on combinatorics and algorithms.

In the first section we introduce basic terminology and combinatorics of arrangements. In Section 21.2 we describe substructures in arrangements and their combinatorial complexity. Section 21.3 deals with data structures for representing arrangements and with special refinements of arrangements. The following two sections focus on algorithms: algorithms for constructing full arrangements are described in Section 21.4, and algorithms for constructing substructures in Section 21.5. Situations where arrangements have lower complexity than the general worst-case bounds are presented in Section 21.6. In Section 21.7 we discuss the relation between arrangements and other structures. Several applications of arrangements are reviewed in Section 21.8.

21.1 BASICS

In this section we review basic terminology and combinatorics of arrangements, first for arrangements of hyperplanes and then for arrangements of curves and surfaces.

21.1.1 ARRANGEMENTS OF HYPERPLANES

GLOSSARY

Arrangement of hyperplanes: Let $\mathcal{H}$ be a finite set of hyperplanes in $\mathbb{R}^d$. The hyperplanes in $\mathcal{H}$ induce a decomposition of $\mathbb{R}^d$ (into connected open cells), the arrangement $\mathcal{A}(\mathcal{H})$. A d-dimensional cell in $\mathcal{A}(\mathcal{H})$ is a maximal connected region of $\mathbb{R}^d$ not intersected by any hyperplane in $\mathcal{H}$; any k-dimensional cell in $\mathcal{A}(\mathcal{H})$, for $0 \leq k \leq d-1$, is a maximal connected region in the intersection of a subset

of the hyperplanes in $\mathcal{H}$ that is not intersected by any other hyperplane in $\mathcal{H}$. It follows that any cell in an arrangement of hyperplanes is convex.

Simple arrangement: An arrangement $\mathcal{A}(\mathcal{H})$ of a set $\mathcal{H}$ of n hyperplanes in $\mathbb{R}^d$, with $n \geq d$, is called simple if every d hyperplanes in H meet in a single point and if any $d+1$ hyperplanes have no point in common.

Vertex, edge, face, facet: $0, 1, 2$, and $(d-1)$-dimensional cell of the arrangement, respectively.

k-cell : A k-dimensional cell in the arrangement.

Combinatorial complexity of an arrangement: The overall number of cells of various dimensions in the arrangement.

EXAMPLE: AN ARRANGEMENT OF LINES

Let $\mathcal{L}$ be a finite set of lines in the plane, and let $\mathcal{A}(\mathcal{L})$ be a simple arrangement induced by $\mathcal{L}$. A 0-dimensional cell (a vertex) is the intersection point of two lines in $\mathcal{L}$; a 1-dimensional cell (an edge) is a maximal connected portion of a line in $\mathcal{L}$ that is not intersected by any other line in $\mathcal{L}$; and a 2-dimensional cell (a face) is a maximal connected region of $\mathbb{R}^2$ not intersected by any line in $\mathcal{L}$. See Figure 21.1.1.

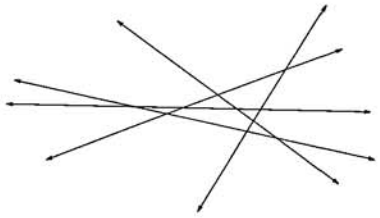


FIGURE 21.1.1
A simple arrangement of 5 *lines.*
It has 10 *vertices,* 25 *edges (*10 *of which are unbounded),*
and 16 *faces (*10 *of which are unbounded).*

COUNTING CELLS

A fundamental question in the study of arrangements is how complex a certain arrangement (or portion of it) can be. Answering this question is often a prerequisite to the analysis of algorithms on arrangements.

THEOREM 21.1.1

Let $\mathcal{H}$ be a set of hyperplanes in $\mathbb{R}^d$. The maximum number of k-dimensional cells in the arrangement $\mathcal{A}(\mathcal{H})$, for $0 \leq k \leq d$, is

$$\sum_{i=0}^{k} \binom{d-i}{k-i} \binom{n}{d-i}.$$

The maximum is attained exactly when $\mathcal{A}(\mathcal{H})$ is simple.

We assume henceforth that the dimension d is a (small) constant. With few exceptions, we will not discuss *exact* combinatorial complexity bounds, as in the

theorem above, but rather use the big-O notation. Theorem 21.1.1 implies the following:

COROLLARY 21.1.2

The maximum combinatorial complexity of an arrangement of n hyperplanes in $\mathbb{R}^d$ is $O(n^d)$. If the arrangement is simple its complexity is $\Theta(n^d)$. In these bounds the constant of proportionality depends on d.

21.1.2 ARRANGEMENTS OF CURVES AND SURFACES

We now introduce more general arrangements, allowing for objects that are nonlinear and/or bounded. We distinguish between planar arrangements and arrangements in three or higher dimensions. For planar arrangements we require only that the objects defining the arrangement be x-monotone Jordan arcs with a constant maximum number of intersections per pair. For arrangements of surfaces in three or higher dimensions we require that the surfaces be algebraic of constant maximum degree (a more precise definition is given below). This requirement simplifies the analysis and computation of such arrangements, and it does not seem to be too restrictive, as in most applications the arrangements that arise are of low-degree algebraic surfaces.

In both cases we assume that the objects (curves or surfaces) are in general position. This is a generalization to the current setting of the simplicity assumption for hyperplanes made above. All the other definitions in the Glossary carry over to arrangements of curves and surfaces.

PLANAR ARRANGEMENTS

Let $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$ be a collection of Jordan arcs in the xy-plane, such that each arc is ***x-monotone*** (i.e., every line parallel to the y-axis intersects an arc in at most one point) and each pair of arcs in $\mathcal{C}$ intersect in at most s points for some fixed constant s. The arrangement $\mathcal{A}(\mathcal{C})$ is the decomposition of the plane into open cells of dimensions $0, 1$, and 2 induced by the arcs in $\mathcal{C}$. Here, a 0-dimensional cell (a vertex) is either an endpoint of one arc or an intersection point of two arcs. See Figure 21.1.2.

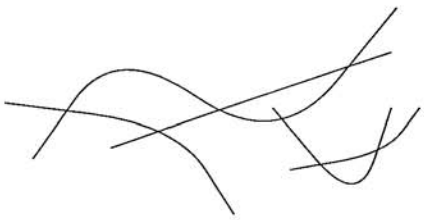


FIGURE 21.1.2
A simple arrangement of 5 *bounded arcs, where* $s = 2$. *It has* 17 *vertices (*10 *of which are arc endpoints),* 19 *edges, and* 4 *faces (one of which is unbounded).*

We assume that the arcs in $\mathcal{C}$ are in ***general position***, namely, that each intersection of a pair of arcs in $\mathcal{C}$ is either a common endpoint or a transversal intersection at a point in the relative interior of both arcs, and that no three arcs intersect at a common point.

THEOREM 21.1.3

If $\mathcal{C}$ is a collection of n Jordan arcs as defined above, then the maximum combinatorial complexity of the arrangement $\mathcal{A}(\mathcal{C})$ is $O(n^2)$. There are such arrangements whose complexity is $\Theta(n^2)$. In these bounds the constant of proportionality depends linearly on s.

THREE AND HIGHER DIMENSIONS

We denote the coordinate axes of $\mathbb{R}^d$ by $x_1, x_2, \ldots, x_d$. For a collection $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ of (hyper)surface patches in $\mathbb{R}^d$ we make the following assumptions:

1. Each surface patch is contained in an algebraic surface of constant maximum degree.
2. The boundary of each surface patch is determined by at most some constant number of algebraic surface patches of constant maximum degree each.
3. Every d surface patches in $\mathcal{S}$ meet in at most s points.
4. Each surface patch is ***monotone*** in $x_1, \ldots, x_{d-1}$, namely every line parallel to the x_d-axis intersects the surface patch in at most one point.
5. The surface patches in $\mathcal{S}$ are in *general position* (e.g., see [Sha94, p. 328]).

We use the simplified term ***arrangement of surfaces*** to refer to arrangements whose defining objects satisfy the assumptions above. A few remarks regarding these assumptions:

- Assumptions (1) and (2), together with the general position assumption (5), imply that every d-tuple of surfaces meet in at most some constant number of points. One can bound this number using Bézout's Theorem (see Chapter 29). The bound s on the number of d-tuple intersection points turns out to be a crucial parameter in the combinatorial analysis of substructures in arrangement. Often, one can get a better estimate for s than the bound implied by Bézout's theorem.
- Assumption (4) is used in results cited below. It can however be easily relaxed without affecting these results. If a surface patch does not satisfy this assumption, it can be decomposed into pieces that satisfy the assumption, and by assumptions (1) and (2) the number of these pieces will be bounded by a constant.
- Assumption (5) often does not affect the worst-case combinatorial bounds obtained for arrangements or their substructures, because it can be shown that the asymptotically highest complexity is obtained when the surfaces are in general position [Sha94]. For algorithms, this assumption is more problematic. There are general relaxation methods but these seem to introduce new difficulties [Sei94].

THEOREM 21.1.4

Given a collection $\mathcal{S}$ of n surfaces in $\mathbb{R}^d$, as defined above, the maximum combinatorial complexity of the arrangement $\mathcal{A}(\mathcal{S})$ is $O(n^d)$. There are such arrangements whose complexity is $\Theta(n^d)$. The constant of proportionality in these bounds depends on d and on the maximum algebraic degree of the surfaces and of their boundaries.

ARRANGEMENTS ON CURVED SURFACES

Although we do not discuss such arrangements directly in this chapter, many of the combinatorial and algorithmic results that we survey carry over to arrangements on curved surfaces with only slight adjustments. Arrangements on spheres are especially prevalent in applications. The ability to analyze or construct arrangements on curved surfaces is implicitly assumed in the results for arrangements of surfaces in Euclidean space, since we often need to consider the lower-dimensional arrangement induced on a surface by its intersections with all the other surfaces that define the arrangement.

ADDITIONAL TOPICS

We focus in this chapter on simple arrangements. We note, however, that nonsimple arrangements raise interesting questions; see, for example, [PA95, Chapter 11] and [Szé]. Another noteworthy topic that, for lack of space, we will not cover here is ***combinatorial equivalence*** of arrangements [BLS+93].

21.2 SUBSTRUCTURES IN ARRANGEMENTS

A substructure in an arrangement (i.e., a portion of an arrangement), rather than the entire arrangement, may be sufficient to solve a problem at hand. Also, the analysis of several algorithms for constructing arrangements relies on combinatorial bounds for substructures. We survey substructures that are known in general to have significantly smaller complexity than that of the entire arrangement, together with combinatorial complexity bounds for these substructures. For simplicity, some of the substructures are defined below only for the planar case.

GLOSSARY

Let $\mathcal{C}$ be a collection of n x-monotone Jordan arcs as defined in Section 21.1.

Lower (upper) envelope: For this definition we regard each curve c_i in $\mathcal{C}$ as the graph of a continuous univariate function $c_i(x)$ defined on an interval. The lower envelope Ψ of the collection $\mathcal{C}$ is the pointwise minimum of these functions: $\Psi(x) = \min c_i(x)$, where the minimum is taken over all functions defined at x. (The lower envelope is the 0-level of the arrangement $\mathcal{A}(\mathcal{C})$; see below.) Similarly, the upper envelope of the collection $\mathcal{C}$ is defined as the pointwise maximum of these functions. Lower and upper envelopes are completely symmetric structures, and from this point on we will discuss only lower envelopes.

Minimization diagram of $\mathcal{C}$: The subdivision of the x-axis into maximal intervals so that on each interval the same subset of functions attains the minimum.

Zone: For an additional curve γ, the collection of faces of the arrangement $\mathcal{A}(\mathcal{C})$ intersected by γ. See Figure 21.4.1. The zone is sometimes called the ***horizon***.

Single cell: In this section, a d-cell in an arrangement in $\mathbb{R}^d$.

Sides and borders: Let e be an edge in an arrangement of lines, and let l be

the line containing e. The line l divides the plane into two halfplanes h_1, h_2. We regard e as two-sided, and denote the two sides by (e, h_1) and (e, h_2). The edge e is on the boundary of two faces f_1 and f_2 in the arrangement. e is said to be a ***1-border*** of either face, marked (e, f_1) and (e, f_2), respectively. Similarly a vertex in a simple arrangement of lines has four sides, and it is a ***0-border*** of four faces. The definition extends to arrangements of hyperplanes in higher dimensions and to arrangements of curved surfaces.

k-level: We assume here, for simplicity, that the curves are unbounded; the definition can be extended to the case of bounded curves. A point p in the plane is said to be at level k, if there are exactly k curves in $\mathcal{C}$ lying strictly below p (i.e., a ray emanating from p in the negative y direction intersects exactly k curves in $\mathcal{C}$). The level of an (open) edge e in $\mathcal{A}(\mathcal{C})$ is the level of any point of e. The k-level of $\mathcal{A}(\mathcal{C})$ is the closure of the union of edges of $\mathcal{A}(\mathcal{C})$ that are at level k; see Figure 21.2.1. The ***at-most-k-level*** of $\mathcal{A}(\mathcal{C})$, denoted $(\leq \boldsymbol{k})$***-level***, is the union of points in the plane at level j, for $0 \leq j \leq k$. Different texts use slight variations of the above definitions. In particular, in some texts the ray is directed upwards thus counting the levels from top to bottom. k-levels in arrangements of hyperplanes are closely related to k-sets in point configurations; see Chapter 1.

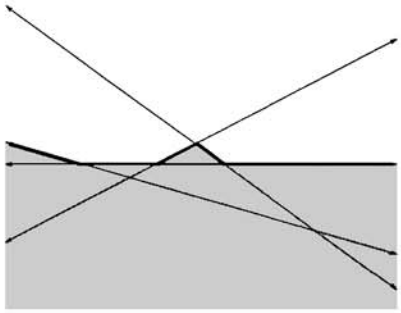


FIGURE 21.2.1
The bold line is the 2-level of the arrangement of four lines. The shaded region is the (≤ 2)*-level of the arrangement.*

Union boundary: If each surface s in an arrangement in $\mathbb{R}^d$ is the boundary of a d-dimensional object, then the boundary of the union of the objects is another interesting substructure. The study of union boundary has largely been motivated by robot motion planning problems; for details see Chapter 40.

$\boldsymbol{\alpha(n)}$***:*** The extremely slowly-growing functional inverse of Ackermann's function. See Section 40.4.

MEASURING THE COMPLEXITY OF A SUBSTRUCTURE

For an arrangement in $\mathbb{R}^d$, if a substructure consists of a collection C of d-cells, its combinatorial complexity is defined to be the overall number of cells of any dimension on the boundary of each of the d-cells in C. This means that we count certain cells of the arrangement with multiplicity. For example, for the zone of a line l in an arrangement of lines, each edge of the arrangement that intersects l will be counted twice. However, since we assume that our arrangements reside in a fixed (low) dimensional space, this only implies a constant multiplicative factor in

our count.

The complexity of the lower envelope of an arrangement is defined to be the complexity of its minimization diagram. In three or higher dimensions, this means that we count features that do not appear in the original arrangement. For example, in the lower envelope of a collection of triangles in 3-space, the projection of the edges of two distinct triangles may intersect in the minimization diagram although the two triangles are disjoint in 3-space.

The complexity of a k-level in an arrangement is defined in a similar way to the complexity of an envelope. The complexity of the $(\leq k)$-level is defined as the overall number of cells of the arrangement that lie in the region of space whose points are at level at-most-k.

COMBINATORIAL COMPLEXITY BOUNDS FOR SUBSTRUCTURES

In the rest of this section we list bounds on the maximum combinatorial complexity of substructures. For lines, hyperplanes, Jordan arcs, and surfaces, these are arranged in Tables 21.2.1, 21.2.2, 21.2.3, and 21.2.4, respectively. Bounds in the tables using ϵ should read "for any $\epsilon > 0$."

TABLE 21.2.1 Substructures in arrangements of n lines in the plane.

SUBSTRUCTURE	BOUND	NOTES
Envelope	n edges	
Single face	n edges	
Zone of a line	$\Theta(n)$	see [Ede87] for an exact bound on the number of 0- and 1-borders
m faces	$\Theta(m^{2/3}n^{2/3} + m + n)$	upper bound [CEG+90]; lower bound [Ede87]
k-level	$O(n\sqrt{k+1}/\log^*(k+1))$	[PSS92]
	$\Omega(n\log(k+1))$	[EW85]
$(\leq k)$-level	$\Theta(n(k+1))$	[AG86]

TABLE 21.2.2 Substructures in arrangements of n hyperplanes in $\mathbb{R}^d$.

SUBSTRUCTURE	BOUND	NOTES
Envelope	$\Theta(n^{\lfloor \frac{d}{2} \rfloor})$	upper bound theorem [McM70]
Zone of a hyperplane	$\Theta(n^{d-1})$	[ESS93]
Single cell	$\Theta(n^{\lfloor \frac{d}{2} \rfloor})$	upper bound theorem [McM70]
m cells	$O(m^{1/2}n^{d/2}\log^{(\lfloor \frac{d}{2} \rfloor - 1)/2} n)$	this bound is almost tight [AMS94b]; for bounds on the number of facets see [AA92]
$(\leq k)$-level	$\Theta(n^{\lfloor d/2 \rfloor}(k+1)^{\lceil d/2 \rceil})$	[CS89]

Aronov et al. [APS93] showed that the maximum complexity of the zone of a p-dimensional algebraic surface of constant maximum degree in an arrangement of

hyperplanes in $\mathbb{R}^d$ is $O(n^{\lfloor (d+p)/2 \rfloor} \log^{\gamma} n)$, where $\gamma = d + p \pmod 2$. This bound is almost tight in the worst case.

CURVES

For a collection $\mathcal{C}$ of n well-behaved curves as defined in Section 21.1, the complexity bounds for certain substructures involve functions related to *Davenport-Schinzel sequences*. The function $\lambda_s(n)$ is defined as the maximum length of a Davenport-Schinzel sequence of order s on n symbols, and it is almost linear in n for any fixed s. Davenport-Schinzel sequences play a central role in the analysis of substructures of arrangements of curves and surfaces. See Section 40.4 for more details.

THEOREM 21.2.1

For a set $\mathcal{C}$ of n x-monotone Jordan arcs such that each pair intersects in at most s points, the maximum number of intervals in the minimization diagram of $\mathcal{C}$ is $\lambda_{s+2}(n)$. If the curves are unbounded, then the maximum number of intervals is $\lambda_s(n)$.

The connection between a zone and a single cell. As observed in [EGP+92], a bound on the complexity of a single cell in general arrangements of arcs implies the same asymptotic bound on the complexity of the zone of an additional well-behaved curve γ in the arrangement; "well-behaved" meaning that γ does not intersect any curve in $\mathcal{C}$ more than some constant number of times. This observation extends to higher dimensions and is exploited in the result for zones in arrangements of surfaces [HS95a].

The results in Table 21.2.3 are for Jordan arcs (bounded curves). There are slightly better bounds in the case of unbounded curves.

TABLE 21.2.3 Substructures in arrangements of n Jordan arcs.

SUBSTRUCTURE	BOUND	NOTES
Envelope	$\Theta(\lambda_{s+2}(n))$	
Single face, zone	$\Theta(\lambda_{s+2}(n))$	[GSS89]
m cells	$O(m^{1/2}\lambda_{s+2}(n))$	[EGP+92]
	$\Omega(m^{2/3}n^{2/3})$	lower bound for lines
$(\leq k)$-level	$\Theta((k+1)^2\lambda_{s+2}(\lfloor \frac{n}{k+1} \rfloor))$	[Sha91]

ADDITIONAL COMBINATORIAL BOUNDS

The following bounds, while not bounds on the complexity of substructures, are useful in the analysis of algorithms for computing substructures and in obtaining other combinatorial bounds on arrangements.

TABLE 21.2.4 Substructures in arrangements of n surfaces.

OBJECTS	SUBSTRUCTURE	BOUND	NOTES
Surfaces in $\mathbb{R}^d$	lower envelope	$O(n^{d-1+\epsilon})$	[HS94],[Sha94]
Surfaces in $\mathbb{R}^3$	single cell, zone	$O(n^{2+\epsilon})$	[HS95a]
$(d-1)$-simplices in $\mathbb{R}^d$	lower envelope	$\Theta(n^{d-1}\alpha(n))$	[Ede89]
	single cell, zone	$O(n^{d-1}\log n)$	[Tag95]
$(d-1)$-spheres in $\mathbb{R}^d$	lower envelope, single cell	$\Theta(n^{\lceil \frac{d}{2} \rceil})$	linearization

Sum of squares. Let $\mathcal{H}$ be a collection of n hyperplanes in $\mathbb{R}^d$. For each d-cell c of the arrangement $\mathcal{A}(\mathcal{H})$, let $f(c)$ denote the number of cells of any dimension on the boundary of c. Aronov et al. [AMS94b] show that $\sum_c f^2(c) = O(n^d \log^{\lfloor \frac{d}{2} \rfloor - 1} n)$, where the sum extends over all d-cells of the arrangement. They use it to obtain bounds on the complexity of m cells in the arrangement.

Overlay of envelopes. For two sets A and B of objects in $\mathbb{R}^d$, the complexity of the *overlay of envelopes* is defined as the complexity of the subdivision of $\mathbb{R}^{d-1}$ induced by superposing the minimization diagram of A on that of B. Given two sets $\mathcal{C}_1$ and $\mathcal{C}_2$ of n x-monotone Jordan arcs each, such that no pair of (the collection of $2n$) arcs intersects more than s times, the complexity of the overlay is easily seen to be $\Theta(\lambda_{s+2}(n))$. In 3-space, given two sets of n well-behaved surfaces each, the complexity of the overlay is $O(n^{2+\epsilon})$ [ASS95]. The bound is applied to obtain a simple divide-and-conquer algorithm for computing the envelope in 3-space, and for obtaining bounds on the complexity of *transversals* (see Chapter 4).

OPEN PROBLEMS

1. What is the complexity of the k-level in an arrangement of lines in the plane? For the gap between the known lower and upper bounds see Table 21.2.1. This is a long-standing open problem in combinatorial geometry.

2. What is the complexity of m faces in an arrangement of well-behaved Jordan arcs? For lines a tight bound is known, whereas for curves a considerable gap still exists—see Table 21.2.3.

3. What is the complexity of the boundary of the union of n infinite unit cylinders in 3-space (namely, each cylinder is the Minkowski sum of a line in 3-space and a unit ball)? Only trivial bounds are known to date: the known upper bound is $O(n^3)$ (Theorem 21.1.4), and a lower bound of $\Omega(n^2)$ is easy to construct.

21.3 REPRESENTATIONS AND DECOMPOSITIONS

Before describing algorithms for arrangements in the next sections, we discuss how to represent an arrangement. The appropriate data structure for representing an

arrangement depends on its intended use. Two typical ways of using arrangements are: (i) traversing the entire arrangement cell by cell; and (ii) directly accessing certain cells of the arrangement. We will present three structures, each providing a method for traversing the entire arrangement: the *incidence graph*, the *cell-tuple structure*, and the *complete skeleton*. We will then discuss refined representations that further subdivide an arrangement into subcells. These refinements are essential to allow for efficient access to cells of the arrangement. For algebraic geometry-oriented representations and decompositions see Chapters 29 and 40.

GLOSSARY

Let $\mathcal{S}$ be a collection of surfaces in $\mathbb{R}^d$ (or curves in $\mathbb{R}^2$) as defined in Section 21.1, and $\mathcal{A}(\mathcal{S})$ the arrangement induced by $\mathcal{S}$. Let c_1 be a k_1-dimensional cell of $\mathcal{A}(\mathcal{S})$ and c_2 a k_2-dimensional cell of $\mathcal{A}(\mathcal{S})$.

Subcell, supercell: If $k_2 = k_1 + 1$ and c_1 is on the boundary of c_2, then c_1 is a subcell of c_2, and c_2 is a supercell of c_1.

(−1)-dimensional cell, (d+1)-dimensional cell: Most representations assume the existence of two additional cells in an arrangement. The unique (-1)-dimensional is a subcell of every vertex (0-dimensional cell) in the arrangement, and the unique $(d+1)$-dimensional cell is a supercell of all the d-dimensional cells in the arrangement.

Incidence: If c_1 is a subcell of c_2, then c_1 and c_2 are *incident* to one another. We say that c_1 and c_2 define an ***incidence***.

21.3.1 REPRESENTATIONS

INCIDENCE GRAPH

The incidence graph (sometimes called the ***facial lattice***) of the arrangement $\mathcal{A}(\mathcal{S})$ is a graph $G = (V, E)$ where there is a node in V for every k-cell of $\mathcal{A}(\mathcal{S})$ $-1 \leq k \leq d + 1$, and an arc between two nodes if the corresponding cells are incident to one another (cf. Figure 13.1.3). For an arrangement of n surfaces in $\mathbb{R}^d$ the number of nodes in V is $O(n^d)$ by Theorem 21.1.4. This is also a bound on the number of arcs in E: every cell (besides the (-1)-dimensional cell) in an arrangement $\mathcal{A}(\mathcal{S})$ in general position has at most a constant number of supercells. For an exact bound in the case of hyperplanes, see [Ede87, Section 1.2].

CELL-TUPLE STRUCTURE

While the incidence graph captures all the cells in an arrangement and their connectivity, it misses *order* information between cells. For example, there is a natural order among the edges that appear along the boundary of a face in a planar arrangement. This leads to the ***cell-tuple structure*** [Bri93] which is a generalization to any dimension of the two-dimensional ***quad-edge structure*** of Guibas and Stolfi [GS85] and the three-dimensional ***facet-edge structure*** of Dobkin and

Laszlo [DL89]. The cell-tuple structure gives a simple and uniform representation of the incidence and ordering information in the arrangement.

SKELETON

Let $\mathcal{H}$ be a finite set of hyperplanes in $\mathbb{R}^d$. A ***skeleton*** in the arrangement $\mathcal{A}(\mathcal{H})$ is a connected subset of edges and vertices of the arrangement. The ***complete skeleton*** is the union of all the edges and vertices of the arrangement. Edelsbrunner [Ede87] proposes a representation of the skeleton as a digraph, which allows for a systematic traversal of the entire arrangement (in the case of a complete skeleton) or a substructure of the arrangement. Using a one-dimensional skeleton to represent an arrangement in an arbitrary-dimensional space is a notion that appears also in algebro-geometric representations. There, however, the skeleton, or ***roadmap***, is far more complicated (indeed it represents more general arrangements); see Chapter 40.

21.3.2 DECOMPOSITIONS

A raw arrangement may still be an unwieldy structure as cells may have complicated shapes and many bounding subcells. It is often desirable to decompose the cells of the arrangement into subcomponents so that each subcomponent has a constant descriptive complexity and is homeomorphic to a ball. Besides the obvious convenience that such a decomposition offers (just like a triangulation of a simple polygon), it turns out to be crucial to the design and analysis of randomized algorithms for arrangements, as well as to combinatorial analysis of arrangements.

For a decomposition to be useful, we aim to add as few extra features as possible. The three decompositions described in this section have the property that the complexity of the decomposed arrangement is asymptotically close to (sometimes the same as) that of the original arrangement. (This is still not known for the *vertical decomposition* in higher dimensions—see the open problem below.)

BOTTOM VERTEX DECOMPOSITION OF ARRANGEMENTS

Consider an arrangement of lines $\mathcal{A}(\mathcal{L})$ in the plane. For a face f let $v_b = v_b(f)$ be the bottommost vertex of f (the vertex with lowest y coordinate). Extend an edge from v_b to each vertex on the boundary of f that is not incident to an edge incident to v_b; see Figure 21.3.1. Repeat for all faces of $\mathcal{A}(\mathcal{L})$. The original arrangement, together with the added edges, constitute the ***bottom vertex decomposition*** of $\mathcal{A}(\mathcal{L})$, which is a decomposition of $\mathcal{A}(\mathcal{L})$ into triangles. The notion extends to arrangements of hyperplanes in higher dimensions, and it is carried out recursively [Cla88]. The combinatorial complexity of the decomposition is asymptotically the same as that of the original arrangement.

VERTICAL DECOMPOSITION

The bottom vertex decomposition does not in general extend to arrangements of nonlinear objects. Fortunately there is an alternative, rather simple, decomposition method that applies to almost any reasonable arrangement. This is the ***vertical***

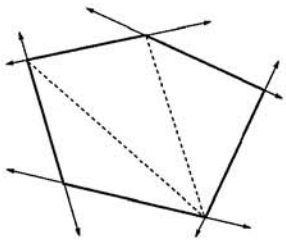


FIGURE 21.3.1
The bottom vertex decomposition of a face in an arrangement of lines.

decomposition or ***trapezoidal decomposition.*** See Figure 21.3.2. It is optimal for two-dimensional arrangements, namely its complexity is asymptotically the same as that of the underlying arrangement. It is near-optimal in three dimensions. In higher dimensions it is still the general decomposition method that is known to have the best (lowest) complexity.

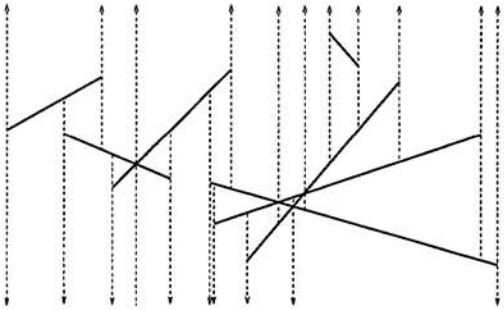


FIGURE 21.3.2
The vertical decomposition of an arrangement of segments: a vertical line segment is extended upward and downward from each vertex of the arrangement until it either hits another segment or extends to infinity.

The extension to higher dimensions is defined recursively. For details of the extension to three dimensions, see [CEG+90] for the case of spheres, and [dBGH96] for the case of triangles. The extension to higher dimensions is given in [CEGS91]. Table 21.3.1 summarizes the bounds on the maximum combinatorial complexity of the vertical decomposition for several types of arrangements and substructures. Certain assumptions that curves and surfaces are "well-behaved" are not detailed.

TABLE 21.3.1 Combinatorial bounds on the maximum complexity of the vertical decomposition of n objects.

OBJECTS	BOUND	NOTES
Curves in $\mathbb{R}^2$	$\Theta(K)$	K is the complexity of $\mathcal{A}$
Surfaces in $\mathbb{R}^d$	$O(n^{2d-3}\beta(n))$	[CEGS91], [SA95]
		$\beta(n) = \lambda_q(n)/n$, q a constant
Triangles in $\mathbb{R}^3$	$\Theta(n^3)$	[dBGH96]
Triangles in $\mathbb{R}^3$	$O(n^2\alpha(n)\log n + K)$	K is the complexity of $\mathcal{A}$ [Tag95]
Surfaces in $\mathbb{R}^3$, single cell	$O(n^{2+\epsilon})$	[SS96]
Surfaces in $\mathbb{R}^3$, $(\leq k)$-level	$O(n^{2+\epsilon}k)$	for this and more refined bounds see [AES95]
Hyperplanes in $\mathbb{R}^4$	$O(n^4 \log n)$	[GHMS95]

THE SLICING AND CHOPPING THEOREMS

Aronov and Sharir devised alternative decomposition methods for arrangements of simplices [AS90], [AS94]. These are more involved than the decompositions described above and we omit their descriptions here. These methods were instrumental in obtaining improved combinatorial bounds and efficient algorithms for arrangements of simplices.

STRUCTURES FOR POINT LOCATION AND RAY SHOOTING

To access certain cells of an arrangement without traversing the entire arrangement, we need more elaborate structures than those described above. See Chapters 31 and 32 for details.

OPEN PROBLEMS

1. Obtain an improved combinatorial bound on the complexity of the vertical decomposition of arrangements of surfaces in four and higher dimensions. Such a result would have a wide-ranging effect on other combinatorial bounds, on algorithms, and on a variety of applications of arrangements.
2. Consider the decomposition in Figure 21.3.2. Original edges of the arrangement are in fact *two-sided*. They have different sets of endpoints of vertical extensions on either side. This raises the question of how to maintain these sets so that moving from one side of an edge to the other will be carried out efficiently (while traversing the arrangement). In the two-dimensional case there is an elegant solution, where some of the vertical extensions are further extended so that along an edge of one face there are at most two neighboring faces on the other side [Cha86]. However, this idea does not seem to extend directly to higher dimensions.

21.4 ALGORITHMS FOR ARRANGEMENTS

This section covers constructing an arrangement: producing a representation of an arrangement in one of the forms described in the previous section (or in a similar form). We distinguish between algorithms for the construction of the entire arrangement (surveyed in this section), and algorithms for constructing substructures of an arrangement (in the next section). We start with deterministic algorithms and then describe randomized ones.

MODEL OF COMPUTATION

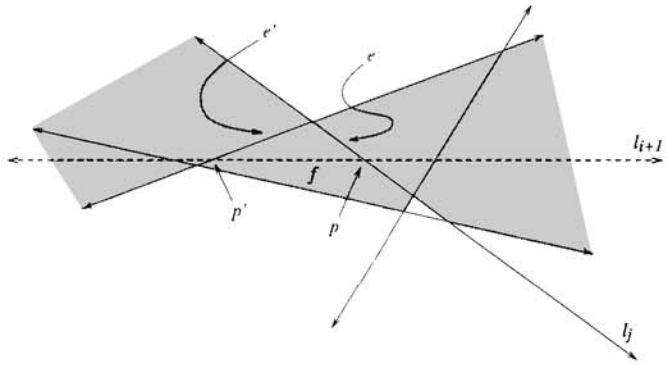
We assume the standard model in computational geometry: infinite precision real arithmetic [PS85]. For algorithms computing arrangements of curves or surfaces, we further assume that certain operations on a small number of curves or surfaces

each take unit time. For algebraic curves or surfaces, the unit cost assumption for these operations is theoretically justified by results on the solution of sets of polynomial equations; see Chapter 29.

21.4.1 DETERMINISTIC ALGORITHMS

Incremental construction. The incremental algorithm proceeds by adding one object after the other to the arrangement while maintaining (a representation of) the arrangement of the objects added so far. This approach yields an optimal-time algorithm for arrangements of hyperplanes. The analysis of the running time is based on the zone result [ESS93] (Section 21.2). We describe it next for a collection $\mathcal{L} = \{l_1, \ldots, l_n\}$ of n lines in the plane, assuming that the arrangement $\mathcal{A}(\mathcal{L})$ is simple.

FIGURE 21.4.1
Adding the line l_{i+1} to the arrangement $\mathcal{A}(\mathcal{L}_i)$.
The shaded region is the zone of l_{i+1} in the arrangement of the other four lines.



Let $\mathcal{L}_i$ denote the set $\{l_1, \ldots, l_i\}$. At stage $i+1$ we add l_{i+1} to the arrangement $\mathcal{A}(\mathcal{L}_i)$. We maintain the quad-edge representation for $\mathcal{A}(\mathcal{L}_i)$, so that in addition to the incidence information, we also have the order of edges along the boundary of each face. The addition of l_{i+1} is carried out in two steps: (i) we find a point p of intersection between l_{i+1} and an edge of $\mathcal{A}(\mathcal{L}_i)$ and split that edge into two, and (ii) we walk along l_{i+1} from p to the left (assuming l_{i+1} is not vertical) updating $\mathcal{A}(\mathcal{L}_i)$ as we go; we then walk along l_{i+1} from p to the right completing the construction of $\mathcal{A}(\mathcal{L}_{i+1})$. See Figure 21.4.1.

Finding an edge of $\mathcal{A}(\mathcal{L}_i)$ that l_{i+1} intersects can be done in $O(i)$ time by choosing one line l_j from $\mathcal{L}_i$ and checking all the edges of $\mathcal{A}(\mathcal{L}_i)$ that lie on l_j for intersection with l_{i+1}. This intersection point p lies on an edge e that borders two faces of $\mathcal{A}(\mathcal{L}_i)$. We split e into two edges at p. Next, consider the face f intersected by the part of l_{i+1} to the left of p. Using the order information, we walk along the edges of f away from p and we check for another intersection p' of l_{i+1} with an edge e' on the boundary of f. At the intersection we split e' into two edges, we add an edge to the arrangement for the portion $\overline{pp'}$ of l_{i+1}, and we move to the face on the other (left) side of e'. Once we are done with the faces of $\mathcal{A}(\mathcal{L}_i)$ crossed

by l_{i+1} to the left of p, we go back to p and walk to the other side. This way we visit all the faces of the zone of l_{i+1} in $\mathcal{A}(\mathcal{L}_i)$, as well as some of its edges. The amount of time spent is proportional to the number of edges we visit, and hence bounded by the complexity of the zone. The space required for the algorithm is the space to maintain the quad-edge structure. The same approach extends to higher dimensions. For details see [Ede87, Chapter 7] (note that the algorithm as described in [Ede87] uses the incidence graph for maintaining the arrangement).

THEOREM 21.4.1

If $\mathcal{H}$ is a set of n hyperplanes in $\mathbb{R}^d$ such that $\mathcal{A}(\mathcal{H})$ is a simple arrangement, then $\mathcal{A}(\mathcal{H})$ can be constructed in $\Theta(n^d)$ time and space.

The time and space required by the algorithm are clearly optimal. However, it turns out that for arrangements of lines one can do better in terms of *working space*. This is explained below in the subsection *topological sweep*.

The incremental approach can be applied to constructing arrangements of curves, using the vertical decomposition of the arrangement [EGP+92]:

THEOREM 21.4.2

Let $\mathcal{C}$ be a set of n Jordan arcs as defined in Section 21.1. *The arrangement $\mathcal{A}(\mathcal{C})$ can be constructed in $O(n\lambda_{s+2}(n))$ time using $O(n^2)$ space.*

Sweeping over the arrangement. The sweep paradigm, a fundamental paradigm in computational geometry, is also applicable to constructing arrangements. For planar arrangements, its worst-case running time is slightly inferior to that of the incremental construction described above. It is, however, output sensitive.

THEOREM 21.4.3

Let $\mathcal{C}$ be a set of n Jordan arcs as defined in Section 21.1. *The arrangement $\mathcal{A}(\mathcal{C})$ can be constructed in $O((n+k)\log n)$ time and $O(n+k)$ space, where k is the number of intersection points in the arrangement.*

One can similarly sweep a plane over an arrangement of surfaces in $\mathbb{R}^3$. There is an output-sensitive algorithm for constructing the vertical decomposition of an arrangement of n surfaces that runs in time $O(n\lambda_q(n)\log n + V\log n)$, where V is the combinatorial complexity of the vertical decomposition ($\lambda_q(n)$ is a near-linear function related to Davenport-Schinzel sequences), and q is a constant that depends on the degree of the surface patches and their boundaries. For details see [dBGH96].

Topological sweep. Edelsbrunner and Guibas [EG89] devised an algorithm for constructing an arrangement of lines that requires only linear working storage and still runs in optimal $O(n^2)$ time. Instead of sweeping the arrangement with a straight line, they sweep it with a pseudoline that serves as a "topological wavefront."

21.4.2 RANDOMIZED ALGORITHMS

Most randomized algorithms for arrangements follow one of two paradigms: (i) incremental construction or (ii) divide-and-conquer using random sampling. The randomization in these algorithms is in choices made by the algorithm; for example,

the order in which the objects are handled in an incremental construction. In the expected performance bounds, the expectation is with respect to the random choices made by the algorithm. We do not make any assumptions about the objects' distribution in space. See also Chapter 34.

In constructing a full arrangement, these two paradigms are rather straightforward to apply. Most of these algorithms use an efficient decomposition as discussed in Section 21.3. At the end of this section we remark on the *derandomization* techniques used to turn these algorithms into deterministic algorithms.

Incremental construction. Here the randomization is in the order that the objects defining the arrangement are inserted. For the construction of an arrangement of curves, the algorithm is similar to the deterministic construction mentioned above.

THEOREM 21.4.4

Let $\mathcal{C}$ be a set of n Jordan arcs as defined in Section 21.1. *The arrangement $\mathcal{A}(\mathcal{C})$ can be constructed by a randomized incremental algorithm in $O(n \log n + k)$ expected time and $O(n + k)$ expected space, where k is the number of intersection points in the arrangement.*

Divide-and-conquer by random sampling. For a set $\mathcal{V}$ of n objects in $\mathbb{R}^d$ the paradigm is: choose a subset $\mathcal{R}$ of the objects at random, construct the arrangement $\mathcal{A}(\mathcal{R})$, decompose it further into constant complexity components (using, for example, one of the methods described in Section 21.3), and recursively construct the portion of the arrangement in each of the resulting components. Then glue all the substructures together into the full arrangement. The theory of random sampling is then used to show that with high probability the size of each subproblem is considerably smaller than that of the original problem, and thus efficient resource bounds can be proved.

The result stated in the following theorem is obtained by applying this technique to arrangements of algebraic surfaces and it is based on the vertical decomposition of the arrangement.

THEOREM 21.4.5

Given a collection $\mathcal{S}$ of n algebraic surfaces in $\mathbb{R}^d$ as defined in Section 21.1, then for any $\epsilon > 0$, a data structure of size $O(n^{2d-3+\epsilon})$ for the arrangement $\mathcal{A}(\mathcal{S})$ can be constructed in $O(n^{2d-3+\epsilon})$ time, so that a point-location query can be answered in $O(\log n)$ time. In these bounds the constant of proportionality depends on ϵ, the dimension d, and the maximum algebraic degree of the surfaces and their boundaries.

If only traversal of the entire arrangement is needed, it is plausible that a simpler structure such as the incidence graph could be constructed using less time and storage space, probably close to $O(n^d)$ for both. See [Can93],[BPR96] for algebro-geometric methods.

Derandomization. Techniques have been proposed to derandomize many of the randomized geometric algorithms, often without increase in their asymptotic running time; see Chapter 34. However, in most cases the randomized versions are conceptually much simpler and hence may be better candidates for efficient implementation.

21.4.3 OTHER ALGORITHMIC ISSUES

For algebro-geometric tools, see Chapter 29. See Chapter 35 for a discussion of precision and degeneracies. Parallel algorithms are discussed in Chapter 36.

21.5 CONSTRUCTING SUBSTRUCTURES

ENVELOPE AND SINGLE CELL IN ARRANGEMENTS OF HYPERPLANES

Any single cell in an arrangement of hyperplanes is a convex polyhedron. Also, the lower envelope of a collection of hyperplanes is the boundary of the cell that lies below all the hyperplanes. Therefore computing a single cell or an envelope in such an arrangement is the same as computing the polyhedron which is the intersection of the appropriate halfspaces. See Chapter 19.

Using linearization [AM94], we can solve these problems for arrangements of spheres in $\mathbb{R}^d$. We first transform the spheres into hyperplanes in $\mathbb{R}^{d+1}$, and then solve the corresponding problems in $\mathbb{R}^{d+1}$.

LOWER ENVELOPE

The lower envelope of a collection of well-behaved curves can be computed by a simple divide-and-conquer algorithm that runs in time $O(\lambda_{s+2}(n)\log n)$ and requires $O(\lambda_{s+2}(n))$ storage. Hershberger [Her89] devised an improved algorithm that runs in time $O(\lambda_{s+1}(n)\log n)$. In 3-space, Agarwal et al. [ASS95] showed that a simple divide-and-conquer scheme can be used to compute the envelope of n surfaces in time $O(n^{2+\epsilon})$. This is an application of the bound on the complexity of the overlay of envelopes cited in Section 21.2. Boissonnat and Dobrindt give a randomized incremental algorithm for computing the envelope [BD96]. There are efficient algorithms for computing the envelope of $(d-1)$-simplices in $\mathbb{R}^d$, and an efficient data structure for point location in the minimization diagram of surfaces in $\mathbb{R}^4$.

SINGLE CELL AND ZONE

All the results cited below for a single cell hold for the zone problem as well (see the remark in Section 21.2 on the connection between the problems).

Computing a single face in an arrangement of n Jordan arcs as defined in Section 21.1 can be accomplished in worst-case near-optimal time: deterministically in $O(\lambda_{s+2}(n)\log^2 n)$ time, and using randomization in $O(\lambda_{s+2}(n)\log n)$ time [SA95]. There are also algorithms for special cases of planar arrangements such as arrangements of rays.

In three dimensions, Schwarzkopf and Sharir [SS96] give an algorithm with running time $O(n^{2+\epsilon})$ for any $\epsilon > 0$ to compute a single cell in an arrangement of n well-behaved surfaces. Algorithms with improved running time to compute a single cell in three-dimensional arrangements are known for arrangements of surfaces in-

duced by certain motion planning problems [Hal92], [Hal94], and for arrangements of triangles [dBDS95].

LEVELS

In an arrangement of n lines in the plane, the k-level can be computed in $O(n \log f + f \log^2 n)$ time [EW86],[Cha95], where f is the combinatorial complexity of the k-level, and the $(\leq k)$-level in worst-case optimal time $O(n \log n + kn)$ [ERvK93]. Algorithms for computing the $(\leq k)$-level in an arrangement of Jordan arcs and the $(\leq k)$-level in an arrangement of planes in $\mathbb{R}^3$ are described in [AdBMS94]. For computing the k-level in an arrangement of hyperplanes in $\mathbb{R}^d$ see [Cha95].

MANY CELLS

There are efficient algorithms (deterministic and randomized) for computing a set of selected faces in arrangements of lines or segments in the plane. These algorithms are nearly worst-case optimal [AMS94a]. Algorithms for arrangements of planes are described in [EGS90], and for arrangements of triangles in 3-space in [AS90].

OPEN PROBLEMS

Devise efficient algorithms for computing:

1. The lower envelope of an arrangement of surfaces in five and higher dimensions; for an algorithm in four dimensions see [AAS94].
2. A single cell in an arrangement of surfaces in four and higher dimensions; for an algorithm in three dimensions see [SS96].

21.6 SPARSE ARRANGEMENTS

So far we have discussed arrangements of n objects in $\mathbb{R}^d$ where each object has constant descriptive complexity and the total complexity of the entire arrangement can be $\Omega(n^d)$ in the worst case. In many situations arrangements do not achieve this worst-case complexity, or there are additional parameters that control the complexity of the arrangement. In this section we survey several such situations.

Let $\mathcal{C}$ be a collection of n Jordan arcs, where each pair of arcs in $\mathcal{C}$ intersects at most a constant number of times, and with the additional condition that any vertical line intersects at most k of the curves in $\mathcal{C}$. In this case the maximum combinatorial complexity of the arrangement $\mathcal{A}(\mathcal{C})$ is $\Theta(nk)$. For an application of this result and for more results on arrangements with low ***vertical stabbing number*** (the number of objects stabbed by any vertical line) see [dBHOvK].

A general way to take advantage of reduced complexity of an arrangement is to construct the arrangement using an output-sensitive algorithm. However, by understanding the source of the reduced complexity it may be possible to devise algorithms that perform better than general purpose output-sensitive algorithms. In several cases this has indeed been achieved. The collection of atom spheres in

the geometric model of molecules exhibits sparseness properties that have led to improved combinatorial bounds and relatively simple algorithms. These algorithms have been implemented and perform well in practice [HO94], [FHK+96]. Another area where results of this nature have been obtained is robot motion planning among *fat obstacles*; see Section 40.3.

ARRANGEMENTS OF CONVEX POLYTOPES

Consider the subdivision of 3-space induced by k convex polytopes with a total of n vertices. To bound the complexity of this arrangement we can regard this as an arrangement of $O(n)$ triangles in 3-space, implying an upper bound $O(n^3)$. However, Aronov et al. [ABE91] showed that the complexity of such an arrangement is $O(nk^2)$. More generally they showed that the complexity of an arrangement of k convex polytopes in $\mathbb{R}^d$ with a total of n facets is $\Theta(n^{\lfloor \frac{d}{2} \rfloor} k^{\lceil \frac{d}{2} \rceil})$.

A useful substructure in an arrangement of convex polytopes is the collection of ***maximally covered cells***, namely cells of the arrangement that are covered by more polytopes than any other cell in their immediate neighborhood [GHH+95]. The ability to access these cells efficiently has led to an efficient and practical algorithm to test whether an object consisting of polyhedral parts is interlocked (i.e., cannot be taken apart with two hands).

21.7 RELATION TO OTHER STRUCTURES

Arrangements relate to a variety of additional structures. Since the machinery for analyzing and computing arrangements is rather well developed, problems on related structures are often solved by first constructing (or reasoning about) the corresponding arrangement.

Using *duality* one can transform a set (or *configuration*) of points in $\mathbb{R}^d$ (the primal space) into a set of hyperplanes in $\mathbb{R}^d$ (the dual space) and vice versa. Different duality transforms are advantageous in different situations [O'R94]. Edelsbrunner [Ede87, Chapter 12] describes a collection of problems stated for point configurations and solved by operating on their corresponding dual arrangements. See also Chapter 1.

Plücker coordinates are a tool that enables one to treat k-flats in $\mathbb{R}^d$ as points or hyperplanes in a possibly different dimensional space. This has been taken advantage of in the study of families of lines in 3-space—see Chapter 32.

Lower envelopes (or more generally the k-level in arrangements) relate to Voronoi diagrams—see Chapter 20.

For the connection of arrangements to polytopes and zonotopes see [Ede87] and Section 13.1.4 of this Handbook.

21.8 APPLICATIONS

A typical application of arrangements is for solving a problem on related structures. We first transform the original structure (e.g., a point configuration) into

an arrangement and then solve the problem on the resulting arrangement. See Section 21.7 above and Chapters 1, 20, and 32.

Another strand of applications consists of the "robotic" or "physical world" applications [HS95b]. In these problems a continuous space is decomposed into a finite number of cells so that in each cell a certain invariant is maintained. Here, arrangements are used to discretize a continuous space without giving up the completeness or exactness of the solution. An example of an application of this kind solves the following problem: Given a convex polyhedron in 3-space, determine how many combinatorially distinct orthographic and perspective views it induces; see Table 25.6.3. The answer is given using an arrangement of circles on the sphere (for orthographic views) and an arrangement of planes in 3-space (for perspective views) [BD90].

Arrangements have been applied to solving problems in robot motion planning (Chapter 40) and several of its variants (Chapter 41). For example, the most efficient algorithm known for computing a collision-free path for a polygonal robot moving by translation and rotation among polygonal obstacles in the plane is based on computing a single connected component in an arrangement of surfaces in 3-space.

As mentioned earlier, arrangements on spheres are prevalent in applications. Aside from vision applications, they also occur in: computer-assisted radio-surgery [SAL93], molecular modeling, assembly planning (Section 41.3), and more.

Arrangements have been used to solve problems in many other areas. More applications can be found in the sources cited below and in several other chapters in this book.

21.9 SOURCES AND RELATED MATERIAL

FURTHER READING

The study of arrangements through the early 1970's is covered by Grünbaum in [Grü67, Chapter 18], [Grü71], and [Grü72]. See also the monograph by Zaslavsky [Zas75].

In this chapter we have concentrated on more recent results. Details of many of these results can be found in the following books. The book by Edelsbrunner [Ede87] takes the view of "arrangements of hyperplanes" as a unifying theme for a large part of discrete and computational geometry until 1987. Sharir and Agarwal's book [SA95] is an extensive report on recent results for arrangements of curves and surfaces.

Chapters dedicated to arrangements of hyperplanes in recently published books: Mulmuley emphasizes randomized algorithms [Mul94], O'Rourke discusses basic combinatorics, relations to other structures and applications [O'R94], and Pach and Agarwal [PA95] discuss problems involving arrangements in discrete geometry.

RELATED CHAPTERS

REFERENCES

[AA92] P.K. Agarwal and B. Aronov. Counting facets and incidences. *Discrete Comput. Geom.*, 7:359–369, 1992.

[AAS94] P.K. Agarwal, B. Aronov, and M. Sharir. Computing envelopes in four dimensions with applications. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 348–358, 1994.

[ABE91] B. Aronov, M. Bern, and D. Eppstein. The complexity of an arrangements of polytopes, 1991. Manuscript.

[AdBMS94] P.K. Agarwal, M. de Berg, J. Matoušek, and O. Schwarzkopf. Constructing levels in arrangements and higher order Voronoi diagrams. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 67–75, 1994.

[AES95] P.K. Agarwal, A. Efrat, and M. Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 39–50, 1995.

[AG86] N. Alon and E. Győri. The number of small semispaces of a finite set of points in the plane. *J. Combin. Theory Ser. A*, 41:154–157, 1986.

[AM94] P.K. Agarwal and J. Matoušek. On range searching with semialgebraic sets. *Discrete Comput. Geom.*, 11:393–418, 1994.

[AMS94a] P.K. Agarwal, J. Matoušek, and O. Schwarzkopf. Computing many faces in arrangements of lines and segments. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 76–84, 1994.

[AMS94b] B. Aronov, J. Matoušek, and M. Sharir. On the sum of squares of cell complexities in hyperplane arrangements. *J. Combin. Theory Ser. A*, 65:311–321, 1994.

[APS93] B. Aronov, M. Pellegrini, and M. Sharir. On the zone of a surface in a hyperplane arrangement. *Discrete Comput. Geom.*, 9:177–186, 1993.

[AS90] B. Aronov and M. Sharir. Triangles in space or building (and analyzing) castles in the air. *Combinatorica*, 10:137–173, 1990.

[AS94] B. Aronov and M. Sharir. Castles in the air revisited. *Discrete Comput. Geom.*, 12:119–150, 1994.

[ASS95] P.K. Agarwal, O. Schwarzkopf, and M. Sharir. The overlay of lower envelopes in three dimensions and its applications. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 182–189, 1995.

[BD90] K.W. Bowyer and C.R. Dyer. Aspect graphs: An introduction and survey of recent results. *Internat. J. Imaging Syst. Tech.*, 2:315–328, 1990.

[BD96] J.-D. Boissonnat and K.T.G. Dobrindt. On-line construction of the upper envelope of triangles and surface patches in three dimensions. *Comput. Geom. Theory Appl.*, 5:303–320, 1996.

[BLS+93] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G. Ziegler. *Oriented Matroids*, volume 46 of *Encyclopedia Math. Appl.*. Cambridge University Press, Cambridge, 1993.

[BPR96] S. Basu, R. Pollack, and M.-F. Roy. Computing roadmaps of semi-algebraic sets. In *Proc. 28th Annu. ACM Sympos. Theory Comput.*, pages 168–173. 1996.

[Bri93] E. Brisson. Representing geometric structures in d dimensions: Topology and order. *Discrete Comput. Geom.*, 9:387–426, 1993.

[Can93] J. Canny. Computing roadmaps in general semialgebraic sets. *Comput. J.*, 36:504–514, 1993.

[CEG+90] K. Clarkson, H. Edelsbrunner, L. Guibas, M. Sharir, and E. Welzl. Combinatorial complexity bounds for arrangements of curves and spheres. *Discrete Comput. Geom.*, 5:99–160, 1990.

[CEGS91] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. A singly-exponential stratification scheme for real semi-algebraic varieties and its applications. *Theoret. Comput. Sci.*, 84:77–105, 1991. An improved bound appears in the proceedings version: *Proc. ICALP 1989*, volume 372 of *Lecture Notes in Comput. Sci.*, Springer-Verlag, New York, 1989.

[Cha86] B. Chazelle. Filtering search: a new approach to query-answering. *SIAM J. Comput.*, 15:703–724, 1986.

[Cha95] T.M. Chan. Output-sensitive results on convex hulls, extreme points, and related problems. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 10–19, 1995.

[Cla88] K.L. Clarkson. A randomized algorithm for closest-point queries. *SIAM J. Comput.*, 17:830–847, 1988.

[CS89] K.L. Clarkson and P.W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.

[dBDS95] M. de Berg, K. Dobrindt, and O. Schwarzkopf. On lazy randomized incremental construction. *Discrete Comput. Geom.*, 14:261–286, 1995.

[dBGH96] M. de Berg, L.J. Guibas, and D. Halperin. Vertical decompositions for triangles in 3-space. *Discrete Comput. Geom.*, 15:35–61, 1996.

[dBHOvK] M. de Berg, D. Halperin, M.H. Overmars, and M. van Kreveld. Sparse arrangements and the number of views of polyhedral scenes. *Internat. J. Comput. Geom. Appl.*, to appear.

[DL89] D.P. Dobkin and M.J. Laszlo. Primitives for the manipulation of three-dimensional subdivisions. *Algorithmica*, 4:3–32, 1989.

[Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer-Verlag, Heidelberg, 1987.

[Ede89] H. Edelsbrunner. The upper envelope of piecewise linear functions: Tight complexity bounds in higher dimensions. *Discrete Comput. Geom.*, 4:337–343, 1989.

[EG89] H. Edelsbrunner and L.J. Guibas. Topologically sweeping an arrangement. *J. Comput. Syst. Sci.*, 38:165–194, 1989. Corrigendum in 42:249–251, 1991.

[EGP+92] H. Edelsbrunner, L. Guibas, J. Pach, R. Pollack, R. Seidel, and M. Sharir. Arrangements of curves in the plane: Topology, combinatorics, and algorithms. *Theoret. Comput. Sci.*, 92:319–336, 1992.

[EGS90] H. Edelsbrunner, L. Guibas, and M. Sharir. The complexity of many cells in arrangements of planes and related problems. *Discrete Comput. Geom.*, 5:197–216, 1990.

[ERvK93] H. Everett, J.-M. Robert, and M. van Kreveld. An optimal algorithm for the $(\leq k)$-levels, with applications to separation and transversal problems. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 38–46, 1993.

[ESS93] H. Edelsbrunner, R. Seidel, and M. Sharir. On the zone theorem for hyperplane arrangements. *SIAM J. Comput.*, 22:418–429, 1993.

[EW85] H. Edelsbrunner and E. Welzl. On the number of line separations of a finite set in the plane. *J. Combin. Theory Ser. A*, pages 15–29, 1985.

[EW86] H. Edelsbrunner and E. Welzl. Constructing belts in two-dimensional arrangements with applications. *SIAM J. Comput.*, 15:271–284, 1986.

[FHK+96] P.W. Finn, D. Halperin, L. Kavraki, J.-C. Latombe, R. Motwani, C. Shelton, and S. Venkatsubramanian. Geometric manipulation of flexible ligands. In *Proc. 1st ACM Workshop Appl. Comput. Geom.*, volume 1148 of *Lecture Notes in Comput. Sci.*, pages 67–78. Springer-Verlag, Berlin, 1996.

[GHH+95] L.J. Guibas, D. Halperin, H. Hirukawa, J.-C. Latombe, and R.H. Wilson. A simple and efficient procedure for assembly partitioning under infinitesimal motions. In *Proc. 12th IEEE Internat. Conf. Robot. Autom.*, pages 2553–2560, 1995.

[GHMS95] L.J. Guibas, D. Halperin, J. Matoušek, and M. Sharir. On vertical decomposition of arrangements of hyperplanes in four dimensions. *Discrete Comput. Geom.*, 14:113–122, 1995.

[Grü67] B. Grünbaum. *Convex Polytopes.* Wiley, New York, 1967.

[Grü71] B. Grünbaum. Arrangements of hyperplanes. *Congr. Numer.*, 3:41–106, 1971.

[Grü72] B. Grünbaum. *Arrangements and Spreads.* Volume 10 of *CBMS Regional Conf. Ser. in Math.*, Amer. Math. Soc., Providence, 1972.

[GS85] L.J. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Trans. Graph.*, 4:74–123, 1985.

[GSS89] L.J. Guibas, M. Sharir, and S. Sifrony. On the general motion planning problem with two degrees of freedom. *Discrete Comput. Geom.*, 4:491–521, 1989.

[Hal92] D. Halperin. *Algorithmic Motion Planning via Arrangements of Curves and of Surfaces.* Ph.D. thesis, Computer Science Dept., Tel-Aviv Univ., 1992.

[Hal94] D. Halperin. On the complexity of a single cell in certain arrangements of surfaces related to motion planning. *Discrete Comput. Geom.*, 11:1–34, 1994.

[Her89] J. Hershberger. Finding the upper envelope of n line segments in $O(n \log n)$ time. *Inform. Process. Lett.*, 33:169–174, 1989.

[HO94] D. Halperin and M.H. Overmars. Spheres, molecules, and hidden surface removal. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 113–122, 1994.

[HS94] D. Halperin and M. Sharir. New bounds for lower envelopes in three dimensions, with applications to visibility in terrains. *Discrete Comput. Geom.*, 12:313–326, 1994.

[HS95a] D. Halperin and M. Sharir. Almost tight upper bounds for the single cell and zone problems in three dimensions. *Discrete Comput. Geom.*, 14:385–410, 1995.

[HS95b] D. Halperin and M. Sharir. Arrangements and their applications in robotics: Recent developments. In K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson, editors, *Algorithmic Foundations of Robotics*, pages 495–511. A K Peters, Boston, 1995.

[McM70] P. McMullen. The maximal number of faces of a convex polytope. *Mathematica*, 17:179–184, 1970.

[Mul94] K. Mulmuley. *Computational Geometry: An Introduction through Randomized Algorithms.* Prentice Hall, Englewood Cliffs, 1994.

[O'R94] J. O'Rourke. *Computational Geometry in C.* Cambridge University Press, 1994.

[PA95] J. Pach and P.K. Agarwal. *Combinatorial Geometry.* Wiley, New York, 1995.

[PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, 1985.

[PSS92] J. Pach, W. Steiger, and E. Szemerédi. An upper bound on the number of planar k-sets. *Discrete Comput. Geom.*, 7:109–123, 1992.

[SA95] M. Sharir and P.K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications.* Cambridge University Press, New York, 1995.

[SAL93] A. Schweikard, J.E. Adler, and J.-C. Latombe. Motion planning in stereotaxic radiosurgery. In *Proc. 10th IEEE Internat. Conf. Robot. Autom.*, pages 764–774, 1993.

[Sei94] R. Seidel. The nature and meaning of perturbations in geometric computing. In *Proc. 11th Sympos. Theoret. Aspects Comput. Sci.*, volume 775 of *Lecture Notes in Comput. Sci.*, pages 3–17. Springer-Verlag, Berlin, 1994.

[Sha91] M. Sharir. On k-sets in arrangements of curves and surfaces. *Discrete Comput. Geom.*, 6:593–613, 1991.

[Sha94] M. Sharir. Almost tight upper bounds for lower envelopes in higher dimensions. *Discrete Comput. Geom.*, 12:327–345, 1994.

[SS96] O. Schwarzkopf and M. Sharir. Vertical decomposition of a single cell in a three-dimensional arrangement of surfaces and its applications. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 20–29, 1996.

[Szé] L.A. Székely. Crossing numbers and hard Erdős problems in discrete geometry. *Combin. Probab. Comput.*, to appear.

[Tag95] B. Tagansky. A new technique for analyzing substructures in arrangements. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 200–210, 1995.

[Zas75] T. Zaslavsky. *Facing up to Arrangements: Face-Count Formulas for Partitions of Space by Hyperplanes*, volume 1:154 of *Mem. Amer. Math. Soc.* Amer. Math. Soc., Providence, 1975.

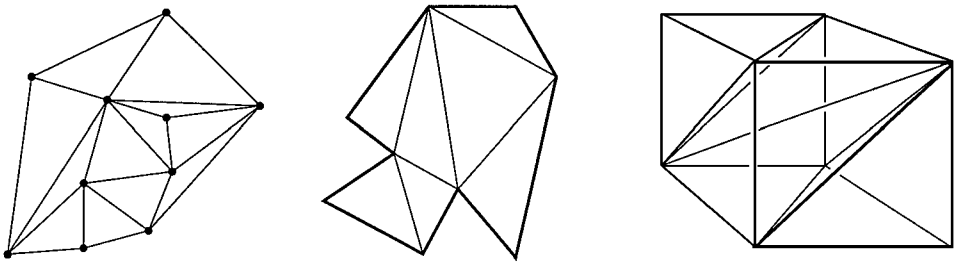
22 TRIANGULATIONS

Marshall Bern

INTRODUCTION

A triangulation is a partition of a geometric domain, such as a point set, polygon, or polyhedron, into simplices that meet only at shared faces. (For point sets, the partition stops at the convex hull.) The first four sections of this chapter discuss two-dimensional triangulations: Delaunay triangulation of point sets (Section 22.1); triangulations of polygons, including constrained Delaunay triangulation (Section 22.2); other optimal triangulations (Section 22.3); and mesh generation (Section 22.4). The last two sections treat polyhedra in $\mathbb{R}^3$ (Section 22.5), and point sets in $\mathbb{R}^d$ (Section 22.6).

FIGURE 22.0.1
Triangulations of a point set, a simple polygon, and a polyhedron.



22.1 DELAUNAY TRIANGULATION

The Delaunay triangulation is the most famous and useful triangulation of a point set. Chapter 20 discusses this construction in conjunction with the Voronoi diagram.

GLOSSARY

Empty circle: No input points in the interior.

Delaunay triangulation: Triangles have empty circumcircles.

Completion: Four or more cocircular points must be further triangulated.

Edge flipping: Local improvement algorithm for Delaunay triangulation.

BASIC FACTS

Let $S = \{s_1, s_2, \ldots, s_n\}$ be a set of points in the Euclidean plane $\mathbb{R}^2$. The Delaunay triangulation (DT) is defined by the ***empty circle condition***: a triangle $s_i s_j s_k$ appears in the DT if and only if its circumcircle neither encloses nor passes through any other points of S.

The DT always includes the convex hull of S. If no four points of S are cocircular, the Delaunay triangulation is indeed a triangulation of S. If four or more points are cocircular, there may be faces with more than three sides, which can be triangulated to ***complete*** the triangulation of S. The DT is the planar dual of the Voronoi diagram, meaning that an edge $s_i s_j$ appears in the DT if and only if the Voronoi cells of s_i and s_j share a boundary edge.

There is a connection between a Delaunay triangulation in $\mathbb{R}^2$ and a convex polytope in $\mathbb{R}^3$. If we ***lift*** S onto the paraboloid with equation $z = x^2 + y^2$ by mapping $s_i = (x_i, y_i)$ to $(x_i, y_i, x_i^2 + y_i^2)$, then the DT turns out to be the projection of the lower convex hull of the lifted points. See Figure 20.1.2.

ALGORITHMS

There are a number of practical planar DT algorithms [For95], including edge flipping, incremental construction, sweep-line, and divide-and-conquer. We describe only the edge flipping algorithm, even though its worst-case running time of $O(n^2)$ is not optimal, because it is most relevant to our subsequent discussion.

The edge flipping algorithm starts from any triangulation of S and then locally optimizes each edge. Let e be an internal (non-convex-hull) edge and Q_e be the triangulated quadrilateral formed by the triangles sharing e. Q_e is *reversed* if the two angles without the diagonal sum to more than 180°, or equivalently, if each triangle circumcircle contains the opposite vertex. If Q_e is reversed, we "flip" it by exchanging e for the other diagonal.

compute an initial triangulation of S
place all internal edges into a queue
while the queue is not empty **do**
    remove the first edge e
    if quadrilateral Q_e is reversed **then** flip it **fi**
    add outside edges of Q_e to the queue **od**

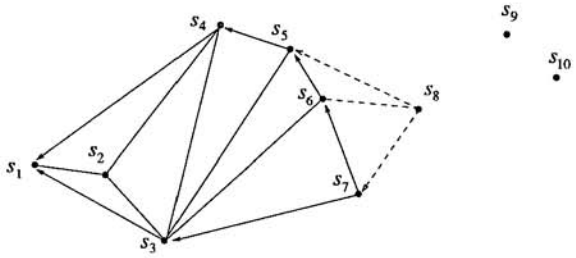


FIGURE 22.1.1
A generic step in computing the initial triangulation.

An initial triangulation can be computed by a sweep-line algorithm, as shown in Figure 22.1.1. This algorithm adds the points of S by x-coordinate order. Upon each addition, the algorithm walks around the convex hull of the already-added points, adding edges until the slope reverses.

The following theorem guarantees the success of edge flipping: a triangulation in which no quadrilateral is reversed must be a completion of the DT. This theorem can be proved using the lifting map; a reversed quadrilateral lifts to a reflex edge, and a surface without reflex edges must be the lower convex hull.

OPTIMALITY PROPERTIES

Certain quality measures [BE95] are improved by flipping a reversed quadrilateral. For example, the minimum angle in a triangle of Q_e must increase. Hence, a triangulation that maximizes the minimum angle cannot have a reversed quadrilateral, implying that it is a completion of the DT. Some completion of the DT:

- minimizes the maximum radius of a circumcircle;
- maximizes the minimum angle (in fact, lexicographically maximizes the angles from smallest to largest);
- minimizes the maximum radius of an enclosing circle;
- maximizes the sum of inscribed circle radii;
- minimizes the "potential energy" of a piecewise-linear interpolating surface; and
- minimizes the surface area of a piecewise-linear interpolating surface for elevations scaled sufficiently small.

Two additional properties of the DT: Delaunay simplices are acyclically ordered by distance from any fixed reference point, and the distance along edges of the DT between any pair of vertices is at most a constant (at most 2.42) times the Euclidean distance between them.

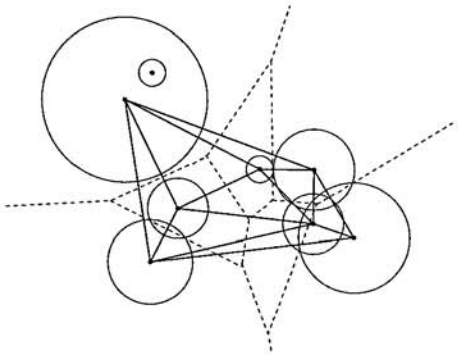


FIGURE 22.1.2
Power diagram and regular triangulation.

REGULAR TRIANGULATIONS

Delaunay triangulations and Voronoi diagrams may be defined for various distance measures (Section 20.3); here we mention one generalization that retains most of

the rich mathematical structure. Suppose each point $s_i = (x_i, y_i)$ in S has a weight w_i. The ***regular triangulation*** of S is the projection of the lower convex hull of the points $(x_i, y_i, x_i^2 + y_i^2 - w_i)$. In general, the regular triangulation is a graph on a subset of the sites S, but in the special case that all weights are zero, the regular triangulation is exactly the DT. See Section 14.3.

The planar dual of the regular triangulation is the ***power diagram***, a Voronoi diagram in which the distance to s_i is the square of the Euclidean distance minus w_i. We can regard the sites in a power diagram as circles, with the radius of site i being $\sqrt{w_i}$. See Figure 22.1.2.

22.2 TRIANGULATIONS OF POLYGONS

We now discuss triangulations of more complicated inputs: polygons and planar straight-line graphs. We start with the problem of simply computing any triangulation and then progress to constrained Delaunay triangulation.

GLOSSARY

Simple polygon: Connected, non-self-intersecting boundary.

Monotone polygon: Intersection with any vertical line is one segment.

Constrained Delaunay triangulation: Allows input edges as well as vertices. Triangles have empty circumcircles, meaning no visible input vertices.

SIMPLE POLYGONS

Triangulating a simple polygon is both an interesting problem in its own right and an important preprocessing step in other computations. For example, the following problems are known to be solvable in linear time once the input polygon P is triangulated: computing link distances from a given source, finding a monotone path within P between two given points, and computing the portion of P illuminated by a given line segment,

How much time does it take to triangulate a simple polygon? For practical purposes, one should use either an $O(n \log n)$ deterministic algorithm (such as the one given below for the more general case of planar straight-line graphs) or a slightly faster randomized algorithm (such as one with running time $O(n \log^* n)$ included in [Mul94]).

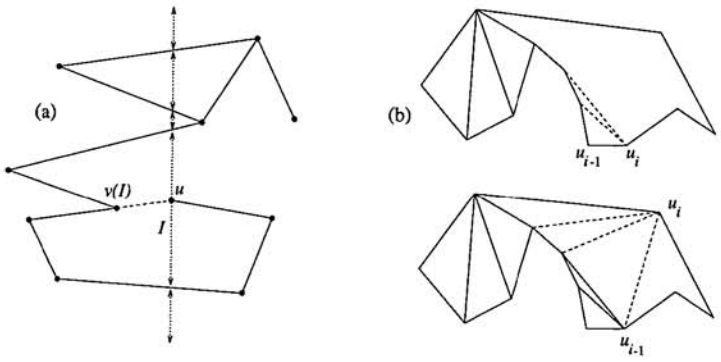
However, for theoretical purposes, achieving the ultimate running time was for several years an outstanding open problem. After a sequence of interim results, Chazelle [Cha91] devised a linear-time algorithm. Chazelle's algorithm, like previous algorithms, reduces the problem to that of computing the ***horizontal visibility map*** of P—the partition obtained by shooting horizontal rays left and right from the vertices. The "up-phase" of this algorithm recursively merges coarse visibility maps for halves of the polygon (polygonal chains); the "down-phase" refines the coarse map into the complete horizontal visibility map.

PLANAR STRAIGHT-LINE GRAPHS

Let G be a planar straight-line graph (PSLG). We describe an $O(n \log n)$ algorithm [PS85] that triangulates G in two stages, called regularization and triangulation. Regularization adds edges to G so that each vertex, except the first and last, has at least one edge extending to the left and one extending to the right. Conceptually, we sweep a vertical line ℓ from left to right across G while maintaining the list of intervals of ℓ between successive edges of G. For each interval I, we remember a vertex $v(I)$ visible to all points of I; this vertex will be either an endpoint of one of the two edges bounding I or a vertex between these edges, lacking a right edge. When we hit a vertex u with no left edge, we add the edge $\{u, v(I)\}$, where I is the interval containing u, as shown in Figure 22.2.1(a). After the left-to-right sweep, we sweep from right to left, adding right edges to vertices lacking them.

start ℓ at left with $v(interval(u)) = (-\infty, 0)$
for each vertex u from left to right **do**
    if u has no left edges **then** add edge $\{u, v(interval(u))\}$ **fi**
    delete u's left edges from interval list
    insert u's right edges with $v()$ set to u **od**
repeat the steps above for vertices from right to left

FIGURE 22.2.1
(a) *Sweep-line algorithm for regularization.* (b) *Stack-based triangulation algorithm.*

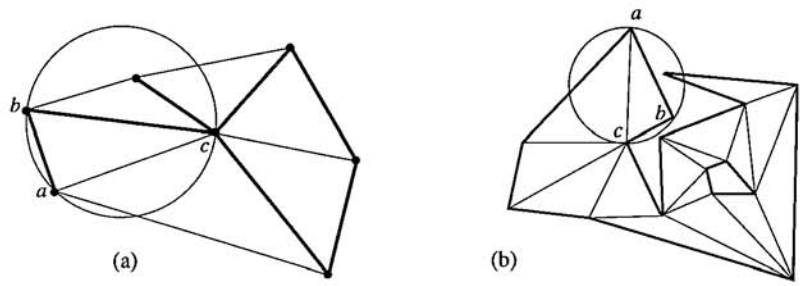


After the regularization stage, each bounded face of G is ***monotone***, meaning that a vertical line intersects the face in at most one segment. We consider the vertices $u_1, u_2, \ldots, u_n$ of a face in left-to-right order, using a stack to store the not-yet-triangulated vertices (a reflex chain) to the left of the current vertex u_i. If u_i is adjacent to u_{i-1}, the topmost vertex on the stack, as shown in the upper picture of Figure 22.2.1(b), then we pop vertices off the stack and add diagonals from these vertices to u_i, until the vertices on the stack—u_i on top—again form a reflex chain. If u_i is instead adjacent to the leftmost vertex on the stack, as shown in the lower picture, then we can add a diagonal from each vertex on the stack, and clear the stack of all vertices except u_i and u_{i-1}.

CONSTRAINED DELAUNAY TRIANGULATION

Constrained Delaunay triangulation [LL86] provides a way to force the edges of planar straight-line graph G into the DT. A point p is ***visible*** to point q if line segment pq does not intersect any edge or vertex in G, except maybe at its endpoints. A triangle abc with vertices from G appears in the ***constrained Delaunay triangulation*** (CDT) if its circumcircle neither contains nor passes through any other vertex of G visible to some point in abc. If G is a graph with vertices but not edges, then this definition generalizes ordinary, unconstrained Delaunay triangulation. If G is a polygon or polygon with holes, as in Figure 22.2.2(b), then the CDT retains only the triangles interior to G.

FIGURE 22.2.2
Constrained Delaunay triangulations of (a) *a PSLG and* (b) *a polygon with a hole.*



The edge flipping algorithm generalizes to the constrained case, with the modification that edges of G are never placed on the queue. There are also $O(n \log n)$-time algorithms for the CDT, and even a randomized $O(n)$ algorithm for the case that G is just a simple polygon [KL93].

22.3 OPTIMAL TRIANGULATIONS

We have already seen two types of optimal triangulations: the DT and the CDT. Some applications, however, demand triangulations with properties other than those optimized by these two triangulations. Table 22.3.1 gives a summary of results; each result holds for arbitrary PSLGs, except the fourth, which applies only to polygons.

GLOSSARY

Edge insertion: Local improvement algorithm, more general than edge flipping.

Local optimum: A solution that cannot be improved by local moves.

Greedy triangulation: At each step, add the shortest valid edge.

Steiner triangulation: Extra, noninput, points are allowed.

TABLE 22.3.1 Optimal triangulation results.

PROPERTY	ALGORITHMS	TIME
Delaunay	various algorithms [For95]	$O(n \log n)$
Minmax angle	fast edge insertion [ETW92]	$O(n^2 \log n)$
Minmax slope	edge insertion [BEE+93]	$O(n^3)$
Min total length	approx'n algorithms [Epp94, LK96]	$O(n \log n)$
Minmax edge length	MST induces polygons [ET91]	$O(n^2)$
Greedy	dynamic Voronoi diagram [LL92]	$O(n^2)$

EDGE FLIPPING AND EDGE INSERTION

The edge flipping DT algorithm can be modified to compute many other optimal triangulations. For example, if we redefine "reversed" to mean a quadrilateral triangulated with the diagonal that forms the larger maximum angle, then edge flipping can be used to minimize the maximum angle. For minmax angle, however, edge flipping computes only a local optimum, not necessarily the true global optimum.

Although edge flipping seems to work well in practice [ETW92], its theoretical guarantees are very weak: the running time is not known to be polynomially bounded and the local optimum it finds may be greatly inferior to the true optimum.

A more general local improvement method, called ***edge insertion*** [BEE+93, ETW92] exactly solves certain minmax optimization problems, including minmax angle and minmax slope of a piecewise-linear surface.

Assume that the input is a planar straight-line graph G, and we are trying to minimize the maximum angle. Starting from some initial triangulation of G, edge insertion repeatedly adds a candidate edge e that subdivides the maximum angle. (In general, edge insertion always breaks up a worst triangle by adding an edge incident to its "worst vertex.") The algorithm then removes the edges that are crossed by e, forming two polygonal holes alongside e. Holes are retriangulated by repeatedly removing ***ears*** (triangles with two sides on the boundary, as shown in Figure 22.3.1) with maximum angle smaller than the old worst angle $\angle cab$. If retriangulation succeeds, then the overall triangulation improves and edge bc is eliminated as a future candidate. If retriangulation fails, then the overall triangulation is returned to its state before the insertion of e, and e is eliminated as a future candidate. Each candidate insertion takes time $O(n)$, giving a total running time of $O(n^3)$.

compute an initial triangulation with all $\binom{n}{2}$ edge slots unmarked
while $\exists$ an unmarked edge e cutting the worst vertex of worst triangle abc **do**
  add e and remove all edges crossed by e
  try to retriangulate by removing ears better than abc
  if retriangulation succeeds **then**
    mark bc
  else mark e and undo e's insertion **fi od**

Edge insertion can compute the minmax "eccentricity" triangulation or the minmax slope surface [BEE+93] in time $O(n^3)$. By inserting candidate edges in a certain order, one can improve the running time to $O(n^2 \log n)$ for minmax angle [ETW92] and maxmin triangle height.

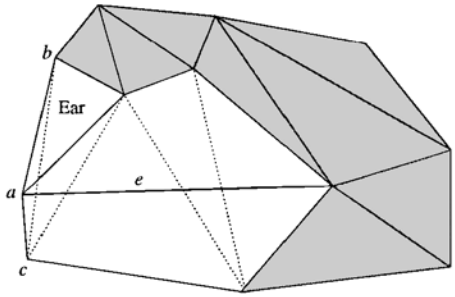


FIGURE 22.3.1
Edge insertion retriangulates holes by removing sufficiently good ears. (From [BE95]*, with permission.)*

MINIMUM WEIGHT TRIANGULATION

Several natural optimization criteria can be defined using edge lengths [BE95]. The most famous such criterion—called ***minimum weight triangulation***—asks for a triangulation of a planar point set minimizing the total edge length. No polynomial-time algorithm is known for this problem, nor is it known to be NP-complete. The best approximation algorithm, by Levcopoulos and Krznaric [LK96], gives a solution within a constant multiplicative factor of the optimal length. Eppstein gave a constant-factor approximation ratio for minimum weight Steiner triangulation, in which extra vertices are allowed.

A commonly used heuristic for minimum weight triangulation is ***greedy triangulation***. This algorithm adds edges one at a time, each time choosing the shortest edge that is not already crossed. Greedy triangulation can be viewed as an optimal triangulation in its own right, because it lexicographically minimizes the sorted vector of edge lengths. For arbitrary planar point sets, the greedy triangulation can be computed in time $O(n^2)$ by dynamic maintenance of a bounded Voronoi diagram [LL92].

Another natural criterion asks for a triangulation minimizing the maximum edge length. Edelsbrunner and Tan [ET91] showed that such a triangulation—like the DT—must contain the edges of the minimum spanning tree (MST). This geometric lemma gives the following polynomial-time algorithm: compute the MST and then triangulate the resulting simple polygons using dynamic programming.

OPEN PROBLEMS

1. Explain the empirical success of edge flipping for non-Delaunay optimization criteria, both solution quality and running time.
2. Settle the complexity of min weight triangulation—in P or NP-complete?

22.4 MESH GENERATION

A ***mesh*** is a decomposition of a geometric domain into ***elements***, usually triangles or quadrilaterals in $\mathbb{R}^2$. Meshes are used to discretize continuous functions, especially solutions to partial differential equations. Practical mesh generation problems tend to be application-specific: one desires small elements where the function

changes rapidly and larger elements elsewhere. However, certain goals apply fairly generally, and computational geometers have formulated problems incorporating these considerations. Table 22.4.1 summarizes these results, and below we discuss some of them in detail.

GLOSSARY

Steiner point: An extra vertex, not an input point.

Conforming mesh: Elements exactly fill out the input domain.

Quad tree: A recursive subdivision of the plane with squares.

TABLE 22.4.1 Mesh generation results. ("Opt" means "Optimal.")

PROPERTY	INPUTS	ALGORITHMS	SIZE
No small angles	PSLGs	quad trees [BEG94], circles [Rup93]	$O(1) \cdot \text{Opt}$
No small solid angles	polyhedra	octrees [MV92]	$O(1) \cdot \text{Opt}$
No small or obtuse	PSLGs	grids [BGR88], quad trees	$O(1) \cdot \text{Opt}$
No obtuse angles	polygons	disk packing [BMR94]	$O(n)$
No obtuse angles	some PSLGs	grids [BE92]	$O(n^4)$
No large angles	PSLGs	propagating horns [Mit93, Tan94]	$O(n^2)$
Conforming Delaunay	PSLGs	blocking & propagation [ET93]	$O(n^3)$

NO SMALL ANGLES

Sharp angles can degrade appearance and accuracy, so most mesh generation methods attempt to avoid small angles. (There is an exception: properly aligned sharp triangles prove quite useful in simulations of viscous flow.)

Baker et al. [BGR88] gave a grid-based algorithm for triangulating a PSLG so that all *new* angles—a sharp angle in the input cannot be erased—measure at least 14°. Bern et al. [BEG94] used quad trees instead of a uniform grid and proved the following efficiency guarantee: the number of triangles is $O(1)$ times the minimum number in any no-small-angle triangulation of the input. The number of triangles required depends not just on the number of input vertices n, but also on the geometry of the input. Ruppert [Rup93] devised a Delaunay-based algorithm with the same guarantee. The main loop of Ruppert's algorithm uses only two steps; one adds the circumcenter of a too-sharp triangle and the other subdivides a boundary edge.

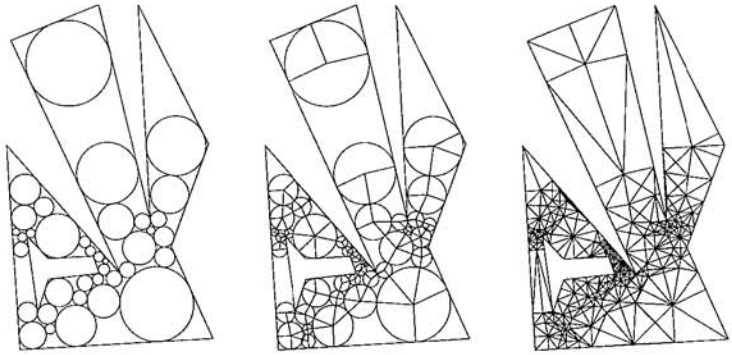
NO OBTUSE ANGLES

Some mesh generation methods also try to avoid obtuse angles, because nonobtuse meshes have some desirable numerical and geometric properties.

Computational geometers have studied the two angle conditions in isolation. Surprisingly, it is possible to triangulate a polygon (with holes) with only $O(n)$

nonobtuse triangles [BMR94]. Figure 22.4.1 illustrates the algorithm: the domain is packed with nonoverlapping disks until each uncovered region has either 3 or 4 sides; radii to tangencies are added in order to split the domain into small polygons; and finally these polygons are triangulated with right triangles, without adding any new subdivision points (vertices embedded within edges).

FIGURE 22.4.1
Nonobtuse triangulation steps. (From [BMR94], [BE95], *with permission.)*



NO LARGE ANGLES

By relaxing the bound on the largest angle from 90° to something larger, researchers have obtained results for arbitrary PSLGs. Mitchell [Mit93] gave an algorithm that uses $O(n^2 \log n)$ triangles to guarantee that all angles measure less than $\frac{7}{8}\pi$. The algorithm traces a cone of possible angle-breaking edges, called a *horn*, from each vertex—including subdivision points—with a larger angle. Horns propagate around the PSLG until meeting an exterior edge or another horn. By adding some more horn-stopping "traps," Tan [Tan94] improved the angle bound to $\frac{11}{15}\pi$ and the complexity bound to $O(n^2)$, matching a lower bound.

CONFORMING DELAUNAY TRIANGULATION

A convenient mesh generation approach adds extra vertices—***Steiner points***—to the input, until the Delaunay triangulation of the vertices "conforms" to the input, meaning that each input edge is a union of Delaunay edges.

There are a number of algorithms for this problem in the plane; all take the basic approach of covering the input edges by disks that do not enclose any input vertices. Edelsbrunner and Tan [ET93] gave an algorithm that uses $O(n^3)$ triangles, currently the only polynomial algorithm.

OPEN PROBLEMS

1. Does every PSLG have a polynomial-size nonobtuse triangulation?
2. Does every PSLG have a conforming Delaunay triangulation of size $O(n^2)$?

22.5 POLYHEDRA

In this section we specifically discuss the triangulation (or ***tetrahedralization***) of three-dimensional polyhedra. A polyhedron P is a flat-sided (connected) solid, usually assumed to satisfy the following nondegeneracy condition: around any point on the boundary of P, a sufficiently small ball contains one connected component of each of the interior and exterior of P. With this assumption, the numbers of vertices, edges, and faces (facets) of P are all linearly related.

GLOSSARY

Reflex edge: An edge with interior dihedral angle greater than 180°. (The dihedral angle between faces is measured on a plane normal to the shared edge.)

Convex polyhedron: A polyhedron without reflex edges.

Simple polyhedron: Topologically equivalent to a ball; edge skeleton forms a planar graph.

General polyhedron: May be topologically equivalent to a solid torus or higher-genus object, and may have more than one boundary component (i.e., cavities).

BAD EXAMPLES

Three dimensions is not as nice as two. Triangulations of the same input may contain different numbers of tetrahedra. For example, a triangulation of an n-vertex convex polyhedron may have as few as $n-3$ or as many as $\binom{n-2}{2}$ tetrahedra. And—what is worse—not all nonconvex polyhedra can be triangulated without Steiner points.

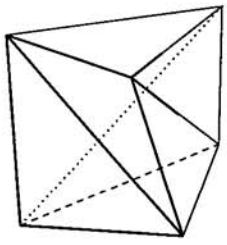


FIGURE 22.5.1
A twisted prism cannot be triangulated without Steiner points.

Schönhardt's polyhedron, shown in Figure 22.5.1, is the simplest example of a polyhedron that cannot be triangulated. Ruppert and Seidel [RS92] proved the

NP-completeness of determining whether a polyhedron can be triangulated without Steiner points, and of testing whether k Steiner points suffice.

Chazelle [Cha84] gave an n-vertex polyhedron that requires $\Omega(n^2)$ Steiner points. This polyhedron is a box with thin wedges removed from the top and bottom faces (Figure 22.5.2). The tips of the wedges nearly meet at the hyperbolic surface $z = xy$ and divide this surface into $\Omega(n^2)$ small squares, no pair of which can lie in the same tetrahedron in a triangulation.

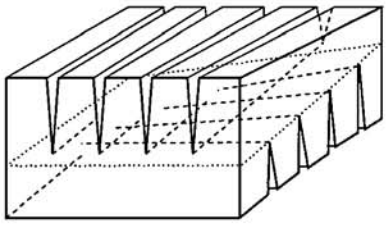


FIGURE 22.5.2
A polyhedron that requires $\Omega(n^2)$ tetrahedra. (From [BE95], *with permission.)*

GENERAL POLYHEDRA

Any polyhedron can be triangulated with $O(n^2)$ tetrahedra, matching the lower bound. One algorithm shoots vertical walls up and down from each edge of the polyhedron boundary; walls stop when they reach some other part of the boundary. The tops and bottoms of the resulting "cylinders" are then triangulated to produce $O(n^2)$ triangular prisms, which can each be triangulated with a single interior Steiner point. An improvement first plucks off "pointed vertices" with unhindered "caps." Such a vertex, together with its incident faces, forms an empty convex cone. The improved algorithm uses $O(n + r^2)$ tetrahedra, where r is the number of reflex edges on the original polyhedron [CP90].

An alternative algorithm [Cha84] divides the polyhedron into convex solids by incrementally bisecting each reflex angle with a plane that extends away from the reflex angle in all directions until it first contacts the polyhedron boundary. This algorithm produces at most $O(nr + r^{7/3})$ tetrahedra [HS92].

SPECIAL POLYHEDRA

Any convex polyhedron can be triangulated with at most $2n-7$ tetrahedra by "starring" from a vertex. The region between two convex polyhedra (the convex hull of the union, minus the polyhedra), with a total of n vertices, can be triangulated without any Steiner points. If Steiner points are allowed, $O(n)$ tetrahedra suffice. The union of three convex polyhedra can also be tetrahedralized without Steiner points. The region between a convex polyhedron and a terrain can be triangulated with $O(n \log n)$ tetrahedra, and in fact, some such regions require $\Omega(n \log n)$ tetrahedra [CS94].

3-d MESH GENERATION

Mesh generation for 3-d solids is an important, largely open, practical problem. Current approaches include octrees (the generalization of quad trees), "advancing

front," and conforming Delaunay triangulation [Bak89, BE95], but no one method gives satisfactory results for all applications. Mitchell and Vavasis [MV92] gave an octree method that guarantees no small angles and efficiency within a constant factor of optimal, the generalization of [BEG94] to $\mathbb{R}^3$. Constrained Delaunay triangulation does not extend to $\mathbb{R}^3$, because not every polyhedron has a triangulation without Steiner points, and even "easy" polyhedra may not have triangulations that use only tetrahedra with empty circumspheres.

OPEN PROBLEMS

1. Is there a polynomial-time algorithm for determining the minimum number of tetrahedra needed to triangulate a convex polyhedron?
2. Can the region between k convex polytopes, with n vertices in total, be triangulated with $O(n + k^2)$ tetrahedra?
3. Give an *input-sensitive* tetrahedralization algorithm, for example, one that uses only $O(1)$ times the smallest number of tetrahedra.

22.6 ARBITRARY DIMENSION

We now discuss triangulation algorithms for arbitrary dimension $\mathbb{R}^d$. In our big-O expressions, we consider the dimension d to be fixed.

GLOSSARY

Polytope: A bounded intersection of halfspaces in $\mathbb{R}^d$.

Face: A subpolytope such as a vertex, edge, or 2-dimensional face.

Simplex: The convex hull of $d + 1$ affinely independent points in $\mathbb{R}^d$.

Circumsphere: The sphere through the vertices of a simplex.

POINT SETS

Delaunay triangulation—and more generally regular triangulation—extends to $\mathbb{R}^d$. The DT contains a simplex if and only if its circumsphere neither encloses nor passes through any other input points. The lifting map generalizes as well, and can be used to show that the DT includes at most $O(n^{\lceil d/2 \rceil})$ simplices. In $\mathbb{R}^3$, however, any point set admits a non-Delaunay triangulation with only $O(n)$ simplices [EPW90].

Due to the lifting relation, any convex hull algorithm can be used to compute DTs. Many of the two-dimensional algorithms mentioned above also generalize to $\mathbb{R}^d$; however, the generalization of the edge flipping algorithm is not entirely straightforward. A "flip" in $\mathbb{R}^d$ can involve any two triangulations of $d + 2$ points in convex position. Flipping from an arbitrary triangulation can get stuck before reaching the DT, but incrementally adding points and then flipping cannot.

Flipping endows the set of all regular triangulations with the structure of a higher-dimensional polytope [BFS90].

Only one optimality property is known to generalize to higher dimensions: the DT minimizes the maximum radius of a simplex enclosing sphere. The ***enclosing sphere*** is the smallest sphere containing a simplex, either the circumsphere or the circumsphere of some face.

The following is known about Steiner triangulations of point sets in $\mathbb{R}^d$. It is always possible to add $O(n)$ Steiner points, so that the DT of the augmented point set has size only $O(n)$, and there is always a nonobtuse Steiner triangulation containing at most $O(n^{\lceil d/2 \rceil})$ path simplices [BCER95]. A path simplex is one containing a path of d pairwise orthogonal edges.

POLYTOPES

Triangulations of polytopes in $\mathbb{R}^d$ arise in combinatorics and algebra [GKZ90, Sta80]. Several algorithms are known for triangulating the hypercube, but there is still a large gap between the most efficient algorithm (least number of simplices) and the best lower bound [Hai91]; see Section 14.5.2. It is known that the region between two convex polytopes—a nonconvex polytope—can always be triangulated without Steiner points [GP88]; see Section 14.3.1.

22.7 SOURCES AND RELATED MATERIAL

SURVEYS

For more complete descriptions and references, consult the following sources.

[Aur91]: Describes a number of generalizations of the Voronoi diagram and Delaunay triangulation.

[For95]: Emphasizes algorithms for Delaunay triangulation.

[BE95]: A survey of optimal triangulation and mesh generation problems.

[Geo91]: Discusses mesh generation for finite element methods.

The World Wide Web currently has a number of good pages on mesh generation and triangulation; see Chapter 52.

RELATED CHAPTERS

Chapter 14: Subdivisions and triangulations of polytopes
Chapter 20: Voronoi diagrams and Delaunay triangulations
Chapter 23: Polygons

REFERENCES

[Aur91] F. Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23:345–405, 1991.

[Bak89] T.J. Baker. Developments and trends in three-dimensional mesh generation. *Appl. Numer. Math.*, 5:275–304, 1989.

[BCER95] M. Bern, L.P. Chew, D. Eppstein, and J. Ruppert. Dihedral bounds for mesh generation in high dimensions. In *Proc. 6th ACM-SIAM Sympos. Discrete Algorithms*, pages 189–196, 1995.

[BE92] M. Bern and D. Eppstein. Polynomial-size nonobtuse triangulation of polygons. *Internat. J. Comput. Geom. Appl.*, 2:241–255, 1992.

[BE95] M. Bern and D. Eppstein. Mesh generation and optimal triangulation. In D.-Z. Du and F.K. Hwang, editors, *Computing in Euclidean Geometry, 2nd Edition*, pages 47–123. World Scientific, Singapore, 1995.

[BEE+93] M. Bern, H. Edelsbrunner, D. Eppstein, S. Mitchell, and T.-S. Tan. Edge-insertion for optimal triangulations. *Discrete Comput. Geom.*, 10:47–65, 1993.

[BEG94] M. Bern, D. Eppstein, and J.R. Gilbert. Provably good mesh generation. *J. Comput. Syst. Sci.*, 48:384–409, 1994.

[BFS90] L. Billera, P. Filliman, and B. Sturmfels. Constructions and complexity of secondary polytopes. *Adv. Math.*, 83:155–179, 1990.

[BGR88] B.S. Baker, E. Grosse, and C.S. Rafferty. Nonobtuse triangulation of polygons. *Discrete Comput. Geom.*, 3:147–168, 1988.

[BMR94] M. Bern, S. Mitchell, and J. Ruppert. Linear-size nonobtuse triangulation of polygons. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 221–230, 1994.

[Cha84] B. Chazelle. Convex partitions of polyhedra: a lower bound and worst-case optimal algorithm. *SIAM J. Comput.*, 13:488–507, 1984.

[Cha91] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6:485–524, 1991.

[CP90] B. Chazelle and L. Palios. Triangulating a nonconvex polytope. *Discrete Comput. Geom.*, 5:505–526, 1990.

[CS94] B. Chazelle and N. Shouraboura. Bounds on the size of tetrahedralizations. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 231–239, 1994.

[Epp94] D. Eppstein. Approximating the minimum weight triangulation. *Discrete Comput. Geom.*, 11:163–191, 1994.

[EPW90] H. Edelsbrunner, F.P. Preparata, and D.B. West. Tetrahedrizing point sets in three dimensions. *J. Symbolic Comput.*, 10:335–347, 1990.

[ET91] H. Edelsbrunner and T.-S. Tan. A quadratic time algorithm for the minmax length triangulation. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 414–423, 1991.

[ET93] H. Edelsbrunner and T.-S. Tan. An upper bound for conforming Delaunay triangulations. *Discrete Comput. Geom.*, 10:197–213, 1993.

[ETW92] H. Edelsbrunner, T.S. Tan, and R. Waupotitsch. A polynomial time algorithm for the minmax angle triangulation. *SIAM J. Sci. Statist. Comput.*, 13:994–1008, 1992.

[For95] S. Fortune. Voronoi diagrams and Delaunay triangulations. In F.K. Hwang and D.-Z. Du, editors, *Computing in Euclidean Geometry, 2nd Edition*, pages 225–265. World Scientific, Singapore, 1995.

[Geo91] P.L. George. *Automatic Mesh Generation.* Wiley, New York, 1991.

[GKZ90] I.M. Gelfand, M.M. Kapranov, and A.V. Zelvinsky. Newton polytopes of the classical discriminant and resultant. *Adv. Math.*, 84:237–254, 1990.

[GP88] J.E. Goodman and J. Pach. Cell decomposition of polytopes by bending. *Israel J. Math.*, 64:129–138, 1988.

[Hai91] M. Haiman. A simple and relatively efficient triangulation of the n-cube. *Discrete Comput. Geom.*, 6:287–289, 1991.

[HS92] J. Hershberger and J. Snoeyink. Convex polygons made from few lines and convex decompositions of polyhedra. In *Proc. 3rd Scand. Workshop on Algorithm Theory*, volume 621 of *Lecture Notes in Comput. Sci.*, pages 376–387. Springer-Verlag, New York, 1992.

[KL93] R. Klein and A. Lingas. A linear-time randomized algorithm for the bounded Voronoi diagram of a simple polygon. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 124–132, 1993.

[LK96] C. Levcopoulos and D. Krznaric. Quasi-greedy triangulations approximating the minimum weight triangulation. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pages 392–401, 1996.

[LL86] D.T. Lee and A. Lin. Generalized Delaunay triangulation for planar graphs. *Discrete Comput. Geom.*, 1:201–217, 1986.

[LL92] C. Levcopoulos and A. Lingas. Fast algorithms for greedy triangulation. *BIT*, 32:280–296, 1992. Also in *Proc. 2nd Scand. Workshop Algorithm Theory*, volume 447 of *Lecture Notes in Comput. Sci.*, pages 238–250. Springer-Verlag, New York, 1990.

[Mit93] S.A. Mitchell. Refining a triangulation of a planar straight-line graph to eliminate large angles. In *Proc. 34th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 583–591, 1993.

[Mul94] K. Mulmuley. *Computational Geometry: An Introduction through Randomized Algorithms.* Prentice-Hall, Englewood Cliffs, 1994.

[MV92] S.A. Mitchell and S. Vavasis. Quality mesh generation in three dimensions. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 212–221, 1992.

[PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, 1985.

[RS92] J. Ruppert and R. Seidel. On the difficulty of tetrahedralizing 3-dimensional non-convex polyhedra. *Discrete Comput. Geom.*, 7:227–253, 1992.

[Rup93] J. Ruppert. A new and simple algorithm for quality 2-dimensional mesh generation. In *Proc. 4th ACM-SIAM Sympos. Discrete Algorithms*, pages 83–92, 1993.

[Sta80] R.P. Stanley. Decompositions of rational convex polytopes. *Ann. Discrete Math.*, 6:333–342, 1980.

[Tan94] T.-S. Tan. An optimal bound for conforming quality triangulations. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 240–249, 1994.

23 POLYGONS

Subhash Suri

INTRODUCTION

Polygons are one of the fundamental building blocks in geometric modeling, and they are used to represent a wide variety of shapes and figures in computer graphics, vision, pattern recognition, robotics, and other computational fields. By a polygon we will mean a region of the plane enclosed by a simple cycle of straight line segments; a ***simple cycle*** means that nonadjacent segments do not intersect and two adjacent segments intersect only at their common endpoint. This chapter describes a collection of results on polygons with both combinatorial and algorithmic flavors. After classifying polygons in the opening section, Section 23.2 covers polygon decomposition, and Section 23.3 polygon intersection. Sections 23.4 and 23.5, respectively, discuss path finding problems and polygon containment problems. Section 23.6 touches upon a few miscellaneous problems and results.

23.1 POLYGON CLASSIFICATION

Polygons can be classified in several different ways depending on their domain of application. In VLSI applications, for instance, the most commonly used polygons have their sides parallel to the coordinate axes.

GLOSSARY

Simple polygon: A closed region of the plane enclosed by a simple cycle of straight line segments.

Convex polygon: The line segment joining any two points of the polygon lies within the polygon.

Monotone polygon: Any line parallel to some fixed direction intersects the polygon in a single connected piece.

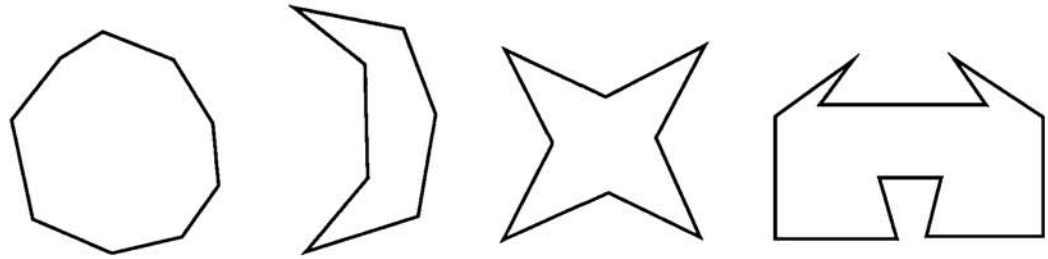
Star-shaped polygon: The entire polygon is visible from some point inside the polygon.

Orthogonal polygon: A polygon with sides parallel to the (orthogonal) coordinate axes. Sometimes called a ***rectilinear polygon.***

POLYGON TYPES

Before starting our discussion on problems and results concerning polygons, we clarify a few technical issues. The qualifier "simple" in the definition of a simple

FIGURE 23.1.1
A classification of polygons.



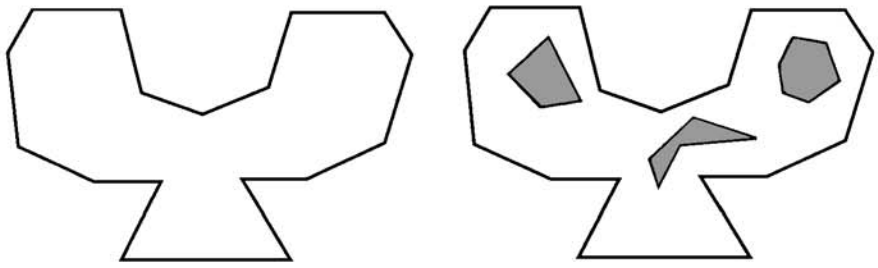
polygon states a *topological* property, meaning "non-self-intersection." Not to be confused with "uncomplicated polygons," in fact, these polygons include the most complex among polygons that are topologically equivalent to a disk (see the classification below). Finally, we will make a standard ***general position*** assumption throughout this chapter that no three vertices of a polygon are collinear.

The following *hierarchical* classification of polygons is one of the most commonly used (see Figure 23.1.1):

$$\text{CONVEX} \subset \begin{matrix}\text{STAR-SHAPED} \\ \text{MONOTONE}\end{matrix} \subset \text{SIMPLE POLYGONS}$$

This hierarchy is best explained using the concept of visibility (see Chapter 25). We say that two points x and y in a polygon P are mutually ***visible*** if the line segment $\overline{xy}$ does not intersect the complement of P; thus the segment $\overline{xy}$ is allowed to graze the polygon boundary but not cross it. We call a set of points $K \subset P$ the ***kernel*** of P if all points of P are visible from every point in the kernel (see Figure 33.4.4). Then, a polygon P is convex if $K = P$; the polygon is star-shaped if $K \neq \emptyset$; otherwise, the polygon is merely a simple polygon. Speaking somewhat loosely, a monotone polygon can be viewed as a special case of a star-shaped polygon

FIGURE 23.1.2
Examples of a simple polygon and a polygon with holes.



with the exterior kernel at infinity—that is, a monotone polygon can be decomposed into two polygonal chains, each of which is entirely visible from the (same) point at infinity in the extended plane. Notice that the star-shaped polygon in Figure 23.1.1 is also a monotone polygon.

By definition, a simple polygon P is a polygon *without holes*—that is, the interior of the polygon is topologically equivalent to a disk. A ***polygon with holes*** is a higher-genus variant of a simple polygon, obtained by removing a nonoverlapping set of strictly interior, simple subpolygons from P. Figure 23.1.2 illustrates the distinction between a simple polygon and a polygon with holes.

Finally, an important class of polygons are the *orthogonal polygons*, where all edges are parallel to the coordinate axes. These polygons arise quite naturally in certain applications such as VLSI design. In this chapter, however, we will be concerned primarily with simple polygons, with no restriction on the orientation of edges.

23.2 POLYGON DECOMPOSITION

Many computational geometry algorithms that operate on polygons first decompose them into more elementary pieces, such as triangles or quadrilaterals. There is a substantial body of literature in computational geometry on this subject. The most celebrated problem in this category is the "polygon triangulation problem."

GLOSSARY

Steiner point: A vertex not part of the input set.

Diagonal: A line segment connecting two polygon nonadjacent vertices and contained in the polygon. An ***edge*** connects adjacent vertices.

Polygon cover: A collection of subpolygons whose union is exactly the input polygon.

Polygon partition: A collection of subpolygons with *pairwise disjoint* interiors whose union is exactly the input polygon.

TRIANGULATION

The polygon triangulation problem is to dissect a polygon into triangles by drawing a maximal number of noncrossing diagonals. Only the vertices of the polygon are used as triangle vertices, and no additional (Steiner) vertices are allowed. It is an easy and well-known result that every simple polygon can be triangulated, and that the number of triangles is invariant over all triangulations. More precisely:

THEOREM 23.2.1

Every simple polygon admits a triangulation, and every triangulation of an n-vertex polygon has $n-3$ diagonals and $n-2$ triangles.

The number of possible diagonals in a polygon may vary from linear (e.g., a

spiral polygon) to quadratic (e.g., a convex polygon). A diagonal that breaks the polygon into two roughly equal halves is called a ***balanced*** diagonal. In designing his $O(n \log n)$ time algorithm for triangulating a polygon, Chazelle [Cha82] proved the following fact, which has found numerous applications in divide-and-conquer based algorithms for polygons:

THEOREM 23.2.2

Every n-vertex simple polygon admits a diagonal that breaks the polygon into two subpolygons, neither one with more than $\lceil 2n/3 \rceil + 1$ *vertices.*

By recursively dividing the polygon using balanced diagonals, we get a balanced decomposition of P, which can be modeled by a tree of height $O(\log n)$. The existence of a balanced diagonal follows easily once we consider the graph-theoretic dual of a triangulation. This dual graph of a polygon triangulation is a tree, with maximum node degree three. Diagonals of the triangulation correspond to the edges of the dual tree, and thus a balanced diagonal corresponds to an edge whose removal breaks the tree into two subtrees, each with at most $\lceil 2n/3 \rceil + 1$ nodes.

The problem of computing a triangulation of a polygon has had a long and distinguished history [O'R87], culminating in Chazelle's linear-time algorithm [Cha91]. Table 23.2.1 lists some of the best-known algorithms for this problem. The algorithm in [Sei91] is a randomized Las Vegas algorithm (see Chapter 34). All others are deterministic algorithms, with worst-case time bounds as shown.

TABLE 23.2.1 Results on triangulating a simple polygon.

YEAR	TIME COMPLEXITY	ALGORITHM
1978	$O(n \log n)$	monotone pieces
1982	$O(n \log n)$	divide-and-conquer
1985	$O(n \log n)$	plane sweep
1991	$O(n \log^* n)$	randomized
1991	$O(n)$	polygon cutting

Finally, if the polygon contains holes, then it has been shown that $\Theta(n \log n)$ time is both necessary and sufficient for triangulating the region [HM85]. See Table 23.2.2.

TABLE 23.2.2 Results on triangulating a polygon with holes.

YEAR	TIME COMPLEXITY	ALGORITHM
1985	$O(n \log n)$	plane sweep
1994	$O(n \log n)$	local sweep

COVERS AND PARTITIONS

The problem of decomposing polygons into different types of simpler polygons has numerous applications within and without computational geometry (see, e.g., Chapter 43). Unlike the triangulation problem, most variants of the covering and partitioning problems turn out to be provably hard. In a covering problem, the goal is to cover the interior of the polygon with the smallest number of subpolygons of a particular type, for instance, convex or star-shaped polygons. Table 23.2.3 lists results for various polygon covering problems. In this table, "cover type" refers to the family of polygons allowed in the cover, while "domain" refers to the polygonal region that needs to be covered. For the most part, we consider only four types of domains: simple polygons, with and without holes, and orthogonal polygons, with and without (orthogonal) holes. In all of these problems, the cover or partition pieces are allowed to use Steiner points for their vertices. Almost all variations of the covering problem are intractable. The last important open problem in this area, determining the complexity of covering polygons by convex pieces, was settled in [CR88]; this paper also serves as a good source of pointers to related work on polygon covering problems.

TABLE 23.2.3 Results on polygon covering problems.

COVER TYPE	DOMAIN	HOLES	COMPLEXITY
Rectangles	orthogonal	Y	NP-complete
Convex–star	polygons	Y	NP-hard
Star	polygons	N	NP-hard
Rectangles	orthogonal	N	NP-hard
Convex	polygons	N	NP-hard

The polygon-partitioning problems are similar to the covering problem, except that the tesselating pieces are not allowed to overlap. Table 23.2.4 collects results on polygon partitioning problems. Polynomial-time algorithms can be achieved for simple polygons using the dynamic programming technique. The same problems, however, turn out to be intractable when the polygon has holes.

TABLE 23.2.4 Results on polygon partitioning problems.

PARTITION	DOMAIN	HOLES	COMPLEXITY
Convex	polygons	N	$O(n^3)$
Convex	polygons	Y	NP-hard
Trapezoids	polygons	N	$O(n^2)$
Trapezoids	polygons	Y	NP-complete
Rectangles	orthogonal	Y	$O(n^{3/2} \log n)$

Two useful references for polygon partitioning problems are [AAI86] and [Kei85]. The latter presents several polynomial-time algorithms for optimally partitioning a simple polygon into convex pieces *without* using Steiner points. See Chapter 43 for applications of polygon decomposition problems.

The intractability of most covering and partitioning problems naturally leads to the question of approximability—how well can we approximate the size of an optimal cover or partition in polynomial time. In many cases, there are only a polynomial number of covering candidates—for instance, rectangle covers or convex polygon covers. In these cases, a greedy set-cover heuristic can be used to achieve an approximation factor of $O(\log n)$.

OPEN PROBLEM

Approximating the number of art gallery guards: Give a polynomial-time algorithm for computing a constant-factor approximation of the minimum number of point guards needed to cover a simple polygon.

23.3 POLYGON INTERSECTION

Polygon intersection problems deal with issues of detection and computation of the collision between two polygonal shapes. In the detection problem, one is only interested in deciding *whether* the two polygons have a point in common. In the intersection computation problem, the algorithm is asked to report the overlapping parts of the two polygons. Such problems arise naturally in robotics and computer games; see Chapter 33 for additional material.

The maximum *number* of points at which two polygons may cross each other depends on the type of polygons. If p and q, respectively, denote the number of vertices of the two polygons, then the maximum number of intersections is $\min(2p, 2q)$ if both polygons are convex, $\max(2p, 2q)$ if one is convex, and pq otherwise.

Algorithmically, intersection-detection between convex polygons can be done significantly faster than intersection computation, if we allow reasonable preprocessing of polygons. By a reasonable preprocessing, we mean that the preprocessing algorithm takes into account the *structure* of the polygons but *not their positions.* In Table 23.3.1, n denotes the total number of vertices in the two polygons; that is, $n = p + q$.

TABLE 23.3.1 Intersecting polygons.

POLYGON TYPES	PREPROCESSING	QUERY
Convex-convex	$O(1)$	$O(n)$
Convex-convex	$O(n)$	$O(\log n)$
Simple-simple	$O(1)$	$O(n)$
Simple-simple	$O(n \log n)$	$O(m \log^2 n)$

The parameter m in the query time for intersections of two simple polygons is the complexity of a *minimum link witness* for the intersection or disjointness of the two polygons, and we always have $m \leq n$. The preprocessing space requirement is linear when the polygons are preprocessed.

23.4 PATHS IN POLYGONS

Path planning in polygons is another well-studied area of research. An abstract robot motion planning problem (Chapter 40) is to find a shortest path for a point in the midst of a collection of disjoint polygons in the plane. This simplified scenario lets us focus exclusively on the combinatorial aspect of the robotics problem, ignoring such practical issues as kinematics and control (Chapter 41). The polygons represent obstacles in the path of the robot, which itself is modeled as a point. The free space is the set of all points accessible to the robot via a free path. By convention, for the case of a single polygon, the free space is defined to be the closed interior of the polygon (think of an art gallery).

GLOSSARY

Free space: The complement of the union of the interiors of obstacle polygons.

Free path: A path lying entirely in the free space.

Shortest path: A free path of minimum total length.

Shortest path tree: The union of shortest paths from one fixed vertex to all other vertices. (Strictly speaking, this may not be a tree in special cases.)

Shortest path map: The minimal partition of the plane with respect to a fixed source point s so that all points in a region have the same combinatorial structure for their shortest path to s, i.e., the list of vertices on the path is the same. See Figure 24.0.1.

Geodesic diameter: The maximum shortest path distance between any points.

Geodesic center: A point minimizing the maximum shortest path distance to all other points.

Minimum link path: An obstacle-avoiding path between two given points with the minimum number of edges.

Link distance: The link distance between two points p and q is the minimum number of straight-line segments needed in any free path connecting p and q.

Window partition: The window partition of a polygon P with respect to a source point s (or a line segment) is the minimal partition of P into regions with the property that all points in a region have the same link distance to s.

EUCLIDEAN MEASURE

The problem of computing a shortest Euclidean path between two points in the presence of polygonal obstacles is one of the best-known problems of computational geometry (see Chapter 24). The geometry of the Euclidean plane ensures that the

shortest path is a non-self-intersecting polygonal path with corners at obstacle vertices. Figure 23.4.1 shows an example of a shortest path problem. A *shortest path tree* [GHL+87] extends the notion of a single shortest path to shortest paths to all vertices of the polygonal domain from a specified source point.

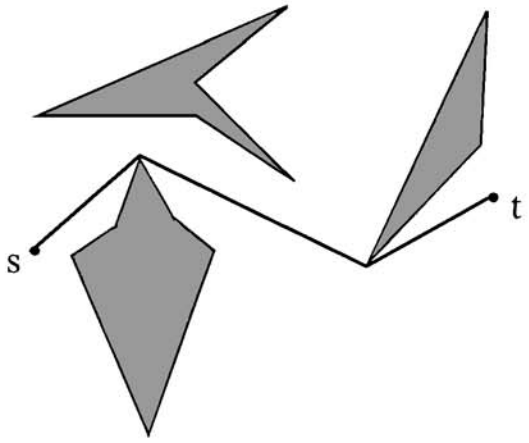


FIGURE 23.4.1
A shortest path among polygons.

The shortest path distance function is a metric, and therefore several natural measures lend themselves to our new setting: in particular, the *shortest path diameter* (also called the *geodesic diameter*) and the *geodesic center*. The following table summarizes the main results known today for these shortest path problems. The long-standing open problem of computing a shortest path map in optimal time was settled only recently [HS93].

TABLE 23.4.1 Results for Euclidean shortest paths in the plane.

PROBLEM	DOMAIN	RESULT
Shortest path	simple polygon	$O(n \log n)$
Shortest path tree	simple polygon	$O(n \log n)$
Shortest path tree	triang simple poly	$O(n)$
Geodesic diameter	simple polygon	$O(n)$
Geodesic center	simple polygon	$O(n \log n)$
Shortest path tree	polygon with holes	$O(E + n \log n)$
Shortest path map	polygon with holes	$O(n \log n)$

In the Table 23.4.1, the use of a triangulated polygon in [GHL+87, HS91] is meant to separate the cost of triangulating the polygon from the cost of computing a shortest path tree. However, since the publication of these results, a linear-time algorithm for polygon triangulation has been achieved [Cha91], making this distinction unnecessary. Interest in the geodesic diameter and center was partly motivated by Lantuejoul and Maisonneuve [LM84], who proposed these measures for quantitative image analysis.

SHORTEST PATH QUERIES

Often it is desirable to preprocess a polygon (with or without holes) to speed up subsequent query answering. For the case where the domain is a simple polygon and all queries are with respect to a fixed source point, an optimal data structure is presented in [GHL+87]. Essentially, the shortest path tree implicitly partitions the polygon into regions that have the same shortest path structure. In combination with a point-location data structure, this partition achieves $O(\log n)$ query time using $O(n)$ space. When the source point is not fixed, the problem is more difficult and requires more advanced data structuring methods. Nevertheless, an optimal solution is known with $O(n)$ space and $O(\log n)$ query time [GH89, Her91].

For polygons with holes, only the case of a fixed source point is satisfactorily solved: the algorithm of Hershberger and Suri [HS93] computes an $O(n)$-space shortest path map, which can be used to answer queries in $O(\log n)$ time apiece. When the source is not fixed, we know of no sublinear time query algorithm! The most promising direction in this case is via fast approximation algorithms; only recently has some progress been made in this direction. An algorithm by Chen [Che95] takes $O(n^{3/2} \log n)$ space and $O(\log n)$ query time to compute a $(6 + \epsilon)$-approximation of the shortest path distance.

LINK MEASURE

Another measure of distance that has received considerable attention in computational geometry is the link distance [Sur87, Sur90]. The motivation behind the link distance comes from situations where the cost of "turning" outweighs the cost of straight-line travel. Figure 23.4.2 shows an example of a minimum link path which is not a shortest path.

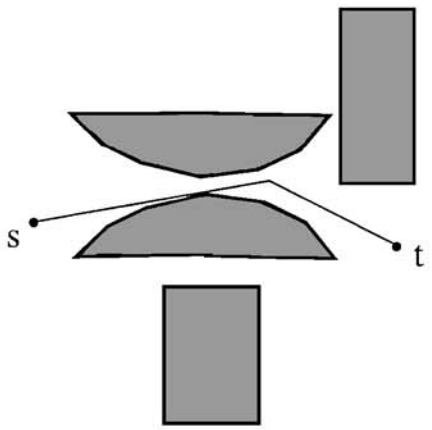


FIGURE 23.4.2
A minimum link path.

For link distance problems in a simple polygon, a construction known as a window partition has proved to be very useful [Sur90]. A window partition is best explained using the idea of visibility. All points of the polygon directly visible from

s are at link distance one. Call this set V_1. The boundary between the visible and invisible region of the polygon consists of a collection of chords, called *windows* of V_1. The points in $P \setminus V_1$ that are visible from some point of a window of V_1 form the region with link distance two. Repeating this construction yield the window partition of P. For a fixed source point in a simple polygon, the ***window tree*** data structure of Suri [Sur90] yields the optimal query time of $O(\log n)$ using $O(n)$ space. When both source and destination are specified as part of the query, the best data structure known is due to Arkin, Mitchell, and Suri [AMS92], achieving $O(\log n)$ time but at the expense of $O(n^3)$ space. Further results on link and geodesic queries can be found in Chiang and Tamassia [CT94], and Section 24.3 of this Handbook.

The problems of computing a shortest path, the diameter, and the center all extend to the link measure, and Table 23.4.2 summarizes the known results for these problems.

TABLE 23.4.2 Results for minimum link path problems.

PROBLEM	DOMAIN	RESULT
Min link path	triang simple poly	$O(n)$
Min link tree	triang simple poly	$O(n)$
Orthogonal min link path	orthogonal obstacles	$O(n \log n)$
Link diameter	simple polygon	$O(n \log n)$
Link center	simple polygon	$O(n \log n)$
Link dist query	simple poly, fixed s	$O(\log n)$, $O(n)$ space
Link dist query	simple poly, arbitrary s, t	$O(\log n)$, $O(n^3)$ space

VISIBILITY AND RAY SHOOTING

Algorithms and data structures for computing visibility have come to occupy an important role in computational geometry, in large part due to their successful application in solving other problems. In a polyhedral environment modeling a real-life scene, determining what is visible from a particular location has obvious relevance to the problem of robot motion planning. The ray shooting problem represents a very specific instance of visibility computation: determine the first point of contact between a query ray and the polyhedral scene. In addition to obvious applications in collision-detection, the ray shooting problem also plays a fundamental role in designing other computational geometry algorithms, such as data structures for the equally important "point-location" problem.

The topic of computing the visibility region of a point, line segment, or other objects is treated in Chapter 25. In the present section, we cover the results on ray shooting, which are presented in Table 23.4.3.

The query performance in the case of polygons with holes is sensitive to the number of holes—if the number of holes is $k \leq n$, then the query time for the last two algorithms improves to $O(\sqrt{k} \log n)$, with preprocessing cost $O(n\sqrt{k} + n \log n + k^{3/2} \log k)$.

TABLE 23.4.3 Results for the ray shooting problem in polygons.

DOMAIN	PREPROCESSING	QUERY
Convex polygon	$O(n)$	$O(\log n)$
Simple polygon	$O(n)$	$O(\log n)$
Polygon with holes	$O(n^{3/2} \log n)$	$O(\sqrt{n} \log n)$

OPEN PROBLEMS

1. *Shortest path query problem:* Build a data structure to compute shortest-path distance between pairs of query points in the presence of polygonal obstacles. The goal is to achieve $O(n \log n)$ space, $O(\log n)$ query time, and $O(1)$ approximation factor on the distance. (The constant of approximation should be small, say, at most 2.) No sublinear query algorithm for the exact problem is known.

2. *Non-Steiner minimum link path problem:* Given a simple polygon P and a pair of points $p, q \in P$, find a minimum link path in P from p to q subject to the condition that the path turns only at the vertices of P. Can this problem be solved in $O(n \log n)$ time?

23.5 POLYGON CONTAINMENT

Polygon containment refers to a class of problems that deals with the placement of one polygonal figure inside another. Polygon inscription, polygon circumscription, and polygon nesting are other variants of this type of problem.

GLOSSARY

Inscribed polygon: We will say that a polygon Q is inscribed in polygon P if $Q \subset P$. P is then called a ***circumscribing polygon.***

Polygon nesting: P, Q is a nested pair if $Q \subset P$ or vice versa.

CONTAINMENT OF POLYGONS

Let P, Q be two simple polygons with p and q vertices, respectively. The polygon containment problem asks for the largest copy of Q that can be contained in P using rotations and translations. (In this section, all scalings are assumed to be *uniform*; thus "shearing" is not permitted.) Several authors have considered the polygonal containment problem under various restrictions on the shape of the polygons and the allowable motions. Table 23.5.1 collects the best results known for the most important cases. See Section 40.4 for a description of the near-linear λ_s function.

TABLE 23.5.1 Results for the polygon containment problem.

P	Q	TRANSFORMS	RESULTS
Ortho-convex	ortho-convex	translate, scale	$O((p+q)^2 \log pq)$
Convex	convex	translate, scale	$O((p+q)^2 \log pq)$
Convex	convex	translate, rotate	$O(qp^2)$
Simple	simple	translate, rotate	$O(p^3q^3 \log pq)$
Convex	polygon with holes	translate, rotate, scale	$O(q^4 p\lambda_4(pq) \log p)$
Convex	points	translate, rotate, scale	$O(q^2 p\lambda_3(pq) \log p)$

See [CK93] for pointers to relevant results on polygon containment.

INSCRIBING/CIRCUMSCRIBING POLYGONS

We now consider problems related to inscribing and circumscribing polygons. In these problems, a polygon P is given, and the task is to find a polygon Q of some specified number of vertices k that is inscribed in (resp. circumscribes) P while maximizing (resp. minimizing) certain measure of Q. The common measures include area and perimeter. See Table 23.5.2 for results concerning this class of problems; n denotes the number of vertices of P. See references [AP88] and [MS90] for these results and other relevant material on this problem.

TABLE 23.5.2 Inscribing and circumscribing polygons.

TYPE	k	P	MEASURE	RESULTS
Inscribe	3	convex	max area	$O(n)$
Inscribe	k	convex	max area/perimeter	$O(kn + n \log n)$
Inscribe	convex	simple	max area	$O(n^7)$
Inscribe	3	simple	max area/perimeter	$O(n^4)$
Circumscribe	3	convex	min area	$O(n)$
Circumscribe	3	convex	min perimeter	$O(n \log n)$
Circumscribe	k	convex	min area	$O(kn + n \log n)$

NESTING POLYGONS

The nested polygon problem asks for a polygon with the smallest number of vertices that fits between two nested polygons. More precisely, given two nested polygons P and Q, where $Q \subset P$, find a polygon K of the least number of vertices such that $Q \subset K \subset P$. Generalizing the notion of nested polygons, one can also pose the problem of determining a polygonal subdivision of the least number of *edges* that "separates" a family of polygons. Table 23.5.3 lists the results on these problems. In this table, n is the total number of vertices in the input polygons, while k is the number of vertices in the output polygon (or subdivision). Reference [MS92] is a good source of pointers to other results on polygon nesting problems.

TABLE 23.5.3 Results for polygon nesting.

TYPES OF P, Q	TYPE OF K	RESULTS
Convex-convex	convex	$O(n \log k)$
Simple-simple	simple	$O(n \log k)$
Polygonal family	subdivision	NP-complete
Polygonal family	subdivision	$O(1)$-Opt in $O(n \log n)$

Several other results on polygon nesting have been obtained. In particular, if the minimum-vertex nested polygon is nonconvex, then it can be found in $O(n)$ time [GM90].

23.6 MISCELLANEOUS

There is a rather large number of results pertaining to polygons, and it would be impossible to cover them all in a single chapter. Having focused on a selected list of topics so far, we now provide below an unorganized collection of some miscellaneous results.

POLYGON MORPHING

To ***morph*** one polygon into another is to find a continuous deformation from the source polygon to the target polygon. Guibas and Hershberger [GH94] introduce the problem of morphing a simple polygon P to another simple polygon Q whose edges, taken in counterclockwise order, are parallel to the corresponding edges of P and oriented the same way. An atomic morphing step is a uniform scaling or translation of a part of the polygon. It is shown in [GH94] that $O(n^{4/3+\epsilon})$ morphing steps are always sufficient to convert one polygon to another. This result was improved shortly afterwards by Hershberger and Suri [HS95], who reduced the number of morphing steps to $O(n \log n)$.

An alternative approach to morphing is suggested by *polyhedral reconstruction*: Given two polygons lying in parallel planes, construct an interpolating polyhedron whose top and bottom faces are the two given polygons and all intermediate slices are simple polygons. See Chapter 26 for more details on reconstruction problems.

OPEN PROBLEMS

1. The transformation in [GH94] is not very natural: it morphs the source polygon to a simple intermediate shape, and then expands it to the target polygon. Explore a more natural morphing transformation.

2. Investigate the morphing problem for polygons with holes.

CSG REPRESENTATION

In [DGHS88] Dobkin et al. consider the problem of deriving a ***Peterson-style*** formula given the boundary representation of a simple polygon. A Peterson-style formula is a "constructive solid geometry" representation, in which the polygon is presented as a set of Boolean operations; see Chapter 47. Peterson proved that every simple polygon in two dimensions admits a representation by a Boolean formula on the halfplanes supporting the edges of the polygon. Furthermore, the resulting formula is *monotone*; that is, there is no negation and each halfplane appears exactly once. Dobkin et al. consider the algorithmic problem of constructing such a formula, and give an $O(n \log n)$ time algorithm, where n is the number of vertices of the polygon. Interestingly, it turns out that not all three-dimensional polyhedra admit a Peterson-style formula [DGHS88].

OPEN PROBLEM

Characterize the three-dimensional polyhedra that can be represented by Peterson-style formulas.

POLYGON SEARCHING

In these problems, the goal is to design *on-line* search strategies for locating an (identifiable) object in a polygon; the word "on-line" means that the searcher does not have a complete knowledge of the polygon, rather it "discovers" the polygon during its navigation. The motivation stems from robotics applications. Table 23.6.1 summarizes some basic results on this class of problems. (The parameter k in the last line denotes the number of distinct initial placements of the robot having the same visibility polygon.) References [IK95] and [DRW95] provide a good starting point for a search on this topic.

TABLE 23.6.1 Results for polygon searching.

ENVIRONMENT	GOAL	COMPETITIVE RATIO
n oriented rectangles	shortest path	$\Theta(\sqrt{n})$
"Street" polygon	shortest path	$1+\frac{3}{2}\pi$
Star-shaped polygon	reach kernel	≈ 5.52
Orthogonal polygon	exploration	randomized 5/4
Simple polygon	localization with min travel	$(k-1)$-Opt

23.7 SOURCES AND RELATED MATERIAL

SURVEYS

The survey article by Mitchell and Suri [MS95] addresses optimization problems in computational geometry, many involving polygons.

RELATED CHAPTERS

Chapter 22: Triangulations
Chapter 24: Shortest paths and networks
Chapter 25: Visibility
Chapter 43: Pattern recognition

REFERENCES

[AAI86] Ta. Asano, Te. Asano, and H. Imai. Partitioning a polygonal region into trapezoids. *J. Assoc. Comput. Mach.*, 33:290–312, 1986.

[AMS92] E.M. Arkin, J.S.B. Mitchell, and S. Suri. Optimal link path queries in a simple polygon. In *Proc. 3rd ACM-SIAM Sympos. Discrete Algorithms*, pages 269–279, 1992.

[AP88] A. Aggarwal and J. Park. Notes on searching in multidimensional monotone arrays. In *Proc. 29th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 497–512, 1988.

[Cha82] B. Chazelle. A theorem on polygon cutting with applications. In *Proc. 23rd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 339–349, 1982.

[Cha91] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6:485–524, 1991.

[Che95] D.Z. Chen. On the all-pairs Euclidean shortest path problem. In *Proc. 6th Annu. Sympos. Discrete Algorithms*, 1995.

[CK93] L.P. Chew and K. Kedem. Placing the largest similar copy of a convex polygon among polygonal obstacles. *Comput. Geom. Theory Appl.*, 3:59–89, 1993.

[CR88] J.C. Culberson and R.A. Reckhow. Covering polygons is hard. In *Proc. 29th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 601–611, 1988.

[CT94] Y.-J. Chiang and R. Tamassia. Optimal shortest path and minimum-link path queries between two convex polygons in the presence of obstacles. Report CS-94-03, Comput. Sci. Dept., Brown Univ., Providence, 1994.

[DGHS88] D. Dobkin, L. Guibas, J. Hershberger, and J. Snoeyink. An efficient algorithm for finding the CSG representation of a simple polygon. *Comput. Graph.* (Proc. SIGGRAPH '88), 22:31–40, 1988.

[DRW95] G. Dudek, K. Romanik, and S. Whitesides. Localizing a robot with minimum travel. *Proc. 6th ACM-SIAM Sympos. Discrete Algorithms*, pages 437–446, 1995. To appear in *SIAM J. Comput.*

[GH89] L.J. Guibas and J. Hershberger. Optimal shortest path queries in a simple polygon. *J. Comput. Syst. Sci.*, 39:126–152, 1989.

[GH94] L. Guibas, and J. Hershberger. Morphing simple polygons. *Proc. 10th Annu. Sympos. Comput. Geom.*, pages 267–276, 1994.

[GHL+87] L.J. Guibas, J. Hershberger, D. Leven, M. Sharir, and R.E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2:209–233, 1987.

[GM90] S.K. Ghosh and A. Maheshwari. An optimal algorithm for computing a minimum nested nonconvex polygon. *Inform. Process. Lett.*, 36:277–280, 1990.

[Her91] J. Hershberger. A new data structure for shortest path queries in a simple polygon. *Inform. Process. Lett.*, 38:231–235, 1991.

[HM85] S. Hertel and K. Mehlhorn. Fast triangulation of the plane with respect to simple polygons. *Inform. Control*, 64:52–76, 1985.

[HS91] J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. In *Proc. 2nd Workshop Algorithms Data Struct.*, volume 519 of *Lecture Notes in Comput. Sci.*, pages 331–342. Springer-Verlag, New York, 1991.

[HS93] J. Hershberger and S. Suri. Efficient computation of Euclidean shortest paths in the plane. In *Proc. 34th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 508–517, 1993.

[HS95] J. Hershberger, and S. Suri. Morphing Binary Trees. *Proc. 6th Annu. Sympos. Discrete Algorithms*, pages 396–404, 1995.

[IK95] C. Icking, and R. Klein. Searching for the kernel of a polygon—A competitive strategy. *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 258–266, 1995.

[Kei85] J.M. Keil. Decomposing a polygon into simpler components. *SIAM J. Comput.*, 14:799–817, 1985.

[LM84] C. Lantuejoul, and F. Maisonneuve. Geodesic methods in quantitative image analysis. *Pattern Recogn.*, 17:177–187, 1984.

[MS90] E.A. Melissaratos and D.L. Souvaine. On solving geometric optimization problems using shortest paths. In *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, pages 350–359, 1990.

[MS92] J.S.B. Mitchell and S. Suri. Separation and approximation of polyhedral surfaces. In *Proc. 3rd ACM-SIAM Sympos. Discrete Algorithms*, pages 296–306, 1992.

[MS95] J.S.B. Mitchell, and S. Suri. Geometric algorithms. In M.O. Ball, T.L. Magnati, C.L. Monma, and G.L. Nemhauser, editors, *Handbook of Operations Research/Management Science*, pages 425–479. Elsevier, Amsterdam, 1995.

[O'R87] J. O'Rourke. *Art Gallery Theorems and Algorithms.* Oxford University Press, New York, 1987.

[Sei91] R. Seidel. A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons. *Comput. Geom. Theory Appl.*, 1:51–64, 1991.

[Sur87] S. Suri. *Minimum link paths in polygons and related problems.* Ph.D. thesis, Dept. Comput. Sci., Johns Hopkins Univ., Baltimore, 1987.

[Sur90] S. Suri. On some link distance problems in a simple polygon. *IEEE Trans. Robot. Autom.*, 6:108–113, 1990.

24 SHORTEST PATHS AND NETWORKS

Joseph S.B. Mitchell

INTRODUCTION

Computing an optimal path in a geometric domain is a fundamental problem in computational geometry, with applications in robotics, geographic information systems (GIS), wire routing, etc.

A taxonomy of shortest-path problems arises from several parameters that define the problem:

1. Objective function (the metric for length).
2. Constraints on the path ("move from s to t," visit a set of points/regions, etc.).
3. Input geometry (the "map"); e.g., types of obstacles allowed.
4. Type of moving object (single point vs. more complex robot geometry).
5. Dimension of the problem (2, 3, or higher).
6. Single shot vs. repetitive mode queries.
7. Static vs. dynamic environments (e.g., allowing insertions/deletions of obstacles, or allow moving obstacles).
8. Exact vs. approximate algorithms.
9. Known vs. unknown map; e.g., discovery of obstacles on-line, using a line-of-sight sensor.

We survey various forms of the problem, primarily in two dimensions, for motion of a single point, since most results have focused on these cases. We discuss shortest paths in a simple polygon (Section 24.1), shortest paths among obstacles (Section 24.2), other metrics for length (Section 24.3), other path/network optimization problems (Section 24.4), and higher dimensions (Section 24.5).

GLOSSARY

Polygonal s-t path: A path from point s to point t consisting of a finite number of line segments (***edges***, or ***links***) joining a sequence of points (***vertices***).

Length of a path: A nonnegative number associated with a path, measuring its total cost according to some prescribed metric. Unless otherwise specified, the length will be the Euclidean length of the path.

Shortest/optimal/geodesic path: A path of minimum length among all paths that are feasible (satisfying all imposed constraints). See Figure 24.0.1.

Shortest-path distance: The metric induced by a shortest-path problem. The shortest-path distance between s and t is the length of a shortest s-t path; in many geometric contexts, it is also referred to as ***geodesic distance.***

Locally shortest/optimal path: A path that cannot be improved by making a small change to it that preserves its ***combinatorial structure***; also known as a ***taut-string*** path in the case of a shortest obstacle-avoiding path.

Simple polygon P of n vertices: A closed, simply-connected region whose boundary is a union of n (straight) line segments (edges), whose endpoints are the vertices of P.

Polygonal domain of n vertices and h holes: A closed, multiply-connected region whose boundary is a union of n line segments, forming $h+1$ closed (polygonal) cycles. A simple polygon is a polygonal domain with $h = 0$.

Triangulation of a simple polygon P: A decomposition of P into triangles such that any two triangles intersect in either a common vertex, a common edge, or not at all. A triangulation of P can be computed in $O(n)$ time. See Section 23.2.

Obstacle: A region of space whose interior is forbidden to paths. The complement of the set of obstacles is the ***free space.*** If the free space is a polygonal domain P, the obstacles are the $h+1$ connected components (h ***holes***, plus the ***face at infinity***) of the complement of P.

Visibility graph $\mathbf{VG}(P)$: A graph whose nodes are the vertices of P and whose edges join pairs of nodes for which the corresponding segment lies inside P. See Chapter 25. An example is shown in Figure 24.0.2.

Single-source query: A query that specifies a goal point t, and requests the length of a shortest path from a *fixed* source point s to t. The query may also require the retrieval of an actual instance of a shortest s-t path; in general, this can be reported in additional time $O(k)$, where k is the complexity of the output (e.g., the number of edges).

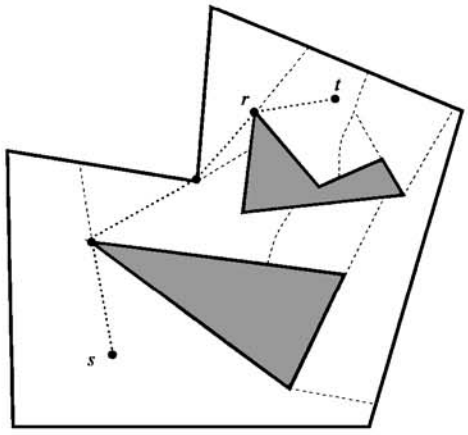


FIGURE 24.0.1
A shortest path map with respect to source point s within a polygonal domain. The dotted path indicates the shortest s-t path, which reaches t via the root r of its cell.

Shortest path map, $\mathbf{SPM}(s)$: A decomposition of free space into regions (***cells***) according to the "combinatorial structure" of shortest paths from a fixed source point s to points in the regions. Specifically, for shortest paths in a polygonal domain, SPM(s) is a decomposition of P into cells such that for all points t interior to a cell, the sequence of obstacle vertices along an s-t path is fixed. In

particular, the *last* obstacle vertex along a shortest s-t path is the ***root*** of the cell containing t. Each cell is ***star-shaped*** with respect to its root, which lies on the boundary of the cell. See Figure 24.0.1, where the root of the cell containing t is labeled r. If SPM(s) is preprocessed for point location (see Chapter 30), then single-source queries can be answered efficiently by locating the query point t within the decomposition.

Two-point query: A query that specifies two points, s and t, and requests the length of a shortest path between them.

Geodesic Voronoi diagram* (VD):** A Voronoi diagram for a set of ***sites, in which the underlying metric is the geodesic distance. See Chapters 20 and 22.

Geodesic center of P: A point within P that minimizes the maximum of the shortest-path lengths to any other point in P.

Geodesic diameter of P: The length of a longest shortest path between a pair of vertices of P.

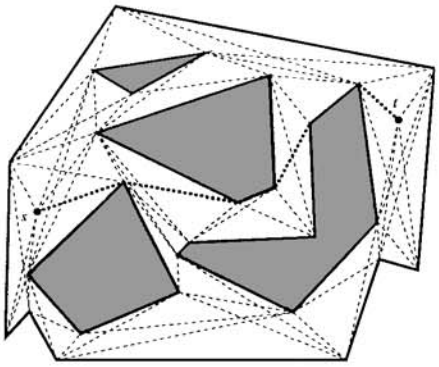


FIGURE 24.0.2
The visibility graph VG(P). Edges of VG(P) are of two types: (1) *the heavy dark boundary edges of P, and* (2) *the edges that intersect the interior of P, shown with thin dashed segments. A shortest s-t path is highlighted.*

24.1 PATHS IN A SIMPLE POLYGON

The most basic geometric shortest-path problem is to find a shortest path inside a ***simple*** polygon P (having no holes), connecting two points, s and t. The complement of P serves as an obstacle through which the path is not allowed to travel. In this case, there is a unique taut-string path from s to t, since there is only one way to "thread" a string through a simply-connected region.

Algorithms for computing a shortest s-t path begin with a triangulation of P ($O(n)$ time; Section 23.2), whose dual graph is a tree. The ***sleeve*** is comprised of the triangles that correspond to the (unique) path in the dual that joins the triangle containing s to that containing t. By considering the effect of adding the triangles in order along the sleeve, it is not hard to obtain an $O(n)$ time algorithm for collapsing the sleeve into a shortest path. At a generic step of the algorithm, the sleeve has been collapsed to a structure called a ***funnel*** (with *base* ab and *root* r) consisting of the shortest path from s to a vertex r, and two (concave) shortest paths joining r to the endpoints of the segment ab that bounds the triangle abc processed next (see Figure 24.1.1). In adding triangle abc, we "split" the funnel in two according to the taut-string path from r to c, which will, in general, include a segment uc joining c to some (vertex) point of tangency u, along one of the two concave chains of the funnel. After the split, we keep that funnel (with base ac

or bc) that contains the s-t taut-string path. The work needed to search for u can easily be charged off to those vertices that are discarded from further consideration. The end result is that a shortest s-t path is found in time $O(n)$, which is worst-case optimal.

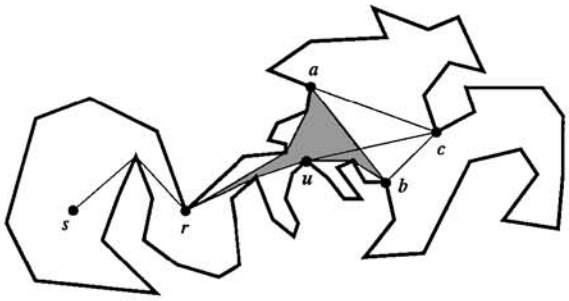


FIGURE 24.1.1
Splitting a funnel.

SHORTEST PATH MAPS

The shortest path map SPM(s) for a simple polygon has a particularly simple structure, as the boundaries between cells in the map are simply (line segment) chords of P obtained by extending appropriate edges of the visibility graph VG(P). It can be computed in time $O(n)$ by using somewhat more sophisticated data structures to do funnel splitting efficiently (since, in this case, we cannot discard one side of each split funnel). Single-source queries can be answered in $O(\log n)$ time, after storing the SPM(s) in an appropriate $O(n)$-size point location data structure (see Chapter 30).

DYNAMIC VERSION

In the dynamic version of the problem, one allows the polygon P to change with addition and deletion of edges and vertices. If the changes are always made in such a way that the set of all edges yields a ***connected planar subdivision*** of the plane into simple polygons (i.e., no "islands" are created), then one can maintain a data structure of size $O(n)$ that supports two-point query time of $O(\log^2 n)$ (plus $O(k)$ if the path is to be reported), and update time of $O(\log^2 n)$ for each addition/deletion of an edge/vertex [GT93].

OTHER RESULTS

Several other problems have been studied with respect to geodesic distances induced by a simple polygon and are summarized in Table 24.1.1. See also Table 23.4.1.

OPEN PROBLEMS

1. Can one devise a simple $O(n)$ time algorithm for computing the shortest path between two points in a simple polygon, *without* resorting to a (complicated) linear-time triangulation algorithm?

TABLE 24.1.1 Shortest paths and geodesic distance in simple polygons.

PROBLEM VERSION	COMPLEXITY	NOTES
Shortest s-t path	$O(n)$	
Single-source query; SPM(s)	$O(\log n)$ query $O(n)$ preproc/space	builds SPM(s) in $O(n)$ time
Two-point query	$O(\log n)$ query $O(n)$ preproc/space	
Dynamic two-point query	$O(\log^2 n)$ update/query $O(n)$ space	
Parallel algorithm (CREW PRAM)	$O(\log n)$ time $O(n/\log n)$ processors	in triangulated polygon also builds SPM(s)
Geodesic VD	$O((n+k)\log(n+k))$	k point sites
All nearest neighbors	$O(n)$	for set of vertices
Geodesic farthest-site VD	$O((n+k)\log(n+k))$ time $O(n+k)$ space	k point sites
All farthest neighbors	$O(n)$	for set of vertices

2. Can the geodesic Voronoi diagram for k sites within P be computed in time $O(n + k \log k)$?

3. Can the geodesic center of a simple polygon be computed in $O(n)$ time? What can be said in the ***bit complexity*** (vs. real RAM) model of computation?

24.2 PATHS IN A POLYGONAL DOMAIN

While in a simple polygon there is a unique taut-string path between two points, in a general polygonal domain P, there can be an exponential number of taut-string simple paths between two points.

The homotopy type of a path can be expressed as a sequence (with repetitions) of triangles visited, for some triangulation of P. For any given homotopy type, expressed with N triangles, a shortest (taut) path of that type can be computed in $O(N)$ time [HS94].

SEARCHING THE VISIBILITY GRAPH

Without loss of generality, we can assume that s and t are vertices of P (since we can make "point" holes in P at s and t). It is easy to show that any locally optimal s-t path must lie on the visibility graph VG(P) (Figure 24.0.2). We can construct VG(P) in output-sensitive time $O(E_{\mathrm{VG}} + n \log n)$, where E_{VG} denotes the number of edges of VG(P), even if we allow only $O(n)$ working space. Given the graph VG(P), whose edges are weighted by their Euclidean lengths, we can use Dijsktra's algorithm to construct a tree of shortest paths from s to all vertices of P, in time $O(E_{\mathrm{VG}} + n \log n)$. Thus, Euclidean shortest paths among obstacles in the plane can be computed in time $O(E_{\mathrm{VG}} + n \log n)$. This bound is worst-case quadratic in n, since $E_{\mathrm{VG}} \leq \binom{n}{2}$; note too that domains exist with $E_{\mathrm{VG}} = \Omega(n^2)$.

Given the tree of shortest paths from s, we can compute SPM(s) in time $O(n \log n)$.

CONTINUOUS DIJKSTRA METHOD

Instead of searching the visibility graph (which may have quadratic size), an alternative paradigm for shortest-path problems is to construct the (linear-size) shortest path map directly. The ***continuous Dijkstra*** method was developed for this purpose.

Building on the success of the method in solving (in nearly linear time) the shortest-path problem for the L_1 metric, Mitchell [Mit96a] developed a version of the continuous Dijkstra method applicable to the Euclidean shortest-path problem, obtaining the first subquadratic ($O(n^{1.5+\epsilon})$) time bound. Subsequently, this result was improved by Hershberger and Suri [HS95], who achieve a nearly optimal algorithm based also on the continuous Dijkstra method. They give an $O(n \log n)$ time and $O(n \log n)$ space algorithm, coming close to the lower bounds of $\Omega(n + h \log h)$ time and $O(n)$ space.

The continuous Dijkstra paradigm involves simulating the effect of a wavefront propagating out from the source point, s. The ***wavefront*** at distance δ from s is the set of all points of P that are at geodesic distance δ from s. It consists of a set of curve pieces, called ***wavelets***, which are arcs of circles centered at obstacle vertices that have already been reached. At certain critical "events," the structure of the wavefront changes due to one of the following possibilities:

(1) a wavelet disappears (due to the closure of a cell of the SPM);

(2) a wavelet collides with an obstacle vertex;

(3) a wavelet collides with another wavelet; or

(4) a wavelet collides with an obstacle edge at a point interior to that edge.

It is not difficult to see from the fact that SPM(s) has linear size, that the total number of such events is $O(n)$. The challenge in applying this propagation scheme is devising an efficient method to know *what* events are going to occur and in being able to *process* each event as it occurs (updating the combinatorial structure of the wavefront).

One approach, used in [Mit96a], is to track a "pseudo-wavefront," which is allowed to run over itself, and to "clip" only when a wavelet collides with a vertex that has already been labeled due to an earlier event. Detection of when a wavelet collides with a vertex is accomplished with range-searching techniques. An alternative approach, used in [HS95], simplifies the problem by first decomposing the domain P using a *conforming subdivision*, which allows one to propagate an approximate wavefront on a cell-by-cell basis. A key property of a conforming subdivision is that any edge of length L of the subdivision has only a constant number of (constant-sized) cells within geodesic distance L.

OTHER RESULTS

Table 24.2.1 summarizes the results discussed so far, and lists some approximation results, especially for the two-point query problem.

TABLE 24.2.1 Shortest paths among planar obstacles, in a polygonal domain.

PROBLEM	COMPLEXITY	NOTES
Shortest s-t path	$O(n \log n)$	$O(n \log n)$ space
	$O(n + h^2 \log n)$	$O(n)$ space
	$O(n^{1.5+\epsilon})$	$O(n)$ space
Approx shortest s-t path	$O((n \log n)/\epsilon)$	$O(n/\epsilon)$ space
	$O((n \log n)/\sqrt{\epsilon})$	$O(n/\sqrt{\epsilon})$ space
SPM(s)/geodesic VD	$O(n \log n)$	$O(n \log n)$ space
	$O(n^{1.5+\epsilon})$	$O(n)$ space
Approx two-point query	$O(n \log n)$ query	$(1+\epsilon)$-approx
	$O(n)$ space	
	$O(n \log n)$ preproc	
Approx two-point query	$O(\log n)$ query	$(1+\epsilon)$-approx
	$O(n^2)$ space	
	$O(n^2 \log n)$ preproc	
Approx two-point query	$O(\log n)$ query	$(3\sqrt{2}+\epsilon)$-approx
	$O(n \log n)$ space	
	$O(n^{3/2}/\log^{1/2} n)$ preproc	

OPEN PROBLEMS

1. Can the Euclidean shortest-path problem be solved in $O(n+h \log h)$ time and $O(n)$ space?
2. How efficiently, and using what size data structure, can one preprocess a polygonal domain for exact two-point queries?
3. How efficiently can one compute a geodesic center/diameter for a polygonal domain?
4. How efficiently can one solve Euclidean problems (shortest paths, geodesic center/diameter) in the bit complexity (rather than the real RAM) model of computation?

24.3 OTHER METRICS FOR LENGTH

In the problems considered so far, the Euclidean metric has been used to measure the length of a path. We consider now several other possible objective functions for measuring path length. Tables 24.3.1 and 24.3.2 summarize results.

GLOSSARY

L_p metric: The L_p distance between $q = (q_x, q_y)$ and $r = (r_x, r_y)$ is given by $d_p(q,r) = [|q_x - r_x|^p + |q_y - r_y|^p]^{1/p}$. The L_p length of a polygonal path is the sum of the L_p lengths of each edge of the path. Special cases of the L_p metric

include the L_1 metric (***Manhattan metric***) and the L_∞ metric ($d_\infty(q,r) = \max\{|q_x - r_x|, |q_y - r_y|\}$).

Rectilinear path: A polygonal path with each edge parallel to a coordinate axis; also known as an ***isothetic*** path.

C-oriented path: A polygonal path with each edge parallel to one of a set C of $c = |C|$ ***fixed orientations***.

Link distance: The minimum number of edges from s to t within a polygonal domain P. If the paths are restricted to be rectilinear or C-oriented, then we obtain the ***rectilinear link distance*** or ***C-oriented link distance***.

Min-link s-t path: A polygonal path from s to t that achieves the link distance.

Weighted region problem: Given a piecewise-constant function $f : \mathbb{R}^2 \to \mathbb{R}$ that is defined by assigning a nonnegative ***weight*** to each face of a given triangulation in the plane. The ***weighted length*** of an s-t path π is the path integral, $\int_\pi f(x,y)d\sigma$, of the weight function along π. The ***weighted region metric*** associated with f defines the distance $d_f(s,t)$ to be the infimum over all s-t paths π of the weighted length of π. The ***weighted region problem*** (WRP) asks for an s-t path of minimum weighted length.

Sailor's problem: Compute a minimum-cost path, where the cost of motion is *direction-dependent*, and there is a cost L per turn (in a polygonal path).

Bounded curvature shortest-path problem: Compute a shortest obstacle-avoiding smooth (C^1) path joining point s, with prescribed velocity orientation, to point t, with prescribed velocity orientation, such that at each point of the path the radius of curvature is at least 1.

Maximum concealment path: A path within polygonal domain P that minimizes the length during which the robot is exposed to a given set of "enemy" observers. This problem is a special case of the weighted region problem, in which weights are 0 (for travel in concealed free space), 1 (for travel in exposed free space), or ∞ (for travel through obstacles).

Total turn for an s-t path: The sum of the absolute values of all turn angles for a polygonal s-t path.

Minimum-time path problem: Find a path to minimize the total time required to move from an initial position, at an initial velocity, to a goal position and velocity, subject to bounds on the allowed acceleration and velocity along the path. This problem is also known as the ***kinodynamic motion planning problem***.

LINK DISTANCE

In the min-link path problem, our goal is to minimize the number of links (and hence the number of turns) in a path connecting s and t. In many problems, the link distance provides a more natural measure of path complexity than the Euclidean length, as well as having applications to curve simplification (e.g., [MS95b]).

In a simple polygon P, a min-link path can be computed in time $O(n)$, as described in Section 23.4. In fact, in time $O(n)$ a ***window partition*** of P with respect to a point s can be computed, after which a min-link path from s to t can be reported in time proportional to the link distance. The algorithm described in

TABLE 24.3.1 Link distance shortest-path problems.

PROBLEM	COMPLEXITY	NOTES
Min link path	$O(E_{VG}\alpha^2(n)\log n)$	polygonal domain
Rectilinear link path	$O(n\log n)$ time, $O(n)$ space	rectilinear obstacles
Rectilinear link path	$O(n)$	rectilinear simple polygon
C-oriented link path	$O(c^2 n\log n)$ time	C-oriented obstacles
	$O(c^2 n\log n)$ space, preproc	builds SPM(s)
Two-point link query	$O(\log n)$ query	simple polygon
	$O(n^3)$ space, preproc	
Two-point rectilinear link query	$O(\log n)$ query	rectilinear simple polygon
	$O(n\log n)$ space, preproc	also is L_1-opt
Shortest k-link path	$O(n^3k^3\log(Nk/\epsilon^{1/k}))$	simple polygon

Section 23.4 computes the partition via "staged illumination," essentially a form of the continuous Dijkstra method under the link distance metric.

In a polygonal domain with holes, min-link paths can also be computed using a staged illumination method, but the algorithm is not simple: it relies on efficient methods for computing a single face in an arrangement of line segments (see Chapter 21). A min-link s-t path can be computed in time $O(E_{VG}\alpha^2(n)\log n)$, where $\alpha(n)$ is the inverse Ackermann function (Section 40.4). If we consider C-oriented and rectilinear link distance, then some better time/space bounds are possible, and some of these apply also to combined metrics, in which there is a cost for length as well as links.

Refer to Table 24.3.1 for many related results on link distance, including rectilinear link distance, and on two-point queries. See also Table 23.4.2 for more results specifically on simple polygons.

L_1 METRIC

Instead of measuring path length according to the L_2 (Euclidean) metric, consider the problem of computing shortest paths in a polygonal domain P that are short according to the L_1 metric.

A method based on visibility graph principles allows one to construct a sparse graph (with $O(n\log n)$ nodes and edges) that is ***path-preserving*** in that it is guaranteed to contain a shortest path between any two vertices. Applying Dijkstra's algorithm then gives an $O(n\log^{1.5} n)$ time ($O(n\log n)$ space) algorithm for L_1-shortest paths.

A method based on the continuous Dijkstra paradigm allows the SPM(s) to be constructed in time $O(n\log n)$, using $O(n)$ space [Mit92]. The special property of the L_1 metric that is exploited in this algorithm is the fact that the wavefront in this case is piecewise-linear, with wavelets that are line segments of slope ± 1, so that the first vertex hit by a wavelet can be determined by rectangular range searching techniques (see Chapter 31).

Methods for finding L_1-shortest paths generalize to the case of C-oriented paths, in which $c = |C|$ fixed directions are given. Shortest C-oriented paths can be computed in time $O(cn\log n)$. Since the Euclidean metric is approximated to within accuracy $O(1/c^2)$ if we use c equally-spaced orientations, this results in

TABLE 24.3.2 Shortest paths in other metrics.

PROBLEM	COMPLEXITY	NOTES
L_1-shortest path, SPM(s)	$O(n \log n)$	polygonal domain
L_1 two-point query	$O(\log^2 n)$ query	polygonal domain
	$O(n^2 \log n)$ space	
	$O(n^2 \log^2 n)$ preproc	
L_1 two-point query	$O(\log n)$ query	rectangle obstacles
	$O(n^2)$ space, preproc	
L_1 two-point query	$O(\sqrt{n})$ query	rectangle obstacles
	$O(n^{1.5})$ space, preproc	
L_1 approx two-point query	$O(\log n)$ query	3-approx
	$O(n \log n)$ space	rectangle obstacles
	$O(n \log^2 n)$ preproc	
Weighted region problem	$O(n^8 L)$	$(1+\epsilon)$-approx
	$L = O(\log \frac{nNW}{\epsilon})$	
Weighted region problem	$O(n^2)$	region weights 0, 1, ∞
L_1 weighted region problem	$O(n \log^{3/2} n)$	rectilinear regions
	$O(n \log n)$ space, $O(\log n)$ query	single-source queries
L_1 WRP, two-point query	$O(\log^2 n)$ query	rectilinear regions
	$O(n^2 \log^2 n)$ space, preproc	
Bounded curvature shortest path	$O(n^4 \log n)$	moderate obstacles
Sailor's problem ($L = 0$)	$O(n^2)$	polygonal domain
Sailor's problem ($L > 0$)	poly(n, ϵ)	ϵ-approx
Max concealment, simple polygon	$O(v^2(v+n)^2)$	v viewpoints
Max concealment, polygonal domain	$O(v^4 n^4)$	v viewpoints
Min total turn	$O(E_{VG} \log n)$	polygonal domain

an algorithm that computes, in time $O((n/\sqrt{\epsilon}) \log n)$, a path guaranteed to have length within a factor $(1+\epsilon)$ of the Euclidean shortest path length.

WEIGHTED REGION METRIC

The weighted region problem (WRP) seeks an optimal s-t path according to the weighted region metric d_f induced by a given piecewise-constant weight function f. This problem is a natural generalization of the shortest-path problem in a polygonal domain: consider a weight function that assigns weight 1 to P and weight ∞ (or a sufficiently large constant) to the obstacles (the complement of P).

The weighted region problem models the minimum-time path problem for a point robot moving in a terrain of varied types (e.g., grassland, brushland, blacktop, bodies of water, etc.), where each type of terrain has an assigned weight equal to the reciprocal of the maximum speed of traversal for the robot.

Assume that f is specified by a triangulation having n vertices, with each face assigned an integer weight $\alpha \in \{0, 1, \ldots, W, +\infty\}$. (We can allow each edge of the triangulation to have a weight that is possibly distinct from that of the triangular facets on either side of it; in this way, linear features such as roads can be modeled.) Using an algorithm based on the continuous Dijkstra method, one can find a path whose weighted length is guaranteed to be within a factor $(1+\epsilon)$ of optimal, where $\epsilon > 0$ is any user-specified degree of precision. The time

complexity of the algorithm is $O(E \cdot S)$, where E is the number of "events" in the continuous Dijkstra algorithm, and S is the complexity of performing a numerical search to solve the following subproblem: Find a $(1+\epsilon)$-shortest path from s to t that goes through a given sequence of k edges of the triangulation. It is known that $E = \Theta(n^4)$. The numerical search can be accomplished using a form of binary search that exploits the local optimality condition: An optimal path bends according to *Snell's Law of Refraction* when crossing a region boundary. This leads to a bound of $S = O(k^2 \log(nNW/\epsilon))$ on the time needed to perform a search on a k-edge sequence, where N is the largest coordinate of any vertex of the triangulation (and all coordinates are integers). Since one can show that $k = O(n^2)$, this yields an overall time bound of $O(n^8 L)$, where $L = \log(nNW/\epsilon)$ can be thought of as the bit complexity of the problem instance.

Various special cases of the weighted region problem admit faster and simpler algorithms. For example, if the weighted subdivision is rectilinear, and path length is measured according to weighted L_1 length, then efficient algorithms for single-source and two-point queries can be based on searching a path-preserving graph [CKT95]. Similarly, if the region weights are restricted to $\{0, 1, \infty\}$ (while edges may have arbitrary (nonnegative) weights), then an $O(n^2)$ algorithm can be based on constructing a path-preserving graph similar to a visibility graph. This also leads to an efficient method for performing ***lexicographic*** optimization, in which one prioritizes various types of regions according to which is most important for path length minimization.

MINIMUM-TIME PATHS

The minimum-time path problem is a difficult optimal control problem; optimal paths will be complicated curves given by solutions to differential equations.

As a first step towards understanding the algorithmic complexity of computing minimum-time paths under dynamic constraints, there is a polynomial-time $(1+\epsilon)$-approximation algorithm. The approach is to discretize the four-dimensional phase space that represents position and velocity, with special care to ensure that the size of the grid is bounded by a polynomial in $1/\epsilon$ and n and that the shortest paths obtained from the resulting graph are guaranteed to be close to optimal.

If there is an upper bound on the L_∞ norm of the velocity and acceleration vectors, one can obtain an *exact*, exponential-time, polynomial-space algorithm, based on characterizing a set of "canonical solutions" (related to "bang-bang" controls) that are guaranteed to include an optimal solution path. This leads to an expression in the first-order theory of the reals, which can be solved exactly; see Chapter 29. However, it remains an open question whether or not a polynomial-time algorithm exists.

A closely related shortest-path problem is the ***bounded curvature shortest-path problem***, in which we require that no point of the path have a radius of curvature less than 1. For this problem, $(1+\epsilon)$-approximation algorithms are known, with polynomial $(O(\frac{n^2}{\epsilon^2} \log n))$ running time [WA96], but the complexity of solving the problem exactly remains open. For the special case in which the obstacles are "moderate" (have differentiable boundary curves, with radius of curvature at least 1), both an approximation algorithm and, most recently, an exact $O(n^4 \log n)$ algorithm have been found [BL96].

OPTIMAL ROBOT MOTION

So far, we have considered only the problem of optimally moving a *point* robot. If the robot is modeled as a circle, or as a nonrotating polygon, then many of the results carry over by simply applying the standard ***configuration space*** approach in motion planning (see Chapters 40 and 41): "shrink" the robot to a (reference) point, and "grow" the obstacles (using a Minkowski sum) so that the complement of the grown obstacles models the region of the plane for which there is no collision with an obstacle if the robot has its reference point placed there.

Optimal motion of *rotating* noncircular robots is a much harder problem. Even the simplest case, of moving a (unit) line segment (a ***ladder***) in the plane, is highly nontrivial. One notion of optimal motion requires that we minimize the average distance traveled by a set of k fixed points, evenly distributed along the ladder. This "d_k-distance" in fact defines a metric (for $k \geq 2$). The special case of $k = 2$ is the well-known ***Ulam's problem***, for which optimal motions are fully characterized in the absence of obstacles [IRWY92]. The case of $k = \infty$ is an especially interesting case, requiring that we compute a minimum *work* motion of a ladder; however, no results are known for this problem. (The work measures the integral (over $\lambda \in [0, 1]$) of the path length, $L(\lambda)$, for each infinitesimal subsegment of length $d\lambda$.) While d_1 does not define a metric, several cases of d_1-motion, and its generalization of measuring the distance traveled by any fixed "focus" F on the ladder, have been studied. In particular, if F is restricted to move on the visibility graph of a polygonal environment, polynomial-time algorithms are known. Without restrictions, minimizing the d_1-distance (for any F not at an endpoint of the ladder) is NP-hard, but there exists an approximation algorithm.

MULTIPLE CRITERION OPTIMAL PATHS

The standard shortest-path problem asks for paths that minimize some *one* objective (length) function. Frequently, however, an application requires us to find paths to minimize *two or more* objectives; the resulting problem is a ***bicriterion*** (or ***multi-criterion***) shortest-path problem. A path is called ***efficient*** or ***Pareto optimal*** if no other path has a better value for one criterion without having a worse value for the other criterion.

Multi-criterion optimization problems tend to be hard. Even the bicriterion path problem in a graph is NP-hard: Does there exist a path from s to t whose length is less than L and whose weight is less than W? Pseudo-polynomial-time algorithms are known, and many heuristics have been devised.

In geometric problems, various optimality criteria are of interest, including any pair from the following list: Euclidean (L_2) length, rectilinear (L_1) length, other L_p metrics, link distance, total turn, and so on.

NP-hardness lower bounds are known for several versions, including [AMP91]: (1) Find a path in a polygonal domain whose L_2 length is at most L, and whose total turn is at most T; (2) Find a path in a polygonal domain whose L_p length is at most λ_p and whose L_q length is at most λ_q ($p \neq q$); and (3) Given a subdivision of the plane into red and blue polygonal regions, find a path whose length within blue regions is at most B and whose length within red regions is at most R.

One problem of particular interest is to compute a Euclidean shortest path

within a polygonal domain, constrained to have at most k links. No exact solution is currently known for this problem. Part of the difficulty is that a minimum-link path will not, in general, lie on the visibility graph (or on any simple discrete graph). Furthermore, the computation of the turn points of such an optimal path appears to require the solution to high-degree polynomials.

For a given k ($k \geq d_L$, where d_L is the s-t link distance), one can compute a path in a simple polygon P whose length is guaranteed to be within a factor $(1+\epsilon)$ of the length of a shortest k-link path, for any tolerance $\epsilon > 0$. The algorithm runs in time $O(n^3k^3 \log(Nk/\epsilon^{1/k}))$, polynomial in n and k, and logarithmic in $1/\epsilon$ and the largest integer coordinate N of any vertex of P. Within the same time bound, one can compute an ϵ-optimal path under any (single) *combined* objective, $f(L, G)$, where L and G denote link distance and Euclidean length, respectively, and f is an increasing function in G for each L.

Does there exist an s-t path that is *simultaneously* close to Euclidean shortest and minimum-link? In a *simple* polygon, one can always find an s-t path whose link length is within a factor of 2 of the link distance from s to t, while also having Euclidean length within a factor of $\sqrt{2}$ of the Euclidean shortest-path length. A corresponding result is not possible for polygons with holes. However, in $O(kE_{\mathrm{VG}}^2)$ time, one can compute a path in a polygonal domain having at most $2k$ links and length at most that of a shortest k-link path.

In a rectilinear polygonal domain, efficient algorithms are known for the bicriterion path problem that combines *rectilinear* link distance and L_1 length [LYW96]. For example, efficient algorithms are known in two or more dimensions for computing optimal paths according to a *combined metric*, defined to be a linear combination of rectilinear link distance and L_1 path length. (Note that this is not the same as computing the Pareto-optimal solutions.) There are $O(n \log^2 n)$ algorithms for computing a shortest k-bend path, a minimum-bend shortest path, or any combined objective that uses a monotonic function of rectilinear link length and L_1 length in a rectilinear polygonal domain. In all of these rectilinear problems, there is an underlying grid graph which can serve as a path-preserving graph.

OPEN PROBLEMS

1. Can a minimum-link path in a polygonal domain be computed in subquadratic time? The only lower bound known is $\Omega(n \log n)$.
2. What is the smallest size data structure for a simple polygon P that allows logarithmic-time two-point link distance queries?
3. For a polygonal domain (with holes), what is the complexity of computing a shortest k-link path between two given points?
4. What is the complexity of the ladder problem for a polygonal domain, in which the cost of motion is the total work involved in translation/rotation?
5. Is it NP-hard to minimize the d_1-distance of a ladder endpoint?
6. What is the complexity of the bounded curvature shortest-path problem?

24.4 OTHER NETWORK OPTIMIZATION PROBLEMS

All of the problems considered so far involved computing a shortest path from one point to another (or from one point to all other points). We consider now some other network optimization problems, in which the objective is to compute a shortest path, cycle, tree, or other graph, subject to various constraints. A summary of results is given in Table 24.4.1.

GLOSSARY

Minimum spanning tree* (MST) *of S: A tree of minimum total length whose nodes are a given set S of n points, and whose edges are line segments joining pairs of points.

Minimum Steiner spanning tree (Steiner tree) of S: A tree of minimum total length whose nodes are a superset of a given set S of n points, and whose edges are line segments joining pairs of points. Those nodes that are not points of S are called ***Steiner points***.

***k-minimum spanning tree* (k-MST):** A minimum-length tree that spans some subset of $k \leq n$ points of S.

***Traveling salesman problem* (TSP):** Find a shortest cycle that visits every point of a set S of n points.

***MAX* TSP:** Find a *longest* cycle that visits every point of a set S of n points.

***Min/max-area* TSP:** Find a cycle on a given set S of points such that the cycle defines a simple polygon of minimum/maximum area.

TSP *with neighborhoods:* Find a shortest cycle that visits at least one point in each of a set of n neighborhoods (e.g., polygons).

Watchman route (path) problem: Find a shortest cycle (path) within a polygonal domain P such that every point of P is visible from some point of the cycle.

Lawnmowing problem: Find a shortest cycle (path) for the center of a disk (a "lawnmower" or "cutter") such that every point of a given (possibly disconnected) region is covered by the disk at some position along the cycle (path).

Milling problem: Similar to the lawnmowing problem, but with the constraint that the cutter must at all times remain inside the given region (the "pocket" to be milled).

Zookeeper's problem: Find a shortest cycle in a simple polygon P (the ***zoo***) through a given vertex v such that the cycle visits every one of a set of k disjoint convex polygons (***cages***), each sharing an edge with P.

Aquarium-keeper's problem: Find a shortest cycle in a simple polygon P (the ***aquarium***) such that the cycle touches every edge of P.

Red-blue separation problem: Find a minimum-length simple polygon that separates a set of "red" points from a set of "blue" points.

***Relative convex hull* of point set S within simple polygon P:** The shortest cycle within P that surrounds S. The relative convex hull is necessarily a simple polygon, with vertices among the points of S and the vertices of P.

TABLE 24.4.1 Other optimal path/cycle/network problems.

PROBLEM	COMPLEXITY	NOTES
Minimum spanning tree (MST)	$O(n \log n)$	uses Delaunay triangulation
Steiner tree	NP-hard	
Steiner tree	$O(n \log n)$	$\frac{2}{\sqrt{3}}$-approx, using MST
Steiner tree	$O(n^{O(1/\epsilon)})$	$(1+\epsilon)$-approx
k-MST	NP-hard	
k-MST	$O(n^{O(1/\epsilon)})$	$(1+\epsilon)$-approx
Traveling salesman problem (TSP)	NP-hard	
Traveling salesman problem (TSP)	$O(n^{O(1/\epsilon)})$	$(1+\epsilon)$-approx
MAX TSP	open	$(1+\epsilon)$-approx
Min area TSP	NP-complete	
Max area TSP	NP-complete	$(1/2)$-approx
TSP with neighborhoods (restricted)	NP-hard	$O(1)$-approx
TSP with neighborhoods (general)	NP-hard	$O(\log n)$-approx
Watchman route in simple polygon	$O(n^4)$	
Watchman route in simple polygon	$O(n^2)$	through a *given* point
Watchman route in rectilinear simple polygon	$O(n)$	
Watchman path in simple polygon	$O(n^{12})$	
Min link watchman in simple polygon	NP-hard	$O(1)$-approx
Watchman route in polygonal domain	NP-hard	$O(\log k)$-approx
Min link watchman in polygonal domain	NP-hard	$O(\log n)$-approx
Lawnmowing problem	NP-hard	$O(1)$-approx
Milling problem in simple polygon	open	$O(1)$-approx
Milling problem in polygonal domain	NP-hard	$O(1)$-approx
Shortest postman path	$O(n^3)$	simple polygon
Simple s-t Hamiltonian path	$O(n^2m(n+m))$	m points in simple n-gon
Simple s-t Hamiltonian path	NP-Complete	in polygonal domain
Aquarium-keeper's problem	$O(n)$	simple polygon
Zookeeper's problem	$O(n \log^2 n)$	simple polygon
Red-blue separation	$O(n^5)$	$O(\log n)$-approx
Relative convex hull	$\Theta(n + k \log kn)$	k points in simple n-gon
Monotone path problem	$O(n^3 \log n)$	

Monotone path problem: Find a shortest monotone path (if any) from s to t in a polygonal domain P. A polygonal path is ***monotone*** if there exists a direction vector d such that every directed edge of the path has a nonnegative inner product with d.

MINIMUM SPANNING TREES

The (Euclidean) minimum spanning tree problem can be solved to optimality in the plane in time $O(n \log n)$ by appealing to the fact that the MST is a subgraph of the Delaunay triangulation; see Chapters 20 and 22.

The Steiner tree and k-MST problems, however, are NP-hard. Polynomial-time approximation schemes have recently been obtained, allowing one, for any fixed $\epsilon > 0$, to get within a factor $(1+\epsilon)$ of optimality in time $O(n^{O(1/\epsilon)})$ [Aro96, Mit96c].

TRAVELING SALESMAN PROBLEM

The traveling salesman problem is a classical problem in combinatorial optimization, and has been studied extensively in its geometric forms. The problem is NP-hard, but has a simple 2-approximation algorithm based on "doubling" the minimum spanning tree. The somewhat more involved Christofides heuristic yields a 1.5-approximation factor, which, until very recently, was the best factor known. There is now a polynomial-time approximation scheme for geometric versions of the planar TSP, allowing one, for any fixed $\epsilon > 0$, to get within a factor $(1+\epsilon)$ of optimal in time $O(n^{O(1/\epsilon)})$ [Aro96, Mit96c].

The **TSP-*with-neighborhoods*** problem arises when we require the tour/path to visit a set of regions, rather than a set of points. Constant-factor approximation algorithms are known for some special cases, and an $O(\log n)$-approximation algorithm is known for the general case in the plane.

A closely related problem is that of computing an optimal path for a lawnmower, modeled as, say, a circular cutter that must sweep out a region that covers a given domain of "grass." This problem is NP-hard in general, but constant-factor approximation algorithms are known.

WATCHMAN ROUTE PROBLEM

Another problem closely related to the TSP is the watchman route problem, which can be thought of as a shortest-path/tour problem in which we have the constraint that the path/tour must visit the visibility region associated with each point of the domain.

In the case of a simple polygonal domain, the watchman route problem has an $O(n^4)$ time algorithm to compute an exact solution [CJN93, N95]; $O(n^2)$ is possible if we are given a point through which the tour must pass. In the case of a polygonal domain with holes, the problem is easily seen to be NP-hard (from Euclidean TSP), and the best approximation algorithm is one with error factor $O(\log n)$.

OPEN PROBLEMS

1. Is the MAX TSP NP-hard?
2. Is the milling problem NP-hard for the case of a simple polygon?
3. Does the TSP-with-neighborhoods problem have an efficient approximation algorithm if the neighborhoods are not connected sets (e.g., if the neighborhoods are pairs of points)?
4. Does the watchman route problem in a polygonal domain have an $O(1)$-approximation algorithm?

24.5 HIGHER DIMENSIONS

GLOSSARY

Polyhedral domain: A set $P \subset \mathbb{R}^3$ whose interior is connected and whose boundary consists of a union of a finite number of triangles. (The definition is readily extended to d dimensions, where the boundary must consist of a union of $(d-1)$-simplices.) The complement of P consists of connected (polyhedral) components, which are the ***obstacles.***

Orthohedral domain: A polyhedral domain having each boundary facet orthogonal to one of the coordinate axes.

Polyhedral surface: A connected union of triangles, with any two triangles intersecting in a common edge, a common vertex, or not at all, and such that every point in the relative interior of the surface has a neighborhood homeomorphic to a disk.

Edge sequence: The ordered list of obstacle edges that are intersected by a path.

COMPLEXITY

In three or more dimensions, most shortest-path problems become very difficult. In particular, there are two sources of complexity, even in the most basic Euclidean shortest-path problem in a polyhedral domain P.

One difficulty arises from algebraic considerations. In general, the structure of a shortest path in a polyhedral domain need not lie on any kind of discrete graph. Shortest paths in a polyhedral domain will be polygonal, with bend points that generally lie *interior* to obstacle edges, obeying a simple "unfolding" property: The path must enter and leave at the same angle to the edge. It follows that any locally optimal subpath joining two consecutive obstacle vertices can be "unfolded" at each edge along its edge sequence, thereby obtaining a straight segment. Given an edge sequence, this local optimality property uniquely identifies a shortest path through that edge sequence. However, to compare the lengths of two paths, each one shortest with respect to two (different) edge sequences, requires exponentially many bits, since the algebraic numbers that describe the optimal path lengths may have exponential degree.

A second difficulty arises from combinatorial considerations. The number of combinatorially distinct (i.e., having distinct edge sequences) shortest paths between two points may be exponential. This fact leads to a proof of the NP-hardness of the shortest-path problem [CR87], even if the obstacles are simply a set of parallel triangles.

Thus, it is natural to consider approximation algorithms for the general case, or to consider special cases for which polynomial bounds are achievable.

SPECIAL CASES

If the polyhedral domain P has only a small number k of convex obstacles, a shortest path can be found in $n^{O(k)}$ time. If the obstacles are known to be vertical

"buildings" (prisms) having only k different heights, then shortest paths can be found in time $O(n^{6k-1})$, but it is not known if this version of the problem is NP-hard if k is allowed to be large.

If we require paths to stay on a polyhedral surface (i.e., the domain P is essentially 2-dimensional), then the unfolding property of optimal paths can be exploited to yield polynomial-time algorithms. The current best methods are based on the continuous Dijkstra paradigm, yielding an $O(n^2)$ time (and $O(n)$ space) algorithm to construct a shortest path map (or a geodesic Voronoi diagram), where n is the number of vertices of the surface.

Several facts are known about the set of edge sequences corresponding to shortest paths on the surface of a *convex* polytope P in $\mathbb{R}^3$. In particular, the worst-case number of distinct edge sequences that correspond to a shortest path between some pair of points is $\Theta(n^4)$, and the exact set of such sequences can be computed in time $O(n^6\beta(n)\log n)$, where $\beta(n) = o(\log^* n)$ [AAOS97]. (A simpler $O(n^6)$ algorithm can compute a small superset of the sequences.) The number of ***maximal*** edge sequences for shortest paths is $\Theta(n^3)$. Some of these results depend on a careful study of the ***star unfolding*** with respect to a point p on the boundary, ∂P, of P. The star unfolding is the (nonoverlapping) cell complex obtained by subtracting from ∂P the shortest paths from p to the vertices of P, and then flattening the resulting boundary.

Results on exact algorithms for special cases are summarized in Table 24.5.1.

APPROXIMATION ALGORITHMS

Papadimitriou [Pap85] was the first to study the general problem from the point of view of approximations. He gave a fully polynomial approximation scheme that produces a path guaranteed to be no longer than $(1+\epsilon)$ times the length of a shortest path. His algorithm requires time $O(n^3(L+\log(n/\epsilon))^2/\epsilon)$, where L is the number of bits necessary to represent the value of an integer coordinate of a vertex of P. Clarkson also gives a fully polynomial approximation scheme, which improves upon that of Papadimitriou in the case that $n\epsilon^3$ is large.

Recently, the analysis of Papadimitriou has been re-examined closely [CSY95]. Using the bit complexity model of computation and a notion of "precision-sensitivity" to write the complexity in terms of a parameter δ measuring the implicit precision of the input instance, an approximation algorithm can be achieved that is polynomial in $1/\delta$ and the other parameters of the input, with only linear dependence on $1/\epsilon$.

Results on approximation algorithms, both in the general case and in special cases, are summarized in Table 24.5.2.

OTHER METRICS

Link distance in a polyhedral domain in $\mathbb{R}^d$ can be approximated (within factor 2) in polynomial time by searching a weak visibility graph whose nodes correspond to simplices in a simplicial decomposition of the domain. The complexity of computing the exact link distance is open.

For the case of orthohedral domains and rectilinear (L_1) shortest paths, the shortest-path problem in $\mathbb{R}^d$ becomes relatively easy to solve in polynomial time,

TABLE 24.5.1 Shortest paths in 3-space: exact algorithms.

OBSTACLES/DOMAIN	COMPLEXITY	NOTES
Polyhedral domain	NP-hard	also for convex obstacles
k convex polytopes	$n^{O(k)}$	fixed k
Vertical buildings	$O(n^{6k-1})$	k different heights
Axis-parallel boxes	$O(n^2 \log^3 n)$	L_1 metric
Axis-parallel disjoint boxes	$O(n^2 \log n)$	L_1 metric
		monotonicity of paths in $\mathbb{R}^d$
Axis-parallel boxes in $\mathbb{R}^d$	$O(n^d \log n)$ preproc	combined L_1, link metric
	$O(\log^{d-1} n)$ query	single-source queries
	$O((n \log n)^{d-1})$ space	
Polyhedral surface	$O(n^2)$ time, $O(n)$ space	builds SPM(s), geodesic Voronoi
Two-point query	$O((\sqrt{n}/m^{1/4}) \log n)$ query	convex polytope
	$O(n^6 m^{1+\delta})$ space, preproc	$1 \leq m \leq n^2$, $\delta > 0$
Geodesic diameter	$O(n^8 \log n)$	convex polytope

TABLE 24.5.2 Shortest paths in 3-space: approximation algorithms.

OBSTACLES/DOMAIN	COMPLEXITY	APPROX FACTOR
Polyhedral domain	$O(n^4(L + \log(\frac{n}{\epsilon}))^2/\epsilon^2)$	$(1+\epsilon)$
	$O(n^2 \text{polylog } n/\epsilon^4)$	$(1+\epsilon)$
k convex polytopes	$O(n)$	$2k$
Convex polyhedral surface	$O(n)$	2
Convex polyhedral surface	$O(n \log \frac{1}{\epsilon} + \frac{1}{\epsilon^3})$	$(1+\epsilon)$
Vertical buildings	$O(n^2)$	1.1
Min link, polyhedral domain	poly(n)	2

since the grid graph induced by the facets of the domain serves as a path-preserving graph that we can search for an optimal path. In $\mathbb{R}^3$, we can do better than to use the $O(n^3)$ grid graph induced by $O(n)$ facets; an $O(n^2 \log^2 n)$ size subgraph suffices, which allows a shortest path to be found using Dijkstra's algorithm in time $O(n^2 \log^3 n)$. More generally, in $\mathbb{R}^d$ one can compute a data structure of size $O((n \log n)^{d-1})$, in $O(n^d \log n)$ preprocessing time, that supports fixed-source link distance queries in $O(\log^{d-1} n)$ time. In fact, this last result can be extended, within the same complexities, to the case of a combined metric, in which path cost is measured as a linear combination of L_1 length and rectilinear link distance.

For the special case of disjoint rectilinear box obstacles and rectilinear (L_1) shortest paths, a recent structural result may help in devising very efficient algorithms: There always exists a coordinate direction such that *every* shortest path from s to t is monotone in this direction [CY96]. In fact, this result has led to an $O(n^2 \log n)$ algorithm for the case $d = 3$.

OPEN PROBLEMS

1. Can one compute shortest paths on a convex polytope in $\mathbb{R}^3$ in subquadratic time?

2. Can one compute a shortest path map for a polyhedral domain in output-sensitive time?
3. What is the complexity of the minimum-link path problem in 3-space?
4. Can one give very simple and efficient approximation algorithms for shortest paths in polyhedral domains or on nonconvex surfaces? (See [AHSV96].)
5. What is the complexity of the shortest-path problem in 3-space for special cases of obstacles—e.g., disjoint aligned boxes, unit spheres, etc.?

24.6 SOURCES AND RELATED MATERIAL

SURVEYS

Explicit bibliographic citations for all results mentioned here are found in the extended version of this survey, [Mit96b]. Several other surveys offer a wealth of additional and related material:

[AW88]: A survey of shortest paths and visibility graphs.

[BE96]: A survey of approximation algorithms for geometric optimization problems.

[HSS87]: A book on motion planning algorithms.

[LYW96]: A recent survey article on rectilinear path problems.

[MS95a]: A survey of computational geometry, with a large section on shortest paths.

[SW92]: A survey of topological network design problems.

RELATED CHAPTERS

Chapter 22: Triangulations
Chapter 23: Polygons
Chapter 25: Visibility
Chapter 40: Algorithmic motion planning

REFERENCES

[AAOS97] P.K. Agarwal, B. Aronov, J. O'Rourke, and C. Schevon. Star unfolding of a polytope with applications. *SIAM J. Comput.*, 1997, to appear.

[AHSV96] P.K. Agarwal, S. Har-Peled, M. Sharir, and K.R. Varadarajan. Approximate shortest paths on a convex polytope in three dimensions. Technical Report CS-1996-12, Duke Univ., Durham, 1996.

[AMP91] E.M. Arkin, J.S.B. Mitchell, and C.D. Piatko. Bicriteria shortest path problems in the plane. In *Proc. 3rd Canad. Conf. Comput. Geom.*, pages 153–156, 1991.

[Aro96] S. Arora. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. In *Proc. 37th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 2–11, 1996.

[AW88] H. Alt and E. Welzl. Visibility graphs and obstacle-avoiding shortest paths. *Zeitschrift für Operations Research*, 32:145–164, 1988.

[BE96] M. Bern and D. Eppstein. Approximation algorithms for NP-hard problems. In D. Hochbaum, editor, *Approximation Problems for NP-Complete Problems*, pages 296–345, PWS Publications, Boston, 1996.

[BL96] J.-D. Boissonnat and S. Lazard. A polynomial-time algorithm for computing a shortest path of bounded curvature amidst moderate obstacles. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 242–251, 1996.

[CKT95] D.Z. Chen, K.S. Klenk, and H-Y.T. Tu. Shortest path queries among weighted obstacles in the rectilinear plane. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 370–379, 1995.

[CJN93] S. Carlsson, H. Jonsson, and B.J. Nilsson. Finding the shortest watchman route in a simple polygon. In *Proc. 4th Annu. Internat. Sympos. Algorithms Comput.*, volume 762 of *Lecture Notes in Comput. Sci.*, pages 58–67. Springer-Verlag, Berlin, 1993.

[CN91] W.-P. Chin and S. Ntafos. Watchman routes in simple polygons. *Discrete Comput. Geom.*, 6:9–31, 1991.

[CR87] J. Canny and J.H. Reif. New lower bound techniques for robot motion planning problems. In *Proc. 28th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 49–60, 1987.

[CSY94] J. Choi, J. Sellen, and C.K. Yap. Approximate Euclidean shortest path in 3-space. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 41–48, 1994.

[CSY95] J. Choi, J. Sellen, and C.K. Yap. Precision-sensitive Euclidean shortest path in 3-space. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 350–359, 1995.

[CY96] J. Choi and C.K. Yap. Monotonicity of rectilinear geodesics in d-space. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 339–348, 1996.

[GT93] M.T. Goodrich and R. Tamassia. Dynamic ray shooting and shortest paths via balanced geodesic triangulations. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 318–327, 1993.

[HS94] J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. *Comput. Geom. Theory Appl.*, 4:63–98, 1994.

[HS95] J. Hershberger and S. Suri. Efficient computation of Euclidean shortest paths in the plane. In *Proc. 34th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 508–517, 1993.

[HSS87] J.E. Hopcroft, J.T. Schwartz, and M. Sharir. *Planning, Geometry, and Complexity of Robot Motion.* Ablex Publishing, Norwood, 1987.

[IRWY92] C. Icking, G. Rote, E. Welzl, and C.-K. Yap. Shortest paths for line segments. *Algorithmica*, 10:182–200, 1993.

[LYW96] D.T. Lee, C.D. Yang, and C.K. Wong. Rectilinear paths among rectilinear obstacles. *Discrete Appl. Math.*, 70:186–215, 1996.

[Mit92] J.S.B. Mitchell. L_1 shortest paths among polygonal obstacles in the plane. *Algorithmica*, 8:55–88, 1992.

[Mit96a] J.S.B. Mitchell. Shortest paths among obstacles in the plane. *Internat. J. Comput. Geom. Appl.*, 6:309–332, 1996.

[Mit96b] J.S.B. Mitchell. Shortest paths and networks. Technical Report, Univ. at Stony Brook, 1996. [Available at `http://ams.sunysb.edu/~jsbm/jsbm.html`.]

[Mit96c] J.S.B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: Part II —A simple polynomial-time approximation scheme for geometric k-MST, TSP, and related problems. Technical Report, Univ. at Stony Brook, 1996. Part I appears in *SODA'96*, pages 402–408. [Both are available at `http://ams.sunysb.edu/~jsbm/jsbm.html`.]

[MS95a] J.S.B. Mitchell and S. Suri. Geometric algorithms. In M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors, *Network Models*, Handbook of Operations Research/Management Science, pages 425–479. Elsevier, Amsterdam, 1995.

[MS95b] J.S.B. Mitchell and S. Suri. Separation and approximation of polyhedral objects. *Comput. Geom. Theory Appl.* 5:95–114, 1995.

[N95] B.J. Nilsson. *Guarding Art Galleries—Methods for Mobile Guards.* PhD thesis, Lund Univ., 1995.

[Pap85] C.H. Papadimitriou. An algorithm for shortest-path motion in three dimensions. *Inform. Process. Lett.*, 20:259–263, 1985.

[SW92] J.M. Smith and P. Winter. Computational geometry and topological network design. In D.-Z. Du and F.K. Hwang, editors, *Computing in Euclidean Geometry*, volume 1 of *Lecture Notes Series on Computing*, pages 287–385. World Scientific, Singapore, 1992.

[WA96] H. Wang and P.K. Agarwal. Approximation algorithms for curvature constrained shortest paths. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms (SODA '96)*, pages 409–418, 1996.

25 VISIBILITY

Joseph O'Rourke

INTRODUCTION

In a geometric context, two objects are "visible" to each other if there is a line segment connecting them that does not cross any obstacles. Over 300 papers have been published on aspects of visibility in computational geometry in the last two decades. The research can be broadly classified as primarily focused on combinatorial issues, or primarily focused on algorithms. We partition the combinatorial work into "art gallery theorems" (Section 25.1) and research on visibility graphs (Section 25.2), and the algorithmic work into that concerned with polygons (Section 25.3), more general planar environments (Section 25.4), and paths (Section 25.5). All of this work concerns visibility in two dimensions. Investigations in three dimensions, both combinatorial and algorithmic, are discussed in the final section (25.6).

25.1 ART GALLERY THEOREMS

A typical "art gallery theorem" provides combinatorial bounds on the number of guards needed to visually cover a polygonal region P (the art gallery) defined by n vertices. Equivalently, one can imagine light bulbs instead of guards and require full direct-light illumination.

GLOSSARY

Guard: A point, a source of visibility or illumination.

Vertex guard: A guard at a polygon vertex.

Point guard: A guard at an arbitrary point.

Interior visibility: A guard $x \in P$ can see a point $y \in P$ if the segment xy is nowhere exterior to P: $xy \subset P$.

Exterior visibility: A guard x can see a point y outside of P if the segment xy is nowhere interior to P.

Star polygon: A polygon visible from a single interior point.

MAIN RESULTS

The most general results obtained to date are summarized in Table 25.1.1. In all cases, the number of guards listed is the number that is necessary for some polygons, and sufficient for all polygons. Thus all bounds listed are tight.

TABLE 25.1.1 Number of guards needed.

PROBLEM NAME	POLYGONS	INT/EXT	GUARD	NUMBER
Art gallery theorem	simple	interior	vertex	$\lfloor n/3 \rfloor$
Fortress problem	simple	exterior	point	$\lceil n/3 \rceil$
Prison yard problem	simple	int & ext	vertex	$\lceil n/2 \rceil$
Orthogonal polygons	simple orthogonal	interior	vertex	$\lfloor n/4 \rfloor$
Orthogonal with holes	orthogonal with h holes	interior	vertex	$\lfloor n/4 \rfloor$
Polygons with holes	polygons with h holes	interior	point	$\lfloor (n+h)/3 \rfloor$

COVERS AND PARTITIONS

Each art gallery theorem above implies a cover result, a cover by star polygons. Many of the theorem proofs rely on certain partitions. For example, the orthogonal polygon result depends on the theorem that every orthogonal polygon may be partitioned via diagonals into convex quadrilaterals. See Section 23.2 for more on covers and partitions.

RELATED PROBLEMS

The above problems revolve around polygons. Another cluster of problems requiring guarding (or illumination) of the boundaries of collections of objects in the plane: convex sets, circles, triangles, isothetic rectangles, congruent squares, and line segments have all been investigated. Still other variations include illumination by line segment light sources (fluorescent light bulbs), equivalent to permitting guards to patrol segments, and, most recently, restricting the angular range of visibility, resulting in "floodlight" problems [BGL+93].

OPEN PROBLEMS

Three notable open questions remain, as well as a fourth area that has yet to be explored fully:

1. *Orthogonal prison yard problem:* How many vertex guards are needed to cover both the interior and the exterior of a simple orthogonal polygon? The best

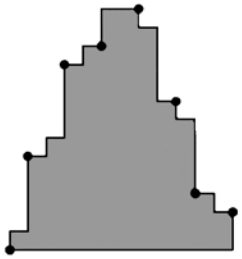


FIGURE 25.1.1
A pyramid polygon with $n = 24$ vertices whose interior and exterior are covered by 8 guards. Repeating the pattern establishes a lower bound of $5n/16 + c$ on the orthogonal prison yard problem [HK93].

lower and upper bounds are $\lceil (5n-10)/16 \rceil$ and $\lfloor 5n/12 \rfloor + 2$ [HK93]. See Figure 25.1.1.

2. *Edge guard problem:* How many edge guards are needed to cover the interior of a simple polygon? The best lower and upper bounds (for $n > 11$) are $\lfloor n/4 \rfloor$ and $\lfloor 3n/10 \rfloor$ [She94].

3. *π-floodlights:* How many floodlights, each with aperture π, and with their apexes at distinct nonexterior points, are sufficient to cover any polygon of n vertices? It is known that $\lceil \frac{2}{5}(n-3) \rceil$ suffice [CT97], but it has been conjectured that $\lfloor n/3 \rfloor$ suffice [Urr].

4. *Approximation algorithms:* The problem of finding the minimum number of guards needed to cover a given simple polygon is NP-complete [O'R87, Chapter 9]. Is there a polynomial approximation algorithm with some guarantee of performance? Since set cover can be approximated within an $O(\log n)$ factor, improving upon $O(\log n)\cdot$opt could be a goal.

25.2 VISIBILITY GRAPHS

Whereas art gallery theorems seek to encapsulate an environment's visibility into one function of n, the study of visibility graphs endeavors to uncover the more fine-grained structure of visibility. The original impetus for their investigation came from pattern recognition, and its connection to shape continues to be one of its primary sources of motivation; see Chapter 43. Another application is graphics (Chapter 42): illumination and radiosity depend on 3D visibility relations (Section 25.6.)

GLOSSARY

Visibility graph: A graph with nodes for each object, and arcs between objects that can see one another.

Vertex visibility graph: The objects are the vertices of a simple polygon.

Endpoint visibility graph: The objects are the endpoints of line segments in the plane. See Figure 25.2.1b.

Segment visibility graph: The objects are whole line segments in the plane, either open or closed.

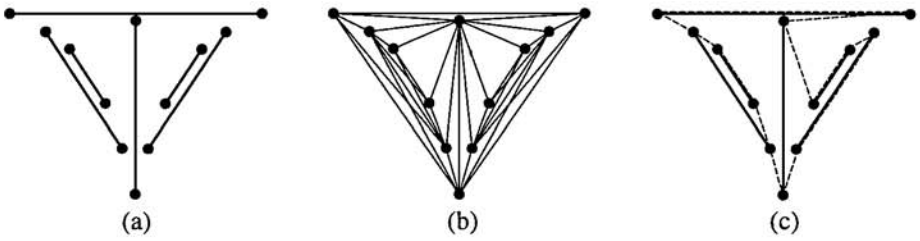
Object visibility: Two objects A and B are visible to one another if there are points $x \in A$ and $y \in B$ such that x sees y.

Point visibility: Two points x and y can see one another if the segment xy is not "obstructed," where the meaning of "obstruction" depends on the problem.

ϵ-visibility: Lines of sight are finite-width beams of visibility.

Hamiltonian: A graph is Hamiltonian if there is a simple cycle that includes every node.

FIGURE 25.2.1
(a) *A set of segments.* (b) *Their endpoint visibility graph G.* (c) *A Hamiltonian cycle in G.*



OBSTRUCTIONS TO VISIBILITY

For polygon vertices, x sees y if xy is nowhere exterior to the polygon, just as in art gallery visibility; this implies that polygon edges are part of the visibility graph. For segment endpoints, x sees y if the closed segment xy intersects the union of all the segments either in just the two endpoints, or in the entire closed segment. This disallows grazing contact with a segment, but includes the segments themselves in the graph.

GOALS

Four goals can be discerned in research on visibility graphs:

1. Characterization: asks for a precise delimiting of the class of graphs realizable by a certain class of geometric objects.
2. Recognition: asks for an algorithm to recognize when a graph is a visibility graph.
3. Reconstruction: asks for an algorithm that will take a visibility graph as input, and output a geometric realization.
4. Counting: concerned with the number of visibility graphs under various restrictions.

VERTEX VISIBILITY GRAPHS

A complete characterization of polygon vertex visibility graphs has remained elusive, but progress has been made by:

1. Restricting the class of polygons: polynomial-time recognition and reconstruction algorithms for orthogonal staircase polygons have been obtained. See Figure 25.2.2.
2. Restricting the class of graphs: every 3-connected vertex visibility graph has a 3-clique ordering, i.e., an ordering of the vertices so that each vertex is part of a triangle composed of preceding vertices.

3. Adding information: assuming knowledge of the boundary Hamiltonian circuit, necessary conditions have been conjectured by Ghosh and others [SM94].

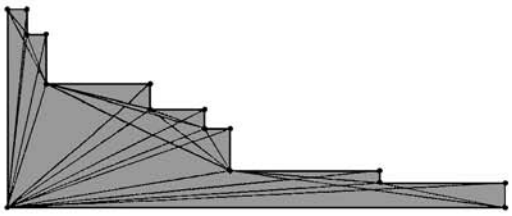


FIGURE 25.2.2
A staircase polygon and its vertex visibility graph.

ENDPOINT VISIBILITY GRAPHS

For segment endpoint visibility graphs, there have been two foci:

1. Are the graphs Hamiltonian? See Figure 25.2.1c. Restricted classes of segments have been settled as Hamiltonian: when all segments touch the hull, and unit segments with lattice endpoints [OR94].
2. Size questions: there must be at least $5n - 4$ edges; the smallest clique cover has size $\Omega(n^2/\log^2 n)$ [AAAS93].

SEGMENT VISIBILITY GRAPHS

Whole segment visibility graphs have been investigated most thoroughly under the restriction that the segments are all (say) vertical and visibility is horizontal. Such segments are often called ***bars***. The visibility is usually required to be ϵ-visibility. Endpoints on the same horizontal often play an important role here, as does the distinction between closed segments and intervals (which may or may not include their endpoints). There are several characterizations:

1. G is representable by segments, with no two endpoints on the same horizontal, iff there is a planar embedding of G such that, for every interior k-face F, the induced subgraph of F has exactly $2k - 3$ edges.
2. G is representable by segments, with endpoints on the same horizontal permitted, iff there is a planar embedding of G with all cutpoints on the exterior face.
3. Every 3-connected planar graph is representable by intervals.

OPEN PROBLEMS

1. Given a visibility graph G and a Hamiltonian circuit C, construct in polynomial time a simple polygon such that its vertex visibility graph is G, with C corresponding to the polygon's boundary.
2. Develop an algorithm to recognize whether a polygon vertex visibility graph is planar. Necessary and sufficient conditions are known [LC94].

25.3 ALGORITHMS FOR VISIBILITY IN A POLYGON

Designing algorithms to compute aspects of visibility in a polygon P was a major focus of the computational geometry community in the 1980's. For most of the basic problems, optimal algorithms were found, several depending on Chazelle's linear-time triangulation algorithm [Cha91].

GLOSSARY

Throughout, P is a polygon.

Kernel: The set of points in P that can see all of P. See Figure 33.4.4.

Point visibility polygon: The region visible from a point in P.

Segment visibility polygon: The region visible from a segment in P.

MAIN RESULTS

The main algorithms are listed in Table 25.3.1. We discuss two of these algorithms below to illustrate their flavor.

TABLE 25.3.1 Polygon visibility algorithms.

ALGORITHM TO COMPUTE	TIME COMPLEXITY	SOURCE
Kernel	$O(n)$	[LP79]
Point visibility polygon	$O(n)$	[JS87]
Segment visibility polygon	$O(n)$	[GHL+87]
Shortest illuminating segment	$O(n)$	[DN94]
Vertex visibility graph	$O(E)$	[Her89]

VISIBILITY POLYGON ALGORITHM

Let $x \in P$ be the visibility source. Lee's linear-time algorithm [JS87] processes the vertices of P in a single counterclockwise boundary traversal. At each step, a vertex is either pushed on or popped off a stack, or a *wait* event is processed. The latter occurs when the boundary at that point is invisible from x. At any stage, the stack represents the visible portion of the boundary processed so far.

Although this algorithm is elementary in its tools, it has proved delicate to implement correctly.

VISIBILITY GRAPH ALGORITHM

In contrast, Hershberger's vertex visibility algorithm [Her89] uses sophisticated tools to achieve output-size sensitive time complexity $O(E)$, where E is the num-

ber of edges of the graph. His algorithm exploits the intimate connection between shortest paths and visibility in polygons. It first computes the *shortest path map* (Chapter 24) in $O(n)$ time for a vertex, and then systematically transforms this into the map of an adjacent vertex in time proportional to the number of changes. Repeating this achieves $O(E)$ time overall.

Most of the above algorithms have been parallelized; see, for example, [GSG92].

25.4 ALGORITHMS FOR VISIBILITY AMONG OBSTACLES

The shortest path between two points in an environment of polygonal obstacles follows lines of sight between obstacle vertices. This has provided an impetus for developing efficient algorithms for constructing visibility regions and graphs in such settings. The obstacles most studied are noncrossing line segments, which can be joined end-to-end to form polygonal obstacles. Many of the questions mentioned in the previous section can be revisited for this environment.

The major results are shown in Table 25.4.1, the last of which will be discussed further below.

TABLE 25.4.1 Algorithms for visibility among obstacles.

ALGORITHM TO COMPUTE	TIME COMPLEXITY
Point visibility region	$O(n \log n)$
Segment visibility region	$\Theta(n^4)$
Endpoint visibility graph	$O(n^2)$
Endpoint visibility graph	$O(n \log n + E)$

ENDPOINT VISIBILITY GRAPH

The largest effort has concentrated on constructing the endpoint visibility graph. Worst-case optimal algorithms were first discovered by constructing the line arrangement dual to the endpoints in $O(n^2)$ time. Since many visibility graphs have less than a quadratic number of edges, an output-size sensitive algorithm was a significant improvement: $O(n \log n + E)$ where E is the number of edges of the graph.

25.5 VISIBILITY PATHS

A fruitful idea was introduced to visibility research in the mid-1980's: the notion of "link distance" between two points, which represents the smallest number of mutually visible relay stations needed to communicate from one point to another (Sections 23.4 and 24.3). A related notion called "watchman tours" was introduced a bit later, mixing shortest paths and visibility problems, and employing many of

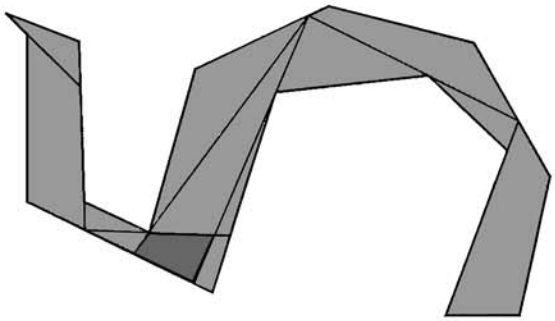


FIGURE 25.5.1
The link center is shown darkly shaded: every point in the polygon can be reached with no more than three links from a point in the center. Several key visibility chords are drawn.

the concepts developed for link-path problems (Section 24.4).

GLOSSARY

Link: A segment.

Link distance: The smallest number of links in a polygonal path connecting the points.

Link diameter of P: The largest link distance between any two points in P.

Link center of P: The collection of points whose maximal link distance to any point of P is as small as possible. See Figure 25.5.1.

Shortest watchman tour in P: A shortest closed path π in a polygon P such that every point of P is visible from some point of π.

MAIN RESULTS

The main results for link centers are shown in Table 25.5.1. See Tables 23.4.2 and 24.3.1 and the related sections for further results.

TABLE 25.5.1 Algorithms for link centers.

LINK CENTER WITHIN	TIME COMPLEXITY
Polygon	$O(n \log n)$
Orthogonal polygon	$O(n)$
Polygon with holes	NP-hard

25.6 VISIBILITY IN THREE DIMENSIONS

Research on visibility in three dimensions (3D) has concentrated on three topics: hidden surface removal, polyhedral terrains, and various 3D visibility graphs.

25.6.1 HIDDEN SURFACE REMOVAL

"Hidden surface removal" is one of the key problems in computer graphics (Chapter 42), and has been the focus of intense research for two decades. The typical problem instance is a collection of (planar) polygons in space, from which the view from $z = \infty$ must be constructed.

The complexity of the output scene can be quadratic in the number of input vertices n. A worst-case optimal $\Theta(n^2)$ algorithm can be achieved by projecting the lines containing each polygon edge to a plane and constructing the resulting arrangement of lines. Most recent work has focused on obtaining output-size sensitive algorithms, whose time complexity depends on the number of vertices k in the output scene (the complexity of the ***visibility map***), which is often less than quadratic in n. See Table 25.6.1 for selected results.

TABLE 25.6.1 Hidden surface algorithm complexities.

ENVIRONMENT	COMPLEXITY
Isothetic rectangles	$O((n+k)\log^2 n)$
Polyhedral terrain	$O((n+k)\log n \log\log n)$
Nonintersecting polyhedra	$O(n^{1+\epsilon}\sqrt{k})$
	$O(n\sqrt{k}\log n)$
Arbitrary intersecting spheres	$O(n^{2+\epsilon})$
Nonintersecting spheres	$O((n+k)\log^2 n)$
	$O(k+n^{3/2}\log n)$
Restricted-intersecting spheres	$O((n+k)\log^2 n)$

25.6.2 POLYHEDRAL TERRAINS

Polyhedral terrains are an important special class of 3D surfaces, arising in a variety of applications, most notably geographic information systems.

GLOSSARY

Polyhedral terrain: A polyhedral surface that intersects every vertical line in at most a single point.

Perspective view: A view from a point.

Orthographic view: A view from infinity (parallel lines of sight).

Ray-shooting query: A query asking which terrain face is first hit by a ray shooting in a given direction from a given point. (See Chapter 32.)

$\boldsymbol{\alpha(n)}$***:*** The inverse Ackermann function (nearly a constant). See Section 40.4.

COMBINATORIAL BOUNDS

Several almost-tight bounds on the maximum number of combinatorially different

views of a terrain have been obtained, as listed in Table 25.6.2. Here we use "$O^*(\ )$" to mean "nearly." For example, the quadratic upper bound in the first line of the table is $O(n^2 2^{\alpha(n)})$. See [AS93].

TABLE 25.6.2 Bounds for polyhedral terrains.

VIEW TYPE	APPROX BOUND
Along vertical	$O^*(n^2)$
Orthographic	$O^*(n^5)$
Perspective	$O^*(n^8)$

ALGORITHMS

Algorithms seek to exploit the terrain constraints to improve on the same computations for general polyhedra:

1. To compute the orthographic view from above the terrain: time $O((k+n)\log n \log\log n)$, where k is the output size [RS88].
2. To preprocess for $O(\log n)$ ray-shooting queries for rays with origin on a vertical line [BDEG94].

25.6.3 3D VISIBILITY GRAPHS

GLOSSARY

Aspect graph: A graph with a node for each combinatorially distinct view of a collection of polyhedra, with two nodes connected by an arc if the views can be reached directly from one another by a continuous movement of the viewpoint.

Isothetic: Edges parallel to Cartesian coordinate axes.

Box visibility graph: A graph realizable by disjoint isothetic boxes in 3D with orthogonal visibility.

K_n: The complete graph on n nodes.

There have been three primary motivations for studying visibility graphs of objects in three dimensions.

1. Computer graphics: Useful for accelerating interactive "walkthroughs" of complex polyhedral scenes [TS91], and for radiosity computations [TH93]. See Chapter 42.
2. Computer vision: "Aspect graphs" are used to aid image recognition. The maximum number of nodes in an aspect graph for a polyhedron of n vertices depends on both convexity and the type of view. See Table 25.6.3. Note that the nonconvex bounds are significantly larger than those for terrains.

TABLE 25.6.3 Combinatorial complexity of visibility graphs.

CONVEXITY	ORTHOGRAPHIC	PERSPECTIVE	SOURCE
Convex polyhedron	$\Theta(n^2)$	$\Theta(n^3)$	[PD90]
Nonconvex polyhedron	$\Theta(n^6)$	$\Theta(n^9)$	[GCS91]

3. Combinatorics: It has been shown that K_{22} is realizable by disjoint isothetic rectangles in "$2\frac{1}{2}$D" with vertical visibility (all rectangles are parallel to the xy-plane) [BEF+94], and that K_{42} is a box visibility graph [BJMO94]. Neither of these results is known to be tight.

OPEN PROBLEM

Finally, we mention the (three-dimensional case of the) difficult ***Hadwiger-Levi covering problem***: What is the smallest number of exterior points sufficient to illuminate any closed convex body in three dimensions? The conjecture is that 8 (in general 2^d) points suffice, but this is proven only for 3-polytopes with some affine symmetry [Bez93]. Illumination here requires the lines of sight to penetrate to the interior.

25.7 SOURCES AND RELATED MATERIAL

SURVEYS

All results not given an explicit reference above may be traced in these surveys.

[O'R87]: A monograph devoted to art gallery theorems and visibility algorithms.

[She92]: A survey of art gallery theorems and visibility graphs, updating [O'R87].

[O'R92]: A short update to [She92].

[Urr]: The latest art gallery results, updating [She92].

[O'R93]: Survey of visibility graph results.

[Dor94]: A survey of hidden surface removal algorithms, emphasizing recent theoretical developments.

RELATED CHAPTERS

Chapter 22: Triangulations
Chapter 23: Polygons
Chapter 24: Shortest paths and networks
Chapter 32: Ray shooting and lines in space
Chapter 33: Geometric intersection
Chapter 42: Computer graphics
Chapter 43: Pattern recognition

REFERENCES

[AAAS93] P.K. Agarwal, N. Alon, B. Aronov, and S. Suri. Can visibility graphs be represented compactly? In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 338–347, 1993.

[AS93] P. Agarwal and M. Sharir. On the number of views of polyhedral terrains. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 55–60, Waterloo, 1993.

[BDEG94] M. Bern, D. Dobkin, D. Eppstein, and R. Grossman. Visibility with a moving point of view. *Algorithmica*, 11:360–378, 1994.

[BEF+94] P. Bose, H. Everett, S. Fekete, A. Lubiw, H. Meijer, K. Romanik, T. Shermer, and S. Whitesides. On a visibility representation for graphs in three dimensions. In D. Avis and P. Bose, editors, *Snapshots of Computational and Discrete Geometry*, volume 3, pages 2–25, McGill University, Montréal, 1994.

[Bez93] K. Bezdek. Hadwiger-Levi's covering problem revisited. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms Combin.*, pages 199–233. Springer-Verlag, Berlin, 1993.

[BGL+93] P. Bose, L. Guibas, A. Lubiw, M. Overmars, D. Souvaine, and J. Urrutia. The floodlight problem. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 399–404, Waterloo, 1993.

[BJMO94] P. Bose, A. Josefczyk, J. Miller, and J. O'Rourke. K_{42} is a box visibility graph. In D. Avis and P. Bose, editors, *Snapshots of Computational and Discrete Geometry*, volume 3, pages 88–91, McGill University, Montréal, 1994.

[Cha91] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6:485–524, 1991.

[CT97] G. Csizmadia and G. Tóth. Note on an art gallery problem. *Comput. Geom. Theory Appl.*, 1997, to appear.

[DN94] G. Das and G. Narasimhan. Optimal linear-time algorithm for the shortest illuminating line segment. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 259–266, 1994.

[Dor94] S.E. Dorward. A survey of object-space hidden surface removal. *Internat. J. Comput. Geom. Appl.*, 4:325–362, 1994.

[GCS91] Z. Gigus, J. Canny, and R. Seidel. Efficiently computing and representing aspect graphs of polyhedral objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13:542–551, 1991.

[GHL+87] L.J. Guibas, J. Hershberger, D. Leven, M. Sharir, and R.E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2:209–233, 1987.

[GSG92] M.T. Goodrich, S. Shauck, and S. Guha. Parallel methods for visibility and shortest path problems in simple polygons. *Algorithmica*, 8:461–486, 1992.

[Her89] J. Hershberger. An optimal visibility graph algorithm for triangulated simple polygons. *Algorithmica*, 4:141–155, 1989.

[HK93] F. Hoffmann and K. Kriegel. A graph coloring result and its consequences for some guarding problems. In *Proc. 4th Annu. Internat. Sympos. Algorithms Comput.*, volume 762 of *Lecture Notes in Comput. Sci.*, pages 78–87. Springer-Verlag, New York, 1993.

[JS87] B. Joe and R.B. Simpson. Correction to Lee's visibility polygon algorithm. *BIT*, 27:458–473, 1987.

[LC94] S.-Y. Lin and C. Chen. Planar visibility graphs. In *Proc. 6th Canad. Conf. Comput. Geom.*, pages 30–35, Saskatoon, 1994.

[LP79] D.T. Lee and F.P. Preparata. An optimal algorithm for finding the kernel of a polygon. *J. Assoc. Comput. Mach.*, 26:415–421, 1979.

[O'R87] J. O'Rourke. *Art Gallery Theorems and Algorithms.* Oxford University Press, New York, 1987.

[O'R92] J. O'Rourke. Computational geometry column 15. *Internat. J. Comput. Geom. Appl.*, 2:215–217, 1992. Also in *SIGACT News*, 23:26–28, 1992.

[O'R93] J. O'Rourke. Computational geometry column 18. *Internat. J. Comput. Geom. Appl.*, 3:107–113, 1993. Also in *SIGACT News*, 24:20–25, 1993.

[OR94] J. O'Rourke and J. Rippel. Two segment classes with Hamiltonian visibility graphs. *Comput. Geom. Theory Appl.*, 4:209–218, 1994.

[PD90] H. Plantinga and C.R. Dyer. Visibility, occlusion, and the aspect graph. *Internat. J. Comput. Vision*, 5:137–160, 1990.

[RS88] J.H. Reif and S. Sen. An efficient output-sensitive hidden-surface removal algorithm and its parallelization. In *Proc. 4th Annu. ACM Sympos. Comput. Geom.*, pages 193–200, 1988.

[She92] T.C. Shermer. Recent results in art galleries. *Proc. IEEE*, 80:1384–1399, 1992.

[She94] T. Shermer. A tight bound on the combinatorial edge guarding problem. In D. Avis and P. Bose, editors, *Snapshots of Computational and Discrete Geometry*, volume 3, pages 191–223, McGill University, Montréal, 1994.

[SM94] G. Srinivasaraghavan and A. Mukhopadhyay. A new necessary condition for the vertex visibility graphs of simple polygons. *Discrete Comput. Geom.*, 12:65–82, 1994.

[TH93] S. Teller and P. Hanrahan. Global visibility algorithms for illumination computations. *Comput. Graph.* (Proc. SIGGRAPH '93), 27:239–246, 1993.

[TS91] S.J. Teller and C.H. Séquin. Visibility preprocessing for interactive walkthroughs. *Comput. Graph.* (Proc. SIGGRAPH '91), 25:61–69, 1991.

[Urr] J. Urrutia. Art gallery and illumination problems. In J.-R. Sack and J. Urrutia, editors, *Handbook on Computational Geometry.* North Holland, Amsterdam, to appear.

26 GEOMETRIC RECONSTRUCTION PROBLEMS

Steven S. Skiena

INTRODUCTION

Many problems from mathematics and engineering can be described in terms of reconstruction from geometric information. ***Reconstruction*** is the algorithmic problem of combining the results of one or more measurements of some aspect of a physical or mathematical object to obtain certain desired information about the object. Geometric reconstruction problems arise in a number of applications, such as robotics and computer-aided tomography, and also are of theoretical interest.

In this chapter, we consider three different classes of geometric reconstruction problems. In Section 26.1, we examine static reconstruction problems, where we are given a geometric structure G' derived from an original structure G, and seek to invert this transformation. In Section 26.2, we consider interactive reconstruction problems, where we are permitted to "probe" the object at arbitrary places and seek to reconstruct the desired structure using the fewest such probes. Finally, in Section 26.3, we provide pointers to the literature for work on (typically ill-defined) geometric reconstruction problems that often arise in practice.

26.1 STATIC RECONSTRUCTION PROBLEMS

Here we consider inverse problems of the following type. Let A be a geometric structure, and T a transformation such that $T(A) \rightarrow B$, where B is some different geometric structure. Now, given T and B, construct a structure A' such that $T(A') \rightarrow B$. If T is 1-1, then $A = A'$. If not, we may be interested finding or counting all solutions.

An example of an important class of reconstruction problems is recognizing visibility graphs, i.e., given a graph G, construct a polygon P whose visibility graph is G. See Section 25.2.

Results on static reconstruction problems are summarized in Table 26.1.1. We characterize each problem by its input instance and desired inverted structure. Static reconstruction problems include reconstructing sets of points from interpoint distances, extended Gaussian images [GH95], points from Voronoi diagrams [AB85], and orthogonal polygons from points [O'R88].

A special class of problems concerns proximity drawability. Given a graph G, we seek a set of points corresponding to vertices of G such that two points are "sufficiently" close if there is an edge in G for the corresponding vertices. Examples of proximity drawability problems include finding points to realize graphs as minimum spanning trees (MST), Delaunay triangulations (Chapter 22), Gabriel graphs, and relative neighborhood graphs (RNGs) (Chapter 43). Although many of the results are quite technical, Di Battista et al. [DLL95] provide an excellent survey

of results on these and other classes of proximity drawings; see also Chapter 44.

To provide some intuition about the minimum spanning tree results, observe that very low degree graphs are easily embedded as point sets. If the maximum degree is 2, i.e., the graph is a simple path, then any straight line embedding will work. To realize a vertex v of degree 6 as a minimum spanning tree, a geometric argument shows that all adjacent points must be spaced at equal angles of 60 degrees around v, a very restrictive condition leading to the hardness result. Such equal spacing is not possible for degree larger than six.

GLOSSARY

Extended Gaussian image: A transform that maps each face of a convex polyhedron to a vector normal to the face whose length is proportional to the area of the face. These vectors uniquely represent convex polyhedra and have been applied to problems in robot vision.

Hammer's X-ray problem: Given a fixed set of X-ray projections of a convex body, can you reconstruct the body?

Determination: A class of sets is determined by n directions if there are n fixed directions such that all sets can be reconstructed from projections along these directions.

Verification: A class of sets is verified by n directions if, for each particular set, there are n projections that distinguish this set from any other.

Gabriel graph: A graph whose vertices are points in $\mathbb{E}^2$, with edge (x, y) iff points x and y define the diameter of an empty circle.

Relative neighborhood graph: A graph whose vertices are points in $\mathbb{E}^2$, with an edge (x, y) iff there exists no point z such that z is closer to x than y is and z is closer to y than x is. See Section 43.2.

Interpoint distances: The complete set of $\binom{n}{2}$ distances defined between pairs of points in an n point set. The distance set is ***labeled*** if the identities of the two points defining the distance are associated with the distance, and ***unlabeled*** otherwise.

A final set of problems concern reconstructing objects from a fixed set of X-ray projections, conventionally called Hammer's X-ray problem. Different problems arise depending upon whether the X-rays originate from a point or line source, and whether we seek to verify or determine the object. A selection of results on parallel X-rays (line sources) are listed in Table 26.1.2. For example, parallel X-rays in certain sets of four directions suffice to determine any convex body; the directions must not be a subset of the edges of an affinely regular polygon. If the directions do form such a subset, then there exist noncongruent polygons that are not distinguished by any number n of parallel X-rays in these directions. Nevertheless, any pair of nonparallel directions suffice to determines "most" (in the sense of Baire category) convex sets.

There is also a collection of results on *point source X-rays*. For example, convex sets in $\mathbb{E}^2$ are determined by directed X-rays from three noncollinear point sources. The substantial literature on such X-ray problems is most ably covered by Gardner's monograph [Gar95], from which several of the open problems listed below are drawn.

TABLE 26.1.1 Static reconstruction problems.

INPUT	INVERTED STRUCTURE	RESULT
Tree with max degree ≤ 5	points embedding it as MST	every tree realizable
Tree with max degree 6	points embedding it as MST	NP-hard
Tree with max degree ≥ 7	points embedding it as MST	no tree realizable
Planar graph	points embedding it as a Gabriel graph	partial characterization
Planar graph	points embedding it as a RNG	partial characterization
Triangulated graph	points embedding it as a Delaunay tri	partial characterization
Points in $\mathbb{E}^2$	orthogonal polygon through them	algorithm: $O(n \log n)$
Planar graph	points embedding it as a Voronoi diag	partial characterization
Extended Gaussian image	convex polyhedra in $\mathbb{E}^3$	algorithm: $O(n \log n)$ per iter
Labeled interpoint dists	points realizing these in $\mathbb{E}^d$	algorithm: $O(2^d n^2)$
Unlabeled interpoint dists	points realizing these on $\mathbb{E}^1$	algorithm: $O(2^n n \log n)$
Unlabeled interpoint dists	points realizing these in $\mathbb{E}^d$	NP-hard

TABLE 26.1.2 Selected results on parallel X-rays (Hammer's problem).

DIM	PROBLEM	SETS	RESULT
2	verify	convex polygons	2 parallel X-rays do not suffice
	verify	convex set	3 parallel X-rays suffice
	determine	convex set	4 parallel X-rays suffice ($/\supseteq$ affinely reg polygon)
	determine	convex set	n parallel X-rays do not suffice ($\subseteq$ affinely reg polygon)
	determine	convex set	2 parallel X-rays "usually" suffice
	determine	star-shaped polygons	no finite set of directions suffice
3	determine	convex body	4 parallel X-rays suffice (coplanar directions)
	determine	convex body	4 parallel X-rays do not suffice (noncoplanar directions)
d	determine	convex body	2 parallel X-rays "usually" suffice
	determine	compact sets	no finite set of directions suffice

OPEN PROBLEMS

1. Give an algorithm (polynomial in n) to reconstruct a set of n points on a line from the set of $\binom{n}{2}$ unlabeled interpoint distances it defines. See [SSL90].
2. Do there exist two distinct n-point sets, $n \geq 7$, realizing identical unlabeled interpoint distance sets, where each distance is unique in the set? See [Blo77].
3. Characterize the convex sets in $\mathbb{E}^2$ that can be determined by two parallel X-rays [Gar95, Problem 1.1].

4. Are convex bodies in $\mathbb{E}^3$ determined by parallel X-rays in some set of five directions [Gar95, Problem 2.2]?

5. Find an algorithm to reconstruct a convex set from its directed X-rays from three noncollinear points [Gar95, Problem 5.5]. The uniqueness proof is nonconstructive.

26.2 INTERACTIVE RECONSTRUCTION PROBLEMS

Geometric probing considers problems of determining a geometric structure or some aspect of that structure from the results of a mathematical or physical measuring device, a ***probe***. A variety of problems from robotics, medical instrumentation, mathematical optimization, integral and computational geometry, graph theory, and other areas fit into this paradigm. The key issue is interaction, where the nth probe depends upon the outcome of the previous probes.

The problem of geometric probing was introduced by Cole and Yap [CY87] and inspired by work in robotics and tactile sensing (Section 41.7). A substantial body of work has followed it, which is extensively surveyed in [Ski92]. A comprehensive collection of open problems in probing appears in [Ski89].

GLOSSARY

Determination: The algorithmic problem of computing how many probes of a particular model are necessary to completely determine or reconstruct an object drawn from a particular class of objects.

Verification: The algorithmic problem, given a supposed description of an object, of computing how many probes of a particular model are necessary to test if the description is valid.

Model-based: A problem where any object is constrained to be one of a known, finite set of m possible objects.

Point probe: An oracle that tests whether a given point is within an object or not.

Finger probe: An oracle that returns the first point of intersection between a directed line and an object.

Hyperplane probe: An oracle that measures the first time at which a hyperplane moving parallel to itself intersects an object.

X-ray probe: An oracle that measures the length of the intersection between a line and an object.

Silhouette probe: An oracle that returns a $(d-1)$-dimensional projection (in a given direction) of a d-dimensional object.

Halfspace probe: An oracle that measures the area or volume of the intersection between a halfspace and an object.

Cut-set probe: An oracle that for a specified graph and partition of the vertices returns the size of the cut-set determined by the partition.

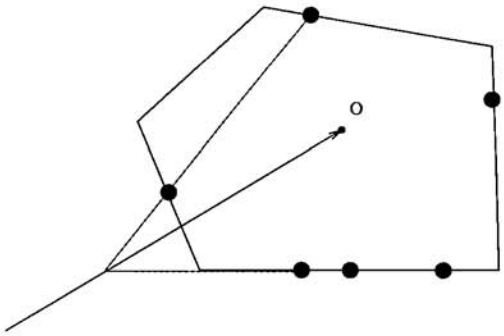


FIGURE 26.2.1
Determining the next edge of P using finger probes.

MAIN RESULTS

For a particular probing model, the determination problem asks how many probes are sufficient to completely reconstruct an object from a given class. For example, Cole and Yap's strategy for reconstructing a convex polygon P from finger probes is based on the observation that three collinear contact points must define an edge. The strategy, illustrated in Figure 26.2.1, aims probes at the intersection point between a confirmed edge (defined by three collinear points) and a conjectured edge (defined by two contact points). If this intersection point is indeed a contact point, another vertex is determined due to convexity; if not, the existence of another edge is known. Since we avoid probing the interior of any edge that has been determined, $\approx 3n$ probes suffice in total since no more than one edge can be hit four times.

Table 26.2.1 summarizes probing results for a wide variety of models. In the table, $f_i(P)$ denotes the number of i-dimensional faces of P.

Cole and Yap's finger probing model is not powerful enough to determine nonconvex objects. There are three major reasons for this. A tiny crack in an edge can go forever undetected, since no finite strategy can explore the entire surface of the polygon. Second, it is easy to construct nonconvex polygons whose features cannot be entirely contacted with straight-line probes originating from infinity. Finally, for nonconvex polygons there exists no constant k such that k collinear probes determine an edge. To generalize the class of objects, enhanced finger probes have been considered. One such probe [ABY90] returns surface normals as well as contact points, eliminating the second problem. When restricted to polygons with no two edges defined by the same supporting line, the first and third problems are eliminated.

In the verification problem, we are given a description of a putative object, and charged with using a small number of probes to prove that the description is correct. Verification is clearly no harder than determination, since we are free to ignore the description in planning the probes, and could simply compare the determined object to its description. Sometimes significantly fewer probes suffice for verification. For example, we can verify a putative convex polygon in $2n$ probes by sending one finger probe to contact each vertex and the interior of each edge. This gives three contact points on each edge, which by convexity suffices to verify the polygon. Table 26.2.2 summarizes results in verification.

Of course, there are other classes of problems that do not fit so easily into the confines of these tables. See [Ski92] for discussions of probing with uncertainty and tactile sensing in robotics.

TABLE 26.2.1 Upper and lower bounds for determination for various probing models.

PROBE	OBJECT	LOWER BOUND	UPPER BOUND
Finger	convex n-gon	$3n$	$3n$
Finger	convex n-gon w known n	$2n+1$	$3n-1$
Finger	convex polyhedra in $\mathbb{E}^d$	$df_0(P)+f_{d-1}(P)$	$f_0(P)+(d+2)f_{d-1}(P)$
Finger	n-gon from m conv models	$n-1$	$n+3$
2 fingers	convex n-gon	$2n-2$	$2n$
3 fingers	convex n-gon	$2n-3$	$2n$
4 or 5 fingers	convex n-gon	$(4n-5)/3$	$\lfloor(4n+2)/3\rfloor$
$k \geq 6$ fingers	convex n-gon	n	$n+1$
Enhanced fingers	n-gon w noncollinear edges	$3n-3$	$3n-3$
Line	convex n-gon	$3n+1$	$3n+1$
Line	n-gon from m conv models	$2n-3$	$2n+4$
Silhouette	convex n-gon	$3n-2$	$3n-2$
Silhouette	convex polyhedra in $\mathbb{E}^3$	$f_2(P)/2$	$5f_0(P)+f_2(P)$
X-ray	convex n-gon	$3n-3$	$5n-19$
Parallel X-ray	convex n-gon	3	3
Parallel X-ray	nondegenerate n-gon	$\lfloor \log n \rfloor - 2$	$2n+2$
Halfplane	convex n-gon	$2n$	$7n+7$
Cut-set	embedded graph	$\binom{n}{2}$	$\binom{n}{2}$
Cut-set	unembedded graph	$\Omega(n^2/\log n)$	$O(n^2/\log n)$

TABLE 26.2.2 Upper and lower bounds for verification for various probing models

PROBE	OBJECT	LOWER BOUND	UPPER BOUND
Finger	convex n-gon	$2n$	$2n$
Finger	convex n-gon with known n	$3\lceil n/2 \rceil$	$3\lceil n/2 \rceil$
Line	convex n-gon	$2n$	$2n$
X-ray	convex n-gon	$3n/2$	$3n/2+6$
Halfplane	convex n-gon	$2n/3$	$n+1$

OPEN PROBLEMS

1. Tighten the gap between the lower and upper bounds for determination for finger probes in higher dimensions [DEY86].

2. Tighten the bounds for determination of convex n-gons with X-ray probes. Does a finite number (i.e., $f(n)$) of parallel X-ray probes suffice to verify or determine simple n-gons? Since each parallel X-ray probe provides a representation of the complete polygon, there is hope to detect arbitrarily small cracks in a finite number of probes; but see [MS93].

3. Consider generalizations of halfplane probes to higher dimensions. How many

probes are necessary to determine convex (or nonconvex) polyhedra?

4. Silhouette probes return the shadow cast by a polytope in a specified direction. These dualize to ***cross-section probes*** that return a slice of the polytope. Tighten the current bounds [DEY86] on determination with silhouettes in $\mathbb{E}^3$.

26.3 ILL-POSED RECONSTRUCTION PROBLEMS

Many geometric reconstruction problems that arise in practice are inherently ill-defined. For example, consider the problem of reconstructing the correct surface given a set of n points on the surface. Fitting an nth degree polynomial to the data set rarely suffices, because there is usually an aesthetic criterion to be satisfied. Perhaps it is possible to give a formal characterization of the desired surface, but more likely heuristics that give "nice" surfaces will be used.

In this section, we mention a class of approximate reconstruction problems, typically inspired by practical problems, and describe a few approaches towards dealing with them. Specific results are not discussed, but pointers to the literature are provided.

COMPUTER VISION

Computer vision is an enormous research area, with the goal of enabling computers to understand and interpret features in digital images. There are a variety of computer vision problems that can be framed as reconstruction problems, particularly those that try to use several fixed images or active sensing, where the robot is free to decide where to look next to obtain more information about its environment [Fau93, Hor86].

A particularly interesting class of active sensing problems involves navigating in an unfamiliar terrain, where we seek a short path to a goal but learn about obstacles only as we encounter them. See [BRS91] for approximation results on this problem, and Sections 40.3 and 41.7 of this Handbook.

Decision trees are a commonly-used classification procedure for recognizing an object drawn from a known class of models. The classification procedure takes the form of a rooted tree, where the models are leaves and each internal node corresponds to a test or probe. All of the probing strategies discussed in previous sections can be reformulated in terms of decision trees, with the goal of minimizing the heights of the trees. The general problem of minimizing the height of a decision tree is NP-complete, but approximation algorithms for minimizing the height of geometric decision trees are known [AMM+93, AGM+93].

SURFACES FROM DATA POINTS

As described in the introduction to this section, interpolating a surface from a finite set of points in three dimensions can be considered a geometric reconstruction problem, albeit ill-posed. These problems often arise in cartographic data, where we seek to construct a model of a mountain given a set of points on the surface. The issue also arises in surface simplification, where given a surface we seek to

approximate it with another surface with fewer points such that the maximum difference between elevations is minimized.

One common approach consists of projecting the points to the plane, triangulating them, and converting this into a triangulated surface by projecting each vertex back into three dimensions. Triangulation-based approaches to surface reconstruction are surveyed in [MSS92].

TOMOGRAPHY AND SURFACES FROM CROSS SECTIONS AND PROJECTIONS

CAT scanners and other tomographic imaging systems represent a tremendous step forward in our ability to diagnose tumors and other medical problems. Herman [Her80] defines tomography as "the process of producing an image of a two-dimensional distribution (usually of some physical property) from estimates of its line integrals along a finite number of lines of known locations." Tomographic scanners estimate line integrals by sending an energy pulse of some type through an object and measuring how much energy is absorbed. Surveys of tomography include [SK78, SSW77].

The most important reconstruction algorithms are transform methods, which are direct implementations of the Radon inversion formula, derived using Fourier transform methods.

An important related geometric problem concerns splicing a series of these parallel slices into polyhedra. The most natural way to proceed is to triangulate between the slices, but this is not always possible without adding extra vertices [GOS96]. Practical algorithms include [Boi88, BS94]; see also Section 23.6.

SHAPE FROM DISTRIBUTION OF CROSS SECTIONS

In such fields as biology and geology, it is often necessary to reconstruct the shape and size distributions of particles from the cross sections of samples. For example, cross sections of cubes can be polygons with 3, 4, 5, or 6 sides, and the probability of each such event is well-defined if the cross sections are taken uniformly at random, as would be the case with small crystals inside a large mineral sample. This has given rise to a field known as ***stereology*** [Eli67, Hau63, Wei83], where such distributions are studied.

3D MODELS FROM 2D IMAGES

In the field of computer vision, it is often desirable to reconstruct a three-dimensional model of an object consistent with one or more two-dimensional images of the object. The model is not necessarily unique, as there may be features that do not appear in any of the images. These problems are surveyed in Section 49.2.

After edge detection has been applied to the image, the primary algorithmic problem concerns identifying whether edges correspond to protrusions or indentations of the main object. Huffman-Clowes labeling is a constraint-based approach resulting from a case analysis of the possible types of junctions and shadows in the scene. Recent articles of such methods include [ABC+90, WG93, Whi89, Sug86].

26.4 SOURCES AND RELATED MATERIAL

SURVEYS

All results not given an explicit reference can be traced through these surveys:

[DLL95]: Survey on embedding proximity graphs (Table 26.1.1).

[Gar95]: Survey of Hammer's X-ray problem and related work in geometric tomography.

[Ski92]: Survey on geometric probing (Table 26.2.1).

RELATED CHAPTERS

Chapter 25: Visibility
Chapter 41: Robotics
Chapter 44: Graph drawing
Chapter 49: Rigidity and scene analysis

REFERENCES

[AB85] P.F. Ash and E.D. Bolker. Recognizing Dirichlet tesselations. *Geom. Dedicata*, 19:175–206, 1985.

[ABC+90] N. Ayache, J.D. Boissonnat, L. Cohen, B. Geiger, J. Levy-Vehel, O. Monga, and P. Sander. Steps toward the automatic interpretation of 3D images. In K.H. Höhne, H. Fuchs, and S.M. Pizer, editors, *3D Imaging in Medicine*, volume 60 of NATO Adv. Sci. Inst. Ser. F: Comput. Systems Sci., pages 107–120. Springer-Verlag, Berlin, 1990.

[ABY90] P.D. Alevizos, J.-D. Boissonnat, and M. Yvinec. Non-convex contour reconstruction. *J. Symbolic Comput.*, 10:225–252, 1990.

[AGM+93] E.M. Arkin, M.T. Goodrich, J.S.B. Mitchell, D. Mount, C.D. Piatko, and S.S. Skiena. Point probe decision trees for geometric concept classes. In *Proc. 3rd Workshop Algorithms Data Struct.*, volume 709 of *Lecture Notes in Comput. Sci.*, pages 95–106. Springer-Verlag, New York, 1993.

[AMM+93] E.M. Arkin, H. Meijer, J.S.B. Mitchell, D. Rappaport, and S.S. Skiena. Decision trees for geometric models. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 369–378, 1993.

[Blo77] G. Bloom. A counterexample to a theorem of Piccard. *J. Comb. Theory Ser. A*, 22:378–379, 1977.

[Boi88] J.-D. Boissonnat. Shape reconstruction from planar cross-sections. *Comput. Vision Graph. Image Process.*, 44:1–29, 1988.

[BRS91] A. Blum, P. Raghavan, and B. Schieber. Navigating in unfamiliar geometric terrain. In *Proc. 23rd Annu. ACM Sympos. Theory Comput.*, pages 494–503, 1991.

[BS94] G. Barrequet and M. Sharir. Piecewise-linear interpolation between polygonal slices. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 93–102, 1994.

[CY87] R. Cole and C.K. Yap. Shape from probing. *J. Algorithms*, 8:19–38, 1987.

[DEY86] D.P. Dobkin, H. Edelsbrunner, and C.K. Yap. Probing convex polytopes. In *Proc. 18th Annu. ACM Sympos. Theory Comput.*, pages 424–432, 1986.

[DLL95] G. Di Battista, W. Lenhart, and G. Liotta. Proximity drawability: a survey. In R. Tamassia and I.G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Comput. Sci.*, pages 328–339. Springer-Verlag, New York, 1995.

[Eli67] H. Elias, editor. *Proc. 2nd Internat. Congress Stereology.* Springer-Verlag, New York, 1967.

[Fau93] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint.* MIT Press, Cambridge, 1993.

[Gar95] R.J. Gardner. *Geometric Tomography.* Cambridge University Press, Cambridge, 1995.

[GH95] P. Gritzmann and A. Hufnagel. A polynomial time algorithm for Minkowski reconstruction. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 1–9, 1995.

[GOS96] C. Gitlin, J. O'Rourke, and V. Subramanian. On reconstructing polyhedra from parallel slices. *Internat. J. Comput. Geom. Appl.*, 6:103–122, 1996.

[Hau63] H. Haug, editor. *Proc. 1st Internat. Congress Stereology.* Druck Congressprint, Kaunitzgasse, 1963.

[Her80] G.T. Herman. *Image Reconstruction from Projections: the Fundamentals of Computerized Tomography.* Academic Press, New York, 1980.

[Hor86] B.K.P. Horn. *Robot Vision.* MIT Press, Cambridge, 1986.

[MS93] H. Meijer and S. Skiena. Reconstructing polygons from X-rays. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 381–386, Waterloo, 1993.

[MSS92] D. Meyers, S. Skinner, and K. Sloan. Surfaces from contours. *ACM Trans. Graph.*, 11:228–258, 1992.

[O'R88] J. O'Rourke. Uniqueness of orthogonal connect-the-dots. In G.T. Toussaint, editor, *Computational Morphology*, pages 97–104. North-Holland, Amsterdam, 1988.

[SK78] L.A. Shepp and J.B. Kruskal. Computerized tomography: the new medical X-ray technology. *Amer. Math. Monthly*, 85:420–439, 1978.

[Ski89] S.S. Skiena. Problems in geometric probing. *Algorithmica*, 4:599–605, 1989.

[Ski92] S.S. Skiena. Interactive reconstruction via geometric probing. *Proc. IEEE*, 80:1364–1383, 1992.

[SSL90] S.S. Skiena, W.D. Smith, and P. Lemke. Reconstructing sets from interpoint distances. In *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, pages 332–339, 1990.

[SSW77] K.T. Smith, D.C. Solomon, and D.L. Wagner. Practical and mathematical aspects of the problem of reconstructing objects from radiographs. *Bull. Amer. Math. Soc.*, 83:1227–1270, 1977.

[Sug86] K. Sugihara. *Machine Interpretation of Line Drawings.* MIT Press, Cambridge, 1986.

[Wei83] W. Weil. Stereology: A survey for geometers. In P. Gruber and J. Wills, editors, *Convexity and Its Applications*, pages 360–412. Birkhauser, Basel, 1983.

[WG93] W. Wang and G.G. Grinstein. A survey of 3D solid reconstruction from 2D projection line drawings. *Comput. Graph. Forum*, 12:137–158, 1993.

[Whi89] W. Whiteley. A matroid on hypergraphs, with applications in scene analysis and geometry. *Discrete Comput. Geom.*, 4:75–95, 1989.

27 COMPUTATIONAL CONVEXITY

Peter Gritzmann and Victor Klee

INTRODUCTION

The subject of Computational Convexity draws its methods from discrete mathematics and convex geometry, and many of its problems from operations research, computer science, and other applied areas. In essence, it is the study of the computational and algorithmic aspects of high-dimensional convex sets (especially polytopes), with a view to applying the knowledge gained to convex bodies that arise in other mathematical disciplines or in the mathematical modeling of problems from outside mathematics.

The name *Computational Convexity* is of recent origin, having first appeared in print in 1989. However, results that retrospectively belong to this area go back a long way. In particular, many of the basic ideas of *Linear Programming* have an essentially geometric character and fit very well into the conception of Computational Convexity. The same is true of the subject of *Polyhedral Combinatorics* and of the *Algorithmic Theory of Polytopes and Convex Bodies.*

The emphasis in Computational Convexity is on problems whose underlying structure is the convex geometry of normed vector spaces of finite but generally *not* restricted dimension, rather than of fixed dimension. This leads to closer connections with the optimization problems that arise in a wide variety of disciplines. Further, in the study of Computational Convexity, the underlying model of computation is mainly the binary (Turing machine) model that is common in studies of computational complexity. This requirement is imposed by prospective applications, particularly in mathematical programming. For the study of algorithmic aspects of convex bodies that are not polytopes, the binary model is often augmented by additional devices called "oracles." Some cases of interest involve other models of computation, but the present discussion focuses on aspects of computational convexity for which binary models seem most natural. Many of the results stated in this chapter are qualitative, in the sense that they classify certain problems as being solvable in polynomial time, or show that certain problems are NP-hard or harder. The tasks remain to find optimal exact algorithms for the problems that are polynomially solvable, and to find useful approximation algorithms or heuristics for those that are NP-hard. In most cases, the known algorithms, even when they run in polynomial time, appear to be far from optimal from the viewpoint of practical application. Hence, the qualitative complexity results should in many cases be regarded as a guide to future efforts but not as final words on the problems with which they deal.

Some of the important areas of computational convexity, such as linear and convex programming, polyhedral combinatorics, packing and covering, and pattern recognition, are covered in other chapters of this Handbook. Hence, after some remarks on presentations of polytopes in Section 27.1, the present discussion concentrates on areas that are not covered elsewhere in the Handbook. The

following sections are closely related to classical convex geometry: 27.2, Algorithmic Theory of Convex Bodies; 27.3, Volume Computations; 27.4, Mixed Volumes; 27.5, Containment Problems; 27.6, Radii. (Other such areas are geometric tomography [Gar95, GG94], discrete tomography [GG], and the computational aspects of the following topics: projections of polytopes [Fil90, BGK96], sections of polytopes [Fil92], Minkowski addition of polytopes [GS93], and the Minkowski reconstruction of polytopes [GH].) The final section, 27.7, Interval Matrices and Qualitative Matrices, is included as an illustration of material that, though not related to classical convex geometry, nevertheless falls under the general conception of computational convexity.

Because of the diversity of topics covered in this chapter, each section has a separate bibliography.

FURTHER READING

[GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, 1979.

[GK93b] P. Gritzmann and V. Klee. Mathematical programming and convex geometry. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, Volume A, pages 627–674. North-Holland, Amsterdam, 1993.

[GK94a] P. Gritzmann and V. Klee. On the complexity of some basic problems in computational convexity: I. Containment problems. *Discrete Math*, 136:129–174, 1994. Reprinted in W. Deuber, H.-J. Prömel, and B. Voigt, editors, *Trends in Discrete Mathematics*, pages 129–174. *Topics in Discrete Math.*, North-Holland, Amsterdam, 1994.

[GK94b] P. Gritzmann and V. Klee. On the complexity of some basic problems in computational convexity: II. Volume and mixed volumes. In T. Bisztriczky, P. McMullen, R. Schneider, and A. Ivić Weiss, editors, *Polytopes: Abstract, Convex and Computational*, volume 440 of NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci., pages 373–466. Kluwer, Dordrecht, 1994.

RELATED CHAPTERS

Chapter 7: Lattice points and lattice polytopes
Chapter 13: Basic properties of convex polytopes
Chapter 39: Mathematical programming

REFERENCES

[BGK96] T. Burger, P. Gritzmann, and V. Klee. Polytope projection and projection polytopes. *Amer. Math. Monthly*, 103:742–755, 1996.

[Fil90] P. Filliman. Exterior algebra and projections of polytopes. *Discrete Comput. Geom.*, 5:305–322, 1990.

[Fil92] P. Filliman. Volumes of duals and sections of polytopes. *Mathematika*, 39:67–80, 1992.

[Gar95] R.J. Gardner. *Geometric Tomography.* Cambridge University Press, New York, 1995.

[GG94] R.J. Gardner and P. Gritzmann. Successive determination and verification of polytopes by their X-rays. *J. London Math. Soc.*, 50:375–391, 1994.

[GG] R.J. Gardner and P. Gritzmann. Discrete tomography: Determination of finite sets by X-rays. *Trans. Amer. Math. Soc.*, to appear.

[GH] P. Gritzmann and A. Hufnagel. On the algorithmic complexity of Minkowski's reconstruction theorem. *J. London Math. Soc.*, to appear.

[GS93] P. Gritzmann and B. Sturmfels. Minkowski addition of polytopes: computational complexity and applications to Gröbner bases. *SIAM J. Discrete Math.*, 6:246–269, 1993.

27.1 PRESENTATIONS OF POLYTOPES

A convex polytope $P \subset \mathbb{R}^n$ can be represented in terms of its vertices or in terms of its facet inequalities. From a theoretical viewpoint, the two possibilities are equivalent. However, as the dimension increases, the number of vertices can grow exponentially in terms of the number of facets, and vice-versa, so that different presentations may lead to different classifications concerning polynomial-time computability or NP-hardness. (See Sections 13.1, 15.3, and 19.3.)

For algorithmic purposes it is usually not the polytope P as a *geometric* object that is relevant, but rather its *algebraic presentation.* The discussion here is based mainly on the *binary* or *Turing machine* model of computation, in which the *size of the input* is defined as the length of the binary encoding needed to present the input data to a Turing machine and the *time-complexity* of an algorithm is also defined in terms of the operations of a Turing machine. Hence the algebraic presentation of the objects at hand must be finite.

Among important special classes of polytopes, the zonotopes are particularly interesting because they can be so compactly presented.

GLOSSARY

Convex body in $\mathbb{R}^n$: A compact convex subset of $\mathbb{R}^n$.

$\mathcal{K}^n$: The family of all convex bodies in $\mathbb{R}^n$.

Proper convex body in $\mathbb{R}^n$: A convex body in $\mathbb{R}^n$ with nonempty interior.

Polytope: A convex body that has only finitely many extreme points.

$\mathcal{P}^n$: The family of all convex polytopes in $\mathbb{R}^n$.

n-polytope: Polytope of dimension n.

Face of a polytope P: P itself, the empty set, or the intersection of P with some supporting hyperplane.

Facet of an n-polytope P: Face of dimension $n-1$.

Simple n-polytope: Each vertex is incident to precisely n edges or, equivalently, to precisely n facets.

Simplicial polytope: A polytope in which each facet is a simplex.

$\mathcal{V}$-presentation of a polytope P: A string $(n, m; v_1, \ldots, v_m)$, where $n, m \in \mathbb{N}$ and $v_1, \ldots, v_m \in \mathbb{R}^n$ such that $P = \text{conv}\{v_1, \ldots, v_m\}$.

$\mathcal{H}$-presentation of a polytope P: A string $(n, m; A, b)$, where $n, m \in \mathbb{N}$, A is a real $m \times n$ matrix, and $b \in \mathbb{R}^m$ such that $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$

$\mathcal{V}$-polytope P: A string $(n, m; v_1, \ldots, v_m)$, where $n, m \in \mathbb{N}$ and $v_1, \ldots, v_m \in \mathbb{Q}^n$. P is usually identified with the geometric object $\text{conv}\{v_1, \ldots, v_m\}$.

$\mathcal{H}$-polytope P: A string $(n, m; A, b)$, where $n, m \in \mathbb{N}$, A is a rational $m \times n$ matrix, $b \in \mathbb{Q}^m$, and the set $\{x \in \mathbb{R}^n \mid Ax \leq b\}$ is bounded. P is usually identified with this set.

Size of a $\mathcal{V}$- or an $\mathcal{H}$-polytope P: Number of binary digits needed to encode the string $(n, m; v_1, \ldots, v_m)$ or $(n, m; A, b)$, respectively.

Zonotope: The vector sum (Minkowski sum) of a finite number of line segments; equivalently, a polytope of which each face has a center of symmetry.

$\mathcal{S}$-presentation of a zonotope Z in $\mathbb{R}^n$: A string $(n, m; c; z_1, \ldots, z_m)$, where $n, m \in \mathbb{N}$ and $c, z_1, \ldots, z_m \in \mathbb{R}^n$, such that $Z = c + \sum_{i=1}^{m} [-1, 1] z_i$.

Parallelotope in $\mathbb{R}^n$: A zonotope $Z = c + \sum_{i=1}^{m} [-1, 1] z_i$, with $z_1, \ldots, z_m$ linearly independent.

$\mathcal{S}$-zonotope Z in $\mathbb{R}^n$: A string $(n, m; c; z_1, \ldots, z_m)$, where $n, m \in \mathbb{N}$ and $c, z_1, \ldots, z_m \in \mathbb{Q}^n$. Z is usually identified with the geometric object $c + \sum_{i=1}^{m} [-1, 1] z_i$.

27.1.1 CONVERSION OF ONE PRESENTATION INTO THE OTHER

The following results indicate the difficulties that may be expected in converting the $\mathcal{H}$-presentation of a polytope into a $\mathcal{V}$-presentation or vice versa.

For $\mathcal{H}$-presented n-polytopes with m facets, the *maximum* possible number of vertices is

$$\mu(m, n) = \binom{m - \lfloor (n+1)/2 \rfloor}{m - n} + \binom{m - \lfloor (n+2)/2 \rfloor}{m - n},$$

and this is also the maximum possible number of facets for a $\mathcal{V}$-presented n-polytope with m vertices. The first maximum is attained within the family of simple n-polytopes, the second within the family of simplicial n-polytopes.

When n is fixed, the number of vertices is bounded by a polynomial in the number of facets, and vice-versa, and it is possible to pass from either sort of presentation to the other in polynomial time. However, the degree of the polynomial goes to infinity with n. A consequence of this is that when the dimension n is permitted to vary in a problem concerning polytopes, the manner of presentation is often influential in determining whether the problem can be solved in polynomial time or is NP-hard. For the case of variable dimension, it is #P-hard even to determine the number of facets of a given $\mathcal{V}$-polytope, or to determine the number of vertices of a given $\mathcal{H}$-polytope.

For *simple* $\mathcal{H}$-presented n-polytopes with m facets, the *minimum* possible number of vertices is $(m - n)(n - 1) + 2$. The large gap between this number and the above sum of binomial coefficients makes it clear that, from a practical standpoint, the worst-case behavior of any conversion algorithm should be measured in terms of *both* input size and output size.

The maximum number of j-dimensional faces of an n-dimensional zonotope formed as the sum of m segments is

$$2\binom{m}{j} \sum_{k=0}^{n-1-j} \binom{m-1-j}{k},$$

and hence, the number of vertices or of facets (or of faces of any dimension) of an $\mathcal{S}$-zonotope is not bounded by any polynomial in the size of the $\mathcal{S}$-presentation.

FURTHER READING

[BL93] M. Bayer and C. Lee. Combinatorial aspects of convex polytopes. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, Volume A, pages 251–305. North-Holland, Amsterdam, 1993.

[Brø83] A. Brøndsted. *An Introduction to Convex Polytopes.* Springer-Verlag, New York, 1983.

[GK94a] P. Gritzmann and V. Klee. On the complexity of some basic problems in computational convexity: I. Containment problems. *Discrete Math*, 136:129–174, 1994. Reprinted in W. Deuber, H.-J. Prömel, and B. Voigt, editors, *Trends in Discrete Mathematics*, pages 129–174. *Topics in Discrete Math.*, North-Holland, Amsterdam, 1994.

[Grü67] B. Grünbaum. *Convex Polytopes.* Wiley-Interscience, London, 1967.

[KK95] V. Klee and P. Kleinschmidt. Convex polytopes and related complexes. In R.L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, Volume I., pages 875–917. North-Holland, Amsterdam, 1995.

[MS71] P. McMullen and G.C. Shephard. *Convex Polytopes and the Upper Bound Conjecture.* Cambridge University Press, 1971.

[Zie94] G.M. Ziegler. *Lectures on Polytopes.* Volume 152 of *Graduate Texts in Math.*, Springer-Verlag, New York, 1995.

RELATED CHAPTERS

Chapter 13: Basic properties of convex polytopes
Chapter 15: Face numbers of polytopes and complexes
Chapter 19: Convex hull computations

27.2 ALGORITHMIC THEORY OF CONVEX BODIES

Polytopes may be $\mathcal{V}$-presented or $\mathcal{H}$-presented. However, a different approach is required to deal with convex bodies K that are not polytopes, since an enumeration of all the extreme points of K or of its polar is not possible. A convenient way to deal with the general situation is to assume that the convex body in question is given by an algorithm (called an *oracle*) that answers certain sorts of questions about the body. A small amount of a priori information about the body may be known, but aside from this, all information about the specific convex body must be obtained from the oracle, which functions as a "black box." In other words, while it is assumed that the oracle's answers are always correct, nothing is assumed about the manner in which it produces those answers. The algorithmic theory of convex bodies was developed in [GLS88] with a view to proper (i.e., n-dimensional) convex bodies in $\mathbb{R}^n$. For many purposes, provisions can be made to deal meaningfully with improper bodies as well, but that aspect is largely ignored in what follows.

GLOSSARY

Outer parallel body of a convex body K: $K(\epsilon) = K + \epsilon B^n$, where B^n is the Euclidean unit ball in $\mathbb{R}^n$.

Inner parallel body of a convex body K: $K(-\epsilon) = K \setminus \big((\mathbb{R}^n \setminus K) + \epsilon B^n\big)$.

Weak membership problem for a convex body K in $\mathbb{R}^n$: Given $y \in \mathbb{Q}^n$, and a rational number $\epsilon > 0$, conclude with one of the following: *assert that* $y \in K(\epsilon)$; or *assert that* $y \notin K(-\epsilon)$.

Weak separation problem for a convex body K in $\mathbb{R}^n$: Given a vector $y \in \mathbb{Q}^n$, and a rational number $\epsilon > 0$, conclude with one of the following: *assert that* $y \in K(\epsilon)$; or *find a vector* $z \in \mathbb{Q}^n$ *such that* $\| z \|_\infty = 1$ *and* $z^T x < z^T y + \epsilon$ *for every* $x \in K(-\epsilon)$.

Weak (linear) optimization problem for a convex body K in $\mathbb{R}^n$: Given a vector $c \in \mathbb{Q}^n$ and a rational number $\epsilon > 0$, conclude with one of the following: *find a vector* $y \in \mathbb{Q}^n \cap K(\epsilon)$ *such that* $c^T x \leq c^T y + \epsilon$ *for every* $x \in K(-\epsilon)$; or *assert that* $K(-\epsilon) = \emptyset$.

Circumscribed convex body K: A positive rational number R is given explicitly such that $K \subset RB^n$.

Well-bounded convex body K: Positive rational numbers r, R are given explicitly such that $K \subset RB^n$ and K contains a ball of radius r.

Centered well-bounded convex body K: Positive rational numbers r, R and a vector $b \in \mathbb{Q}^n$ are given explicitly such that $b + rB^n \subset K$ and $K \subset RB^n$.

Weak membership oracle for a convex body K: Algorithm that solves the weak membership problem for K.

Weak separation oracle for K: Algorithm that solves the weak separation problem for K.

Weak (linear) optimization oracle for K: Algorithm that solves the weak (linear) optimization problem for K.

The three problems above are very closely related in the sense that when the classes of proper convex bodies are appropriately restricted to those that are circumscribed, well-bounded, or centered, and when input sizes are properly defined, an algorithm that solves any one of the problems in polynomial time can be used as a subroutine to solve the others in polynomial time also. The definition of input size involves the size of ϵ, the dimension of K, the given a priori information ($size(r)$, $size(R)$, and/or $size(b)$), and the input required by the oracle. The following theorem of [GLS88] contains a list of the precise relationships among the three basic oracles for proper convex bodies. The notation "$(\mathcal{A}; prop) \rightarrow_\pi \mathcal{B}$" indicates the existence of an (oracle-) polynomial-time algorithm that solves problem $\mathcal{B}$ for every proper convex body that is given by the oracle $\mathcal{A}$ and has all the properties specified in *prop*. ($prop = \emptyset$ means that the statement holds for general proper convex bodies.)

(Weak Membership; centered, well-bounded) $\rightarrow_\pi$ Weak Separation;
(Weak Membership; centered, well-bounded) $\rightarrow_\pi$ Weak Optimization;
(Weak Separation; $\emptyset$) $\rightarrow_\pi$ Weak Membership;
(Weak Separation; circumscribed) $\rightarrow_\pi$ Weak Optimization;

(WEAK OPTIMIZATION; $\emptyset$) $\rightarrow_\pi$ WEAK MEMBERSHIP;
(WEAK OPTIMIZATION; $\emptyset$) $\rightarrow_\pi$ WEAK SEPARATION.

It should be emphasized that there are polynomial-time algorithms that, accepting as input a set P that is a proper $\mathcal{V}$-polytope, a proper $\mathcal{H}$-polytope, or a proper $\mathcal{S}$-zonotope, produce membership, separation, and optimization oracles for P, and also compute a lower bound on the inradius of P, an upper bound on its circumradius, and a "center" b_P for P. This implies that if an algorithm performs certain tasks for convex bodies given by some of the above (appropriately specified) oracles, then the same algorithm can also serve as a basis for procedures that perform these tasks for $\mathcal{V}$- or $\mathcal{H}$-polytopes and for $\mathcal{S}$-zonotopes. Hence the oracular framework, in addition to being applicable to convex bodies that are not polytopes, serves also to modularize the approach to algorithmic aspects of polytopes. On the other hand, there are lower bounds on the performance of approximate algorithms for the oracle model that do not carry over to the case of $\mathcal{V}$- or $\mathcal{H}$-polytopes or $\mathcal{S}$-zonotopes [BF87, BGK96].

FURTHER READING

[GLS88] M. Grötschel, L. Lovász, and A. Schriver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, Berlin, 1988, 1993.

RELATED CHAPTERS

Chapter 7: Lattice points and lattice polytopes

REFERENCES

[BF87] I. Bárány and Z. Füredi. Computing the volume is difficult. *Discrete Comput. Geom.*, 2:319–326, 1987.

[BGK96] A. Brieden, P. Gritzmann, and V. Klee. Oracle-polynomial-time approximation algorithms for norm-maximization and radii-computation. Manuscript, 1996.

27.3 VOLUME COMPUTATIONS

It may be fair to say that the modern study of volume computations began with Kepler [Kep15] who derived the first *cubature formula* for measuring the capacities of wine barrels, and that it was the task of volume computation that motivated the general field of integration. The problem of computing or approximating volumes of convex bodies is certainly one of the basic problems in mathematics.

GLOSSARY

In the following, G is a subgroup of the group of all affine automorphisms of $\mathbb{R}^n$.

Dissection of an n-polytope P into n-polytopes $P_1, \ldots, P_k$: $P = P_1 \cup \ldots \cup P_k$, where the polytopes P_i have pairwise disjoint interiors.

Polytopes $P, Q \subset \mathbb{R}^n$ are ***G-equidissectable:*** For some k there exist dissections $P_1, \ldots, P_k$ of P and $Q_1, \ldots, Q_k$ of Q, and elements $g_1, \ldots, g_k$ of G, such that $P_i = g_i(Q_i)$ for all i.

Polytopes $P, Q \subset \mathbb{R}^n$ are ***G-equicomplementable:*** There are polytopes P_1, P_2 and Q_1, Q_2 such that P_2 is dissected into P and P_1, Q_2 is dissected into Q and Q_1, P_1 and Q_1 are G-equidissectable, and P_2 and Q_2 are G-equidissectable.

Decomposition of a set S: $S = S_1 \cup \ldots \cup S_k$, where the sets S_i are pairwise disjoint.

Sets S, T are ***G-equidecomposable:*** For some k there are decompositions $S_1, \ldots, S_k$ of S and $T_1, \ldots, T_k$ of T, and elements $g_1, \ldots, g_k$ of G, such that $S_i = g_i(T_i)$ for all i.

Valuation on a family $\mathcal{S}$ of subsets of $\mathbb{R}^n$: A functional $\varphi : \mathcal{S} \to \mathbb{R}$ with the property that $\varphi(S_1) + \varphi(S_2) = \varphi(S_1 \cup S_2) + \varphi(S_1 \cap S_2)$ whenever the sets $S_1, S_2, S_1 \cup S_2, S_1 \cap S_2 \in \mathcal{S}$.

G-invariant valuation φ: $\varphi(S) = \varphi(g(S))$ for all $S \in \mathcal{S}$ and $g \in G$.

Simple valuation φ: $\varphi(S) = 0$ whenever $S \in \mathcal{S}$ and S is contained in a hyperplane.

Monotone valuation φ: $\varphi(S_1) \leq \varphi(S_2)$ whenever $S_1, S_2 \in \mathcal{S}$ with $S_1 \subset S_2$.

Class $\mathcal{P}$ of $\mathcal{H}$-polytopes is ***near-simplicial:*** There is a nonnegative integer σ such that $\mathcal{P} = \bigcup_{n \in \mathbb{N}} \mathcal{P}_{\mathcal{H}}(n, \sigma)$, where $\mathcal{P}_{\mathcal{H}}(n, \sigma)$ is the family of all n-dimensional $\mathcal{H}$-polytopes P in $\mathbb{R}^n$ such that each facet of P has at most $n + 1 + \sigma$ vertices.

Class $\mathcal{P}$ of $\mathcal{V}$-polytopes is ***near-simple:*** There is a nonnegative integer τ such that $\mathcal{P} = \bigcup_{n \in \mathbb{N}} \mathcal{P}_{\mathcal{V}}(n, \tau)$, where $\mathcal{P}_{\mathcal{V}}(n, \tau)$ is the family of all n-dimensional $\mathcal{V}$-polytopes P in $\mathbb{R}^n$ such that each vertex of P is incident to at most $n + \tau$ edges.

Class $\mathcal{P}$ of $\mathcal{V}$-polytopes is ***near-parallelotopal:*** There is a nonnegative integer ζ such that $\mathcal{Z} = \bigcup_{n \in \mathbb{N}} \mathcal{Z}_{\mathcal{S}}(n, \zeta)$, where $\mathcal{Z}_{\mathcal{S}}(n, \zeta)$ is the family of all $\mathcal{S}$-zonotopes in $\mathbb{R}^n$ that are represented as the sum of at most $n + \zeta$ segments.

V: The functional that associates with a convex body K its volume.

$\mathcal{H}$-VOLUME: For a given $\mathcal{H}$-polytope P and a nonnegative rational ν, decide whether $V(P) \leq \nu$.

$\mathcal{V}$-VOLUME, $\mathcal{S}$-VOLUME: Similarly for $\mathcal{V}$-polytopes and $\mathcal{S}$-zonotopes.

λ-APPROXIMATION for some functional ρ: Given a positive integer n and a well-bounded convex body K given by a weak separation oracle, determine a positive rational μ such that

$$\frac{\rho(K)}{\mu} \leq 1 + \lambda \quad \text{and} \quad \frac{\mu}{\rho(K)} \leq 1 + \lambda.$$

EXPECTED VOLUME COMPUTATION: Given a positive integer n, a centered well-bounded convex body K in $\mathbb{R}^n$ given by a weak membership oracle, and positive rationals β and ϵ. Determine a positive rational random variable μ such that

$$\text{prob}\left\{\left|\frac{\mu}{V(K)} - 1\right| \leq \epsilon\right\} \geq 1 - \beta.$$

27.3.1 CLASSICAL BACKGROUND, CHARACTERIZATIONS

The results in this subsection connect the subject matter of volume computation with related "classical" problems. In the following, G is a group of affine automorphisms of $\mathbb{R}^n$, as above, and D is the group of isometries.

(i) Two polytopes are G-equidissectable if and only if they are G-equicomplementable.

(ii) Two polytopes P and Q are G-equidissectable if and only if $\varphi(P) = \varphi(Q)$ for all G-invariant simple valuations on $\mathcal{P}^n$.

(iii) Two plane polygons are of equal area if and only if they are D-equidissectable.

(iv) If one agrees that an a-by-b rectangle should have area ab, and also agrees that the area function should be a D-invariant simple valuation, it then follows from the preceding result that the area of any plane polygon P can be determined (at least in theory) by finding a rectangle R to which P is equidissectable. This provides a satisfyingly geometric theory of area that does not require any limiting considerations. The third problem of Hilbert [Hil00] asked, in effect, whether such a result extends to 3-polytopes. A negative answer was supplied by [Deh00], who showed that a regular tetrahedron and a cube are not equidissectable.

(v) If P and Q are n-polytopes in $\mathbb{R}^n$, then for P and Q to be equidissectable under the group of all isometries of $\mathbb{R}^n$, it is necessary that $f^*(P) = f^*(Q)$ for each additive real function f such that $f(\pi) = 0$. (Here $f^*(P)$ is the so-called *Dehn invariant* of P associated with f.) The condition is also sufficient for equidissectability when $n \leq 4$, but the matter of sufficiency is unsettled for $n \geq 5$.

(vi) Two plane polygons are of equal area if and only if they are D-equidecomposable.

(vii) In [Lac90], it was proved that any two plane polygons of equal area are equidecomposable under the group of translations. That paper also settled Tarski's old problem of "squaring the circle" by showing that a square and a circular disk of equal area are equidecomposable; there too, translations suffice. On the other hand, a disk and a square cannot be *scissors congruent*; i.e., there is no equidissection (with respect to rigid motions) into pieces that, roughly speaking, could be cut out with a pair of scissors.

(viii) If X and Y are bounded subsets of $\mathbb{R}^n$ (with $n \geq 3$), and each set has nonempty interior, then X and Y are D-equidecomposable. This is the famous ***Banach-Tarski paradox***.

(ix) Under the group of all volume-preserving affinites of $\mathbb{R}^n$, two n-polytopes are equidissectable if and only if they are of equal volume.

(x) If φ is a translation-invariant, nonnegative, simple valuation on $\mathcal{P}^n$ (resp. $\mathcal{K}^n$), then there exists a nonnegative real α such that $\varphi = \alpha V$.

(xi) A translation-invariant valuation on $\mathcal{P}^n$ that is homogeneous of degree n is a constant multiple of the volume.

(xii) A continuous, rigid-motion-invariant, simple valuation on $\mathcal{K}^n$ is a constant multiple of the volume.

(xiii) A nonnegative simple valuation on $\mathcal{P}^n$ (resp. $\mathcal{K}^n$) that is invariant under all volume-preserving linear maps of $\mathbb{R}^n$ is a constant multiple of the volume.

27.3.2 SOME VOLUME FORMULAS

Since simplex volumes can be computed so easily, the most natural approach to the problem of computing the volume of a polytope P is to produce a *triangulation* of P (see Chapter 14). Then compute the volumes of the individual simplices and add them up to find the volume of P. (This uses the fact that the volume is a simple valuation.) As a simple consequence, one sees that when the dimension n is fixed, the volume of $\mathcal{V}$-polytopes and of $\mathcal{H}$-polytopes can be computed in polynomial time.

Another equally natural method is to dissect P into pyramids with common apex over its facets. Since the volume of such a pyramid is just $1/(n!)$ times the product of its height and the $(n-1)$-volume of its base, the volume can be computed recursively.

Another approach that has become a standard tool for many algorithmic questions in geometry is the *sweep-plane* technique. The general idea is to "sweep" a hyperplane through a polytope P, keeping track of the changes that occur when the hyperplane sweeps through a vertex. As applied to volume computation, this leads to the volume formula given below that does not explicitly involve triangulations, [BN83, Law91].

Suppose that $(n, m; A, b)$ is an irredundant $\mathcal{H}$-presentation of a simple polytope P (see Section 19.2). Let $b = (\beta_1, \ldots, \beta_m)^T$ and denote the row-vectors of A by $a_1^T, \ldots, a_m^T$. Let $M = \{1, \ldots, m\}$ and for each nonempty subset I of M, let A_I denote the submatrix of A formed by rows with indices in I and let b_I denote the corresponding right-hand side. Let $\mathcal{F}_0(P)$ denote the set of all vertices of the polytope $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$. For each $v \in \mathcal{F}_0(P)$, there is a set $I = I_v \subset M$ of cardinality n such that $A_I v = b_I$ and $A_{M \setminus I} v \leq b_{M \setminus I}$. Since P is assumed to be simple and its $\mathcal{H}$-presentation to be irredundant, the set I_v is unique.

Let $c \in \mathbb{R}^n$ be such that $\langle c, v_1 \rangle \neq \langle c, v_2 \rangle$ for any pair of vertices v_1, v_2 that form an edge of P. Then it turns out that

$$V(P) = \frac{1}{n!} \sum_{v \in \mathcal{F}_0(P)} \frac{\langle c, v \rangle^n}{\prod_{i=1}^n e_i^T A_{I_v}^{-1} c |\det(A_{I_v})|}.$$

The ingredients of this volume formula are those that are computed in the (dual) simplex algorithm. More precisely, $\langle c, v \rangle$ is just the value of the objective function at the current basic feasible solution v, $\det(A_{I_v})$ is the determinant of the current basis, and $A_{I_v}^{-1} c$ is the vector of reduced costs, i.e., the (generally infeasible) dual point that belongs to v.

For practical computations, this volume formula has to be combined with some vertex enumeration technique. Its closeness to the simplex algorithm suggests the

use of a *reverse search* method [AF92], which is based on the simplex method with Bland's pivoting rule.

As it stands, the volume formula does not involve triangulation. However, when interpreted in a polar setting, it is seen to involve the faces of the simplicial polytope P° that is the polar of P. Accordingly, generalization to nonsimple polytopes involves polar triangulation. In fact, for general polytopes P, one may apply a "lexicographic rule" for moving from one basis to another, but this amounts to a particular triangulation of P°.

Another possibility for computing the volume of a polytope P is to study the *exponential integral* $\int_P e^{\langle c,x\rangle} dx$, where c is an arbitrary vector of $\mathbb{R}^n$; see [Bar93]. (Note that for $c = 0$, this integral just gives the volume of P.) Exponential integrals satisfy certain relations that make it possible to compute the integrals efficiently in some important cases. In particular, exponential sums can be used to obtain the tractability result for "near-simple" $\mathcal{V}$-polytopes stated in the next subsection.

27.3.3 TRACTABILITY RESULTS

The volume of a polytope P can be computed in polynomial time in the following cases:

(i) when the dimension is fixed and P is a $\mathcal{V}$-polytope, an $\mathcal{H}$-polytope, or an $\mathcal{S}$-zonotope;

(ii) when the dimension is part of the input and P is a near-simple $\mathcal{V}$-polytope, a near-simplicial $\mathcal{H}$-polytope, or a near-parallelotopal $\mathcal{S}$-zonotope.

27.3.4 INTRACTABILITY RESULTS

(i) There is no polynomial-space algorithm for exact computation of the volume of $\mathcal{H}$-polytopes.

(ii) $\mathcal{H}$-VOLUME is #P-hard even for the intersections of the unit cube with one rational halfspace.

(iii) $\mathcal{H}$-VOLUME is #P-hard in the strong sense. (This follows from the result of [BW92] that the problem of computing the number of linear extensions of a given partially ordered set $\mathcal{O} = (\{1, \ldots, n\}, \prec)$ is #P-complete, in conjunction with the fact that this number is equal to $n!V(P_\mathcal{O})$, where the set $P_\mathcal{O} = \{x = (\xi_1, \ldots, \xi_n)^T \in [0,1]^n \mid \xi_i \leq \xi_j \iff i \prec j\}$ is the *order polytope* of $\mathcal{O}$ [Sta86].)

(iv) The problem of computing the volume of the convex hull of the regular $\mathcal{V}$-cross-polytope and an additional integer vector is #P-hard.

(v) $\mathcal{S}$-VOLUME is #P-hard.

27.3.5 DETERMINISTIC APPROXIMATION

(i) There exists an oracle-polynomial-time algorithm that, for any convex body K of $\mathbb{R}^n$ given by a weak optimization oracle, and for each $\epsilon > 0$, finds rationals μ_1 and μ_2 such that

$$\mu_1 \leq V(K) \leq \mu_2 \quad \text{and} \quad \mu_2 \leq n!(1+\epsilon)^n \mu_1.$$

(ii) Suppose that

$$\lambda(n) < \left(\frac{n}{\log n}\right)^{n/2} - 1 \qquad \text{for all } n \in \mathbb{N}.$$

Then there exists no deterministic oracle-polynomial-time algorithm for λ-APPROXIMATION of the volume [BF87].

27.3.6 RANDOMIZED ALGORITHMS

[DFK89] proved that there is a randomized algorithm for EXPECTED VOLUME COMPUTATION that runs in time that is oracle-polynomial in n, $1/\epsilon$, and $\log(1/\beta)$.

The first step is a rounding procedure, using an algorithmic version of John's theorem; see Section 27.5.4. For the second step, one may therefore assume that $B^n \subset K \subset (n+1)\sqrt{n}B^n$. Now, let

$$k = \left\lceil \frac{3}{2}(n+1)\log(n+1) \right\rceil, \qquad \text{and} \qquad K_i = K \cap \left(1+\frac{1}{n}\right)^i B^n \quad \text{for } i = 0, \ldots, k.$$

Then it suffices to estimate each ratio $V(K_i)/V(K_{i-1})$ up to a relative error of order $\epsilon/(n \log n)$ with error probability of order $\beta/(n \log n)$.

The main step of the algorithm of [DFK89] is based on a method for sampling nearly uniformly from within certain convex bodies K_i. It superimposes a chessboard grid of small cubes (say of edge length δ) on K_i, and performs a random walk over the set $\mathcal{C}_i$ of cubes in this grid that intersect a suitable parallel body $K_i + \alpha B^n$, where α is small. This walk is performed by moving through a facet with probability $1/f_{n-1}(C_n) = (2n)^{-1}$ if this move ends up in a cube of $\mathcal{C}_i$, and staying at the current cube if the move would lead outside of $\mathcal{C}_i$. The random walk gives a *Markov chain* that is irreducible (since the moves are connected), aperiodic, and hence ergodic. But this implies that there is a unique stationary distribution, the limit distribution of the chain, which is easily seen to be a *uniform distribution.* Thus after a sufficiently large (but polynomially bounded) number of steps, the current cube in the random walk can be used to sample nearly uniformly from $\mathcal{C}_i$. Having obtained such a uniformly sampled cube, one determines whether it belongs to $\mathcal{C}_{i-1}$ or to $\mathcal{C}_i \setminus \mathcal{C}_{i-1}$.

Now note that if ν_i is the number of cubes in $\mathcal{C}_i$, then the number $\mu_i = \nu_i/\nu_{i-1}$ is an estimate for the volume ratio $V(K_i)/V(K_{i-1})$. It is this number μ_i that can now be "randomly approximated" using the approximation constructed above of a uniform sampling over $\mathcal{C}_i$. In fact, a cube C that is reached after sufficiently many steps in the random walk will lie in $\mathcal{C}_{i-1}$ with probability approximately $1/\mu_i$; hence this probability can be approximated closely by repeated sampling.

This algorithm has been improved by various authors. Recently, [KLS] achieved a bound where n enters only to the fifth power.

FURTHER READING

[Bol78] V.G. Boltyanskii. *Hilbert's Third Problem* (Transl. by R. Silverman). Winston, Washington, 1978.

[GW89] R.J. Gardner and S. Wagon. At long last, the circle has been squared. *Notices Amer. Math. Soc.*, 36:1338–1343, 1989.

[GK94b] P. Gritzmann and V. Klee. On the complexity of some basic problems in computational convexity: II. Volume and mixed volumes. In T. Bisztriczky, P. McMullen, R. Schneider, and A. Ivić Weiss, editors, *Polytopes: Abstract, Convex and Computational*, volume 440 of NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci., pages 373–466. Kluwer, Dordrecht, 1994.

[Had57] H. Hadwiger. *Vorlesungen über Inhalt, Oberfläche und Isoperimetrie.* Springer-Verlag, Berlin, 1957.

[McM93] P. McMullen. Valuations and dissections. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, Volume B, pages 933–988. North-Holland. Amsterdam, 1993.

[MS83] P. McMullen and R. Schneider. Valuations on convex bodies. In P.M. Gruber and J.M. Wills, editors, *Convexity and its Applications*, pages 170–247. Birkhäuser, Basel, 1983.

[Sah79] C.-H. Sah. *Hilbert's Third Problem: Scissors Congruence.* Pitman, San Francisco, 1979.

[Wag85] S. Wagon. *The Banach-Tarski Paradox.* Cambridge University Press, 1985.

RELATED CHAPTERS

Chapter 7: Lattice points and lattice polytopes
Chapter 13: Basic properties of convex polytopes
Chapter 14: Subdivisions and triangulations of polytopes
Chapter 34: Randomized algorithms

REFERENCES

[AF92] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements of polyhedra. *Discrete Comput. Geom.*, 8:295–313, 1992.

[BF87] I. Bárány and Z. Füredi. Computing the volume is difficult. *Discrete Comput. Geom.*, 2:319–326, 1987.

[Bar93] A. Barvinok. Computing the volume, counting integral points, and exponential sums. *Discrete Comput. Geom.*, 10:123–141, 1993.

[BN83] H. Bieri and W. Nef. A sweep-plane algorithm for computing the volume of polyhedra represented in boolean form. *Linear Algebra Appl.*, 52/53:69–97, 1983.

[BW92] G. Brightwell and P. Winkler. Counting linear extensions. *Order*, 8:225–242, 1992.

[Deh00] M. Dehn. Über raumgleiche Polyeder. *Nachr. Akad. Wiss. Göttingen Math.-Phys. Kl.*, 345–354, 1900.

[DFK89] M.E. Dyer, A.M. Frieze, and R. Kannan. A random polynomial time algorithm for approximating the volumes of convex bodies. *J. Assoc. Comput. Mach.*, 38:1–17, 1989.

[Hil00] D. Hilbert. Mathematische Probleme. *Nachr. Königl. Ges. Wiss. Göttingen Math.-Phys. Kl.*, 253–297, 1900; *Bull. Amer. Math. Soc.*, 8:437–479, 1902.

[KLS] R. Kannan, L. Lovász, and M. Simonovits. Random walks and an $O^*(n^5)$ volume algorithm for convex bodies. *Random Structures Algorithms,* to appear.

[Kep15] J. Kepler. *Nova Stereometria doliorum vinariorum.* 1615. See M. Caspar, editor, *Johannes Kepler Gesammelte Werke*, Beck, München, 1940.

[Lac90] M. Laczkovich. Equidecomposability and discrepancy: a solution of Tarski's circle-squaring problem. *J. Reine Angew. Math.*, 404:77–117, 1990.

[Law91] J. Lawrence. Polytope volume computation. *Math. Comp.*, 57:259–271, 1991.

[Sta86] R. Stanley. Two order polytopes. *Discrete Comput. Geom.*, 1:9–23, 1986.

27.4 MIXED VOLUMES

The study of mixed volumes, the *Brunn-Minkowski theory*, forms the backbone of classical convexity theory. It is also useful for applications in other areas, including combinatorics and algebraic geometry. A relationship to solving systems of polynomial equations is described at the end of this section.

GLOSSARY

Mixed volume: Let $K_1, \ldots, K_s$ be convex bodies in $\mathbb{R}^n$, and let $\xi_1, \ldots, \xi_s$ be nonnegative reals. Then the function $V\left(\sum_{i=1}^{s} \xi_i K_i\right)$ is a homogeneous polynomial of degree n in the variables $\xi_1, \ldots, \xi_s$, and can be written in the form

$$V\left(\sum_{i=1}^{s} \xi_i K_i\right) = \sum_{i_1=1}^{s} \sum_{i_2=1}^{s} \cdots \sum_{i_n=1}^{s} \xi_{i_1} \xi_{i_2} \cdots \xi_{i_n} V(K_{i_1}, K_{i_2}, \ldots, K_{i_n}),$$

where the coefficients $V(K_{i_1}, K_{i_2}, \ldots, K_{i_n})$ are invariant under permutations of their argument. The coefficient $V(K_{i_1}, K_{i_2}, \ldots, K_{i_n})$ is called the mixed volume of the convex bodies $K_{i_1}, K_{i_2}, \ldots, K_{i_n}$.

27.4.1 MAIN RESULTS

Mixed volumes are nonnegative, monotone, multilinear, and continuous valuations. They generalize the ordinary volume in that $V(K) = V(\overbrace{K, \ldots, K}^{n})$. If A is an affine transformation, then $V(A(K_1), \ldots, A(K_n)) = |\det(A)| V(K_1, \ldots, K_n)$.

Among the most famous inequalities in convexity theory are the ***Aleksandrov-Fenchel inequality***,

$$V(K_1, K_2, K_3, \ldots, K_n)^2 \geq V(K_1, K_1, K_3, \ldots, K_n)\, V(K_2, K_2, K_3, \ldots, K_n),$$

and its consequence, the ***Brunn-Minkowski theorem***, which asserts that for each $\lambda \in [0,1]$,

$$V^{\frac{1}{n}}((1-\lambda)K_0 + \lambda K_1) \geq (1-\lambda)V^{\frac{1}{n}}(K_0) + \lambda V^{\frac{1}{n}}(K_1).$$

OPEN PROBLEM 27.4.1

Provide a useful geometric characterization of the sequences $(K_1, \ldots, K_n)$ for which equality holds in the Aleksandrov-Fenchel inequality.

27.4.2 TRACTABILITY RESULTS

When n is fixed, there is a polynomial-time algorithm whereby, given s ($\mathcal{V}$- or $\mathcal{H}$-) polytopes $P_1, \ldots, P_s$ in $\mathbb{R}^n$, all the mixed volumes $V(P_{i_1}, \ldots, P_{i_n})$ can be computed.

When the dimension is part of the input, it follows at least that mixed volume computation is not harder than volume computation. In fact, computation (for $\mathcal{V}$-polytopes or $\mathcal{S}$-zonotopes) or approximation (for $\mathcal{H}$-polytopes) of any single mixed volume is #P-easy.

27.4.3 INTRACTABILITY RESULTS

Since mixed volumes generalize the ordinary volume, it is clear that mixed volume computation cannot be easier, in general, than volume computation. In addition, there are hardness results for mixed volumes that do not trivially depend on the hardness of volume computations. One such result is described next.

As the term is used here, a *box* is a rectangular parallelotope with axis-aligned edges. Since the vector sum of boxes $V(Z_1, \ldots, Z_n)$ is again a box, the volume of the sum is easy to compute. Nevertheless, computation of the mixed volume $V(Z_1, \ldots, Z_n)$ is hard. This is in interesting contrast to the fact that the volume of a sum of segments (a zonotope) is hard to compute even though each of the mixed volumes can be computed in polynomial time.

27.4.4 RANDOMIZED ALGORITHMS

Since the mixed volumes of convex bodies $K_1, \ldots, K_s$ are coefficients of the polynomial $\varphi(\xi_1, \ldots, \xi_s) = V(\sum_{i=1}^{s} \xi_i K_i)$, it seems natural to estimate these coefficients by combining an interpolation method with a randomized volume algorithm. However, there are significant obstacles to this approach, even for the case of two bodies. First, for a general polynomial φ there is *no* way of obtaining *relative* estimates of its coefficients from *relative* estimates of the values of φ. This can be overcome in the case of two bodies by using the special structure of the polynomial $p(x) = V(K_1 + xK_2)$. However, even then the absolute values of the entries of the "inversion" that is used to express the coefficients of the polynomial in terms of its approximate values are not bounded by a polynomial, while the randomized volume approximation algorithm is polynomial only in $\frac{1}{\tau}$ but not in $size(\tau)$.

Suppose that $\psi : \mathbb{N} \to \mathbb{N}$ is nondecreasing with

$$\psi(n) \leq n \qquad \text{and} \qquad \psi(n) \log \psi(n) = o(\log n).$$

Then there is a polynomial-time algorithm for the problem whose instance consists

of $n, s \in \mathbb{N}$, $m_1, \ldots, m_s \in \mathbb{N}$ with $m_1 + m_2 + \ldots + m_s = n$ and $m_1 \geq n - \psi(n)$, of well-presented convex bodies $K_1, \ldots, K_s$ of $\mathbb{R}^n$, and of positive rational numbers ϵ and β, and whose output is a random variable $\hat{V}_{m_1,\ldots,m_s} \in \mathbb{Q}$ such that

$$\text{prob}\left\{\frac{|\hat{V}_{m_1,\ldots,m_s} - V_{m_1,\ldots,m_s}|}{V_{m_1,\ldots,m_s}} \geq \epsilon\right\} \leq \beta,$$

where

$$V_{m_1,\ldots,m_s} = V(\overbrace{K_1,\ldots,K_1}^{m_1},\ldots,\overbrace{K_s,\ldots,K_s}^{m_s}).$$

Note that the hypotheses above require that m_1 is close to n, and hence that the remaining m_i's are relatively small. A special feature of an interpolation method as used for the proof of this result is that in order to compute *a specific* coefficient of the polynomial under consideration, it computes essentially *all previous* coefficients. Since there can be a polynomial-time algorithm for computing *all such* mixed volumes only if $\psi(n) \leq \log n$, the above result is essentially best-possible for any interpolation method.

In terms of approximation, [Bar] shows that a mixed volume of n proper convex bodies can be approximated by a randomized polynomial-time algorithm within a factor of $n^{O(n)}$.

OPEN PROBLEM 27.4.2 [DGH]

Is there a polynomial-time randomized algorithm that, for arbitrary given $n, s \in \mathbb{N}$, $m_1, \ldots, m_s \in \mathbb{N}$ with $m_1 + m_2 + \ldots + m_s = n$, well-presented convex bodies $K_1, \ldots, K_s$ in $\mathbb{R}^n$, and positive rationals ϵ and β, computes a random variable $\hat{V}_{m_1,\ldots,m_s} \in \mathbb{Q}$ such that $\text{prob}\{|\hat{V}_{m_1,\ldots,m_s} - V_{m_1,\ldots,m_s}|/V_{m_1,\ldots,m_s} \geq \epsilon\} \leq \beta$?

Even the case $s = n$, $m_1 = \ldots = m_s = 1$ is open in general. See, however, [Bar] for some partial results.

AN APPLICATION

Let $S_1, S_2, \ldots, S_n$ be subsets of $\mathbb{Z}^n$, and consider a system $F = (f_1, \ldots, f_n)$ of Laurent polynomials in n variables, such that the exponents of the monomials in f_i are in S_i for all $i = 1, \ldots, n$. For $i = 1, \ldots, n$, let

$$f_i(x) = \sum_{q \in S_i} c_q^{(i)} x^q,$$

where $f_i \in \mathbb{C}[x_1, x_1^{-1}, \ldots, x_n, x_n^{-1}]$, and x^q is an abbreviation for the monomial $x_1^{q_1} \cdots x_n^{q_n}$; $x = (x_1, \ldots, x_n)$ is the vector of indeterminates and $q = (q_1, \ldots, q_n)$ the vector of exponents. Further, let $\mathbb{C}^* = \mathbb{C} \setminus \{0\}$.

Now, if the coefficients $c_q^{(i)}$ $(q \in S_i)$ are chosen "generically," then the number $L(F)$ of distinct common roots of the system F in $(\mathbb{C}^*)^n$ depends only on the ***Newton polytopes*** $P_i = \text{conv}S_i$ of the polynomials. More precisely,

$$L(F) = n! \cdot V(P_1, P_2, \ldots, P_n).$$

In general, $L(F) \leq n! \cdot V(P_1, P_2, \ldots, P_n)$. These connections can be utilized to develop a numerical continuation method for computing the isolated solutions of sparse polynomial systems. For this, see [Emi94, HS95, Roj94, Ver96].

FURTHER READING

[BZ88] Y.D. Burago and V.A. Zalgaller. *Geometric Inequalities.* Springer-Verlag, Berlin, 1988.

[GK94b] P. Gritzmann and V. Klee. On the complexity of some basic problems in computational convexity: II. Volume and mixed volumes. In T. Bisztriczky, P. McMullen, R. Schneider, and A. Ivić Weiss, editors, *Polytopes: Abstract, Convex and Computational*, volume 440 of NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci., pages 373–466. Kluwer, Dordrecht, 1994.

[San93] J.R. Sangwine-Yager. Mixed volumes. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, Volume A, pages 43–72. North-Holland, Amsterdam, 1993.

[Sch93] R. Schneider. *Convex Bodies: The Brunn-Minkowski Theory.* Volume 44 of *Encyclopedia Math. Appl.*, Cambridge University Press, 1993.

RELATED CHAPTERS

Chapter 13: Basic properties of convex polytopes
Chapter 34: Randomized algorithms

REFERENCES

[Bar] A. Barvinok. Computing mixed discriminants, mixed volumes and permanents. *Discrete Comput. Geom.*, to appear.

[DGH] M.E. Dyer, P. Gritzmann, and A. Hufnagel. On the complexity of computing mixed volumes. *SIAM J. Comput.*, to appear.

[Emi94] I.Z. Emiris. *Sparse Elimination and Applications in Kinematics.* Ph.D. Thesis, Univ. of California, Berkeley, 1994.

[HS95] B. Huber and B. Sturmfels. A polyhedral method for solving sparse polynomial systems. *Math. Comput.*, 64:1541–1555, 1995.

[Roj94] J.M. Rojas. *Cohomology, Combinatorics, and Complexity Arising from Solving Polynomial Systems.* Ph.D. Thesis, Univ. of California, Berkeley, 1994.

[Ver96] J. Verschelde. *Homotopy Continuation Methods for Solving Polynomial Systems.* Ph.D. Thesis, Katholieke Universiteit Leuven, 1996.

27.5 CONTAINMENT PROBLEMS

Typically, containment problems involve two fixed sequences, Γ and Ω, that are given as follows: for each $n \in \mathbb{N}$, let $\mathcal{C}_n$ denote a family of closed convex subsets of $\mathbb{R}^n$, and let $\omega_n : \mathcal{C}_n \longrightarrow \mathbb{R}$ be a functional that is nonnegative and is monotone with respect to inclusion. Then $\Gamma = (\mathcal{C}_n)_{n\in\mathbb{N}}$ and $\Omega = (\omega_n)_{n\in\mathbb{N}}$.

GLOSSARY

(Γ, Ω)-INBODY: Accepts as input a positive integer n, a body K in $\mathbb{R}^n$ that is given by an oracle or is an $\mathcal{H}$-polytope, a $\mathcal{V}$-polytope, or an $\mathcal{S}$-zonotope, and a positive rational λ. It answers the question of whether there is a $C \in \mathcal{C}_n$ such that $C \subset K$ and $\omega_n(C) \geq \lambda$.

(Γ, Ω)-CIRCUMBODY is defined similarly for $C \supset K$.

j-simplex S ***bound to a polytope*** P: Each vertex of S is a vertex of P.

Largest j-simplex in a given polytope: One of maximum j-measure.

27.5.1 THE GENERAL CONTAINMENT PROBLEM

The general containment problem deals with the question of computing, approximating, or measuring extremal bodies of a given class that are contained in or contain a given convex body. Since [GK94a] contains a broad survey of containment problems, the present account is confined to some selected examples.

27.5.2 OPTIMAL CONTAINMENT UNDER HOMOTHETY

The results on (Γ, Ω)-INBODY and (Γ, Ω)-CIRCUMBODY are summarized below for the case in which each C_n is a fixed polytope,

$$\mathcal{C}_n = \{g(C_n) \mid g \text{ is a homothety}\},$$

and

$$\omega_n(g(C_n)) = \rho, \quad \text{when } g(C_n) = a + \rho C_n \text{ for some } a \in \mathbb{R}^n \text{ and } \rho \geq 0.$$

As an abbreviation, these specific problems are denoted by $\mathcal{E}^{\text{Hom}}$-INBODY and $\mathcal{E}^{\text{Hom}}$-CIRCUMBODY, respectively, where $\mathcal{E} = (C_n)_{n \in \mathbb{N}}$ and a subscript ($\mathcal{V}$ or $\mathcal{H}$) is used to indicate the manner in which each C_n is presented.

There are polynomial-time algorithms for the following problems:

$\mathcal{E}^{\text{Hom}}_{\mathcal{V}}$-INBODY for $\mathcal{V}$-polytopes P;	$\mathcal{E}^{\text{Hom}}_{\mathcal{V}}$-CIRCUMBODY for $\mathcal{V}$-polytopes P;
$\mathcal{E}^{\text{Hom}}_{\mathcal{V}}$-INBODY for $\mathcal{H}$-polytopes P;	$\mathcal{E}^{\text{Hom}}_{\mathcal{H}}$-CIRCUMBODY for $\mathcal{V}$-polytopes P;
$\mathcal{E}^{\text{Hom}}_{\mathcal{H}}$-INBODY for $\mathcal{H}$-polytopes P;	$\mathcal{E}^{\text{Hom}}_{\mathcal{H}}$-CIRCUMBODY for $\mathcal{H}$-polytopes P.

These positive results are best possible in the sense that the cases not listed above contain instances of NP-hard problems. In fact, the problem $\mathcal{E}^{\text{Hom}}_{\mathcal{H}}$-INBODY is coNP-complete even when C_n is the standard unit $\mathcal{H}$-cube while P is restricted to the class of all affinely regular $\mathcal{V}$-cross-polytopes centered at the origin. The problem $\mathcal{E}^{\text{Hom}}_{\mathcal{V}}$-CIRCUMBODY is coNP-complete even when C_n is the standard $\mathcal{V}$-cross-polytope while P is restricted to the class of all $\mathcal{H}$-parallelotopes centered at the origin.

There are some results for bodies that are more general than polytopes. Suppose that for each $n \in \mathbb{N}$, C_n is a centrally symmetric body in $\mathbb{R}^n$, and that there exists a number μ_n whose size is bounded by a polynomial in n and an n-dimensional $\mathcal{S}$-parallelotope Z that is strictly inscribed in $\mu_n C_n$ (i.e., the intersection of Z with

the boundary of $\mu_n C_n$ consists of the vertex set of Z), the size of the presentation being bounded by a polynomial in n. Then with $\mathcal{E} = (C_n)_{n\in\mathbb{N}}$, (an appropriate variant of) the problem $\mathcal{E}^{\mathrm{Hom}}$-CIRCUMBODY is NP-hard for the classes of all centrally symmetric $(n-1)$-dimensional $\mathcal{H}$-polytopes in $\mathbb{R}^n$. With the aid of polarity, similar results for $\mathcal{E}^{\mathrm{Hom}}$-INBODY can be obtained.

27.5.3 OPTIMAL CONTAINMENT UNDER AFFINITY: SIMPLICES

This section focuses on the problem of finding a largest j-dimensional simplex in a given n-dimensional polytope, where *largest* means of maximum j-measure.

When an n-polytope P has m vertices, it contains at most $\binom{m}{j+1}$ bound j-simplices. There is always a largest j-simplex that is bound, and hence there is a finite algorithm for finding a largest j-simplex contained in P.

Each largest j-simplex in P contains at least two vertices of P. However, there are polytopes P of arbitrarily large dimension, with an arbitrarily large number of vertices, such that some of the largest n-simplices in P have only two vertices in the vertex-set of P. Hence for $j \geq 2$ it is not clear whether there is a finite algorithm for producing a useful presentation of *all* the largest j-simplices in a given n-polytope.

The problem of finding a largest j-simplex in a $\mathcal{V}$- or $\mathcal{H}$-polytope can be solved in polynomial time when the dimension n of the polytope is fixed. Further, for fixed j, the volumes of all j-simplices in a given $\mathcal{V}$-polytope can be computed in polynomial time (even for varying n).

Suppose that the functions $\psi : \mathbb{N} \to \mathbb{N}$ and $\gamma : \mathbb{N} \to \mathbb{N}$ are both of order $\Omega(n^{1/k})$ for some $k \in \mathbb{N}$, and that $1 \leq \gamma(n) \leq n$ for each $n \in \mathbb{N}$. Then the following problem is NP-complete: Given $n, \lambda \in \mathbb{N}$, and the vertex set V of an n-dimensional $\mathcal{V}$-polytope $P \subset \mathbb{R}^n$ with $|V| \leq n + \psi(n)$, and given $j = \gamma(n)$, decide whether P contains a j-simplex S such that $(j!)^2 \mathrm{vol}(S)^2 \geq \lambda$. Note that the conditions for γ are satisfied when $\gamma(n) = \max\{1, n - \mu\}$ for a nonnegative integer constant μ, and also when $\gamma(n) = \max\{1, \lfloor \mu n \rfloor\}$ for a fixed rational μ with $0 < \mu \leq 1$.

A similar hardness result holds for $\mathcal{H}$-polytopes. There the question is the same, but the growth condition on the function γ is that $1 \leq \gamma(n) \leq n$ and that there exists a function $f : \mathbb{N} \to \mathbb{N}$, bounded by a polynomial in n, such that for each $n \in \mathbb{N}$, $f(n) - \gamma(f(n)) = n$. Note that such an f exists when the function γ is constant, and also when $\gamma(n) = \lfloor \mu n \rfloor$ for fixed rational μ with $0 < \mu < 1$.

The "dual" problem of finding smallest simplices containing a given polytope P seems even harder, since the relationship between a smallest such simplex and the faces of P is much weaker.

CONJECTURE 27.5.1 [GKL95]

For each function $\gamma : \mathbb{N} \to \mathbb{N}$ with $1 \leq \gamma(n) \leq n$, the problem of finding a largest j-simplex in a given n-dimensional $\mathcal{H}$-polytope P is NP*-hard, even for the case in which P is a parallelotope.*

APPLICATIONS

The paper [HKL96] is in part a survey of the problem of finding largest j-simplices in an n-dimensional cube. As outlined in [GK94a], applications of this problem and its relatives include the Hadamard determinant problem, finding optimal weighing

designs, and bounding the growth of pivots in Gaussian elimination with complete pivoting.

27.5.4 OPTIMAL CONTAINMENT UNDER AFFINITY: ELLIPSOIDS

For an arbitrary proper body K in $\mathbb{R}^n$, there is a unique ellipsoid E_0 of maximum volume contained in K, and it is concentric with the unique ellipsoid E of minimum volume containing K. If a is the common center, then $K \subset a + n(E_0 - a)$, where the factor n can be replaced by $\sqrt{n}$ when K is centrally symmetric. E is called the ***Löwner-John ellipsoid*** of K, and it plays an important role in the algorithmic theory of convex bodies.

Algorithmic approximations of the Löwner-John ellipsoid can be obtained by use of the ellipsoid method [GLS88]: There exists an oracle-polynomial-time algorithm that, for any well-bounded body K of $\mathbb{R}^n$ given by a weak separation oracle, finds a point a and a linear transformation A such that

$$a + A(B^n) \subset K \subset a + (n+1)\sqrt{n}A(B^n).$$

Further, the dilatation factor $(n+1)\sqrt{n}$ can be replaced by $\sqrt{n(n+1)}$ when K is symmetric, by $(n+1)$ when K is an $\mathcal{H}$-polytope, and by $\sqrt{n+1}$ when K is a symmetric $\mathcal{H}$-polytope.

[TKE88] and [KT93] give polynomial-time algorithms for approximating the ellipsoid of maximum volume E_0 that is contained in a given $\mathcal{H}$-polytope. For each rational $\gamma < 1$, there exists a polynomial-time algorithm that, given $n, m \in \mathbb{N}$ and $a_1, \ldots, a_m \in \mathbb{Q}^n$, computes an ellipsoid $E = a + A(B^n)$ such that

$$E \subset P = \{x \in \mathbb{R}^n \mid \langle a_i x \rangle \leq 1, \text{ for } i = 1, \ldots, m\} \quad \text{and} \quad \frac{V(E)}{V(E_0)} \geq \gamma.$$

The running time of the algorithm is

$$O\left(m^{3.5} \log\big(mR/(r\log(1/\gamma))\big) \log\big(nR/(r\log(1/\gamma))\big)\right),$$

where the numbers r and R are, respectively, a lower bound on the inradius of P and an upper bound on its circumradius.

It is not known whether a similar result holds for $\mathcal{V}$-polytopes.

As shown in [TKE88], an approximation of E_0 of the kind given above leads to the following inclusion:

$$a + A(B^n) \subset K \subset a + \frac{n(1+3\sqrt{1-\gamma})}{\gamma} A(B^n).$$

FURTHER READING

[GK94a] P. Gritzmann and V. Klee. On the complexity of some basic problems in computational convexity: I. Containment problems. *Discrete Math*, 136:129–174, 1994. Reprinted in W. Deuber, H.-J. Prömel, and B. Voigt, editors, *Trends in Discrete Mathematics*, pages 129–174. *Topics in Discrete Math.*, North-Holland, Amsterdam, 1994.

[GLS88] M. Grötschel, L. Lovász, and A. Schriver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, Berlin, 1988, 1993.

RELATED CHAPTERS

Chapter 39: Mathematical programming

REFERENCES

[GKL95] P. Gritzmann, V. Klee, and D.G. Larman. Largest j-simplices in n-polytopes. *Discrete Comput. Geom.*, 13:477–515, 1995.

[HKL96] M. Hudelson, V. Klee, and D. Larman. Largest j-simplices in d-cubes: Some relatives of the Hadamard maximum determinant problem. *Linear Algebra Appl.*, 241–243:519–598, 1996.

[KT93] L. Khachiyan and M. Todd. On the complexity of approximating the maximal inscribed ellipsoid in a polytope. *Math. Programming*, 61:137–160, 1993.

[TKE88] S.P. Tarasov, L.G. Khachiyan, and I.I. Erlich. The method of inscribed ellipsoids. *Soviet Math. Dokl.*, 37:226–230, 1988.

27.6 RADII

The diameter, width, circumradius, and inradius of a convex body are classical functionals that play an important role in convexity theory and in many applications. For other applications, generalizations have been introduced. The underlying space is a ***Minkowski space*** (finite-dimensional normed space) $\mathbb{M} = (\mathbb{R}^n, \| \quad \|)$. Let B denote its unit ball, j a positive integer, and K a convex body.

GLOSSARY

Outer j-radius $R_j(K)$ of K: Infimum of the positive numbers ρ such that the space contains an $(n-j)$-flat F for which $K \subset F + \rho B$.

j-ball of radius ρ: Set of the form $(q + \rho B) \cap F = \{x \in F \mid \|x - q\| \leq \rho\}$ for some j-flat F in $\mathbb{R}^n$ and point $q \in F$.

Inner j-radius $r_j(K)$ of K: Maximum of the radii of the j-balls contained in K.

Diameter of K: $2r_1(K)$.

Width of K: $2R_1(K)$.

Inradius of K: $r_n(K)$.

Circumradius of K: $R_n(K)$.

For the case of variable dimension (i.e., the dimension is part of the input), Tables 27.6.1, 27.6.2, and 27.6.3 provide a rapid indication of the main complexity results for the most important radii: r_1, R_1, r_n, and R_n; and for the three most important ℓ_p spaces: $\mathbb{R}^n_2$, $\mathbb{R}^n_1$, and $\mathbb{R}^n_\infty$. The designations P, NPC, and NPH indicate respectively polynomial-time computability, NP-completeness, and NP-hardness. The tables provide only a rough indication of results. They are imprecise in the

following respects: (i) the diameter and width are actually equal to $2r_1$ and $2R_1$ respectively; (ii) the results for $\mathbb{R}_2^n$ involve the square of the radius rather than the radius itself; (iii) some of the P entries are based on polynomial-time approximability rather than polynomial-time computability; (iv) the designations NPC and NPH do not refer to computability per se, but to the appropriately related decision problems involving the establishment of lower or upper bounds for the radii in question.

TABLE 27.6.1 Complexity of radii in $\mathbb{R}_2^n$.

Polytope functional		$\mathcal{H}$-polytopes general	$\mathcal{H}$-polytopes symmetric	$\mathcal{V}$-polytopes general	$\mathcal{V}$-polytopes symmetric
Diameter	r_1^2	NPC	NPC	P	P
Inradius	r_n^2	P	P	NPH	NPC
Width	R_1^2	NPC	P	NPC	NPC
Circumradius	R_n^2	NPC	NPC	P	P

TABLE 27.6.2 Complexity of radii in $\mathbb{R}_1^n$.

Polytope functional		$\mathcal{H}$-polytopes general	$\mathcal{H}$-polytopes symmetric	$\mathcal{V}$-polytopes general	$\mathcal{V}$-polytopes symmetric
Diameter	r_1	NPC	NPC	P	P
Inradius	r_n	P	P	P	P
Width	R_1	P	P	P	P
Circumradius	R_n	NPC	NPC	P	P

TABLE 27.6.3 Complexity of radii in $\mathbb{R}_\infty^n$.

Polytope functional		$\mathcal{H}$-polytopes general	$\mathcal{H}$-polytopes symmetric	$\mathcal{V}$-polytopes general	$\mathcal{V}$-polytopes symmetric
Diameter	r_1	P	P	P	P
Inradius	r_n	P	P	NPC	NPC
Width	R_1	NPC	P	NPC	NPC
Circumradius	R_n	P	P	P	P

APPLICATIONS

Applications of radii include conditioning in global optimization, sensitivity analysis of linear programs, orthogonal minimax regression, computer graphics and computer vision, chromosome classification, set separation, and design of membranes and sieves; see [GrK93a].

FURTHER READING

[GK94a] P. Gritzmann and V. Klee. On the complexity of some basic problems in computational convexity: I. Containment problems. *Discrete Math*, 136:129–174, 1994. Reprinted in W. Deuber, H.-J. Prömel, and B. Voigt, editors, *Trends in Discrete Mathematics*, pages 129–174. *Topics in Discrete Math.*, North-Holland, Amsterdam, 1994.

REFERENCES

[GK93a] P. Gritzmann and V. Klee. Computational complexity of inner and outer j-radii of polytopes in finite dimensional normed spaces. *Math. Programming*, 59:163–213, 1993.

27.7 INTERVAL MATRICES, QUALITATIVE MATRICES

The mathematical modeling of practical problems often involves real matrices whose entries are not known precisely, but are known only to lie in specified bounded closed intervals or to be of specified sign. The interval case arises in many applications, while the study of the sign case was motivated by questions concerning the modeling of problems in economics. The associated complexity results and problems can be formulated in terms of systems of parallelotopes or systems of sign cones, and there have been some extensions to more general systems of convex sets. Attention is confined here to the two most-studied topics, solvability of linear algebraic systems and stability of linear dynamical systems.

GLOSSARY

***m* × *n* system:** A sequence $\mathcal{A} = (A_1, \ldots, A_n)$ of n nonempty subsets of $\mathbb{R}^m$.

Matrices associated with an m × n system: The set $\mathcal{M}(\mathcal{A})$ of all $m \times n$ matrices $A = [a_1, \ldots, a_n]$ such that $a_j \in A_j$ for each j, where a_j denotes the jth column of A.

***L*-system:** A system $\mathcal{A}$ such that for each $A \in \mathcal{M}(\mathcal{A})$, the columns of A are linearly independent.

***S*-system:** A system $\mathcal{A}$ such that for each $A \in \mathcal{M}(\mathcal{A})$, the n columns of A are the vertices of an $(n-1)$-simplex in $\mathbb{R}^m$ whose relative interior includes the origin. (Equivalently, the nullspace of A is a line in $\mathbb{R}^n$ that passes through the origin and penetrates the positive orthant of $\mathbb{R}^n$.)

Sign cone: A subset of $\mathbb{R}^m$ that, for some sequence of m signs $(-, 0, +)$, consists of all points of $\mathbb{R}^m$ that exhibit the specified sign pattern.

Qualitative matrix: An $m \times n$ matrix A in which each entry is one of the intervals $(-\infty, 0)$, $\{0\}$, and $(0, \infty)$. This may be viewed instead as an $m \times n$ system $\mathcal{A} = (A_1, \ldots, A_n)$ in which each A_j is a sign cone.

***L*-matrix, *S*-matrix:** A qualitative matrix that gives rise to an L-system or an S-system, respectively.

Interval matrix: An $m \times n$ matrix $A = ([\alpha_{ij}, \beta_{ij}])$ in which each entry is a bounded closed interval in $\mathbb{R}$. This may be viewed instead as an $m \times n$ system $\mathcal{A} = (A_1, \ldots, A_n)$ in which each A_j is the parallelotope $[\alpha_{1j}, \beta_{1j}] \times \ldots \times [\alpha_{mj}, \beta_{mj}]$.

Nonsingularity of a system: An $n \times n$ system $\mathcal{A}$ is nonsingular if every member of $\mathcal{M}(\mathcal{A})$ has this property.

Sign-nonsingular: When $\mathcal{A}$ is a system of sign cones, the preceding notion is called *sign-nonsingularity.* In other words, a sign-nonsingular matrix is a square matrix whose sign pattern guarantees nonsingularity.

Matrix stability: A square real matrix A is ***semistable*** (resp. ***stable***) if each of its eigenvalues has nonnegative (resp. positive) real part. It is ***quasistable*** if it is semistable and, in addition, each eigenvalue with zero real part is a simple root of the minimum polynomial of A. These terms are used for an $n \times n$ system $\mathcal{A}$ when they apply to *every* $A \in \mathcal{M}(\mathcal{A})$.

Matrix sign-stability: When $\mathcal{A}$ is a system of sign cones, the preceding notion is called *sign-stability.* In other words, a ***sign-stable*** matrix is a square matrix whose sign pattern guarantees stability. ***Sign-semistability*** and ***sign-quasistability*** are defined similarly.

Sign-solvability: A system of linear equations, $Ax = b$, is sign-solvable if both the solvability of the system and the sign pattern of the solution x are implied by the sign patterns of A and b.

BASIC FACTS

The problem of testing a square matrix for sign-nonsingularity is polynomially equivalent to the problem of testing a digraph for the presence of a (simple) cycle that has an even number of edges. If either recognition problem admits a polynomial-time algorithm, then so does the other.

The study of sign-solvability can in a sense be decomposed into the study of L-matrices and the study of S-matrices—equivalently, into the study of L-systems and S-systems of sign cones. This result can be extended to more general $m \times n$ systems $\mathcal{A} = (A_1, \ldots, A_n)$ under the assumption that each A_j has nonempty interior relative to the smallest canonical subspace of $\mathbb{R}^m$ that contains it.

For an $n \times n$ real matrix A, consider the system $x' = Ax$ of linear differential equations with constant coefficients. For each point $p_0 \in \mathbb{R}^n$, there is a unique *positive trajectory* $x : [0, \infty) \longrightarrow \mathbb{R}^n$ of this system that has $x(0) = p_0$. The matrix A is stable if and only if each positive trajectory converges to the origin, is quasistable if and only if each positive trajectory is bounded, and is semistable if and only if no positive trajectory runs off to infinity at an exponential rate.

TRACTABILITY RESULTS

There is an $O(n^2)$ algorithm for deciding whether a given $n \times (n+1)$ matrix is an S-matrix. In the general case of systems of polyhedral cones presented in terms of their generators, an algorithm based on linear programming can recognize S-systems in polynomial time [KVL93].

For a properly presented square matrix A, sign-stability, sign-quasistability, and sign-semistability can all be detected in time that is proportional to the number of nonzero entries of A [JKV87].

INTRACTABILITY RESULTS

Deciding whether a given rectangular sign matrix is an L-matrix is NP-hard, and this is true even when the matrix is "almost square" in a certain sense [KLM84].

Testing the nonsingularity of a symmetric square interval matrix is NP-hard, as is testing the stability of such a matrix [Roh94].

OPEN PROBLEM 27.7.1

It is unknown whether there is a polynomial-time algorithm for recognizing sign-nonsingular matrices, or, equivalently, detecting the presence of an even cycle in a digraph.

The problem is close to the "boundary" of NP-completeness, in the sense that among closely related problems there are several that can be solved in polynomial time and several others that are known to be NP-hard. In particular, the following problems are solvable in polynomial time: detecting the presence of an even (or odd) cycle in an undirected graph; detecting the presence of an odd cycle in a digraph; and detecting the presence of an even cycle in a planar digraph [Tho93], and in fact in any digraph that can be embedded in a fixed but arbitrary surface [GL96]. On the other hand, the problem of detecting the presence of an even (or odd) cycle through a specified vertex of a digraph is NP-hard [KLM84, LP84].

FURTHER READING

[BS95] R.A. Brualdi and B.L. Shader. *Matrices of Sign-Solvable Linear Systems.* Cambridge University Press, 1995.

REFERENCES

[GL96] A. Galluccio and M. Loebl. Even cycles and H-homeomorphisms. Preprint, Dept. of Combinatorics and Optimization, Univ. of Waterloo, 1996.

[JKV87] C. Jeffries, V. Klee, and P. Van Den Driessche. Qualitative stability of linear systems. *Linear Algebra Appl.*, 87:1–48, 1987.

[KLM84] V. Klee, R. Ladner, and R. Manber. Sign-solvability revisited. *Linear Algebra Appl.*, 59:131–157, 1984.

[KVL93] V. Klee, B. Von Hohenbalken, and T. Lewis. On the recognition of S-systems. *Linear Algebra Appl.*, 192:187–204, 1993.

[LP84] A.S. LaPaugh and C.H. Papadimitriou. The even path problem for graphs and digraphs. *Networks*, 14:597–514, 1984.

[Roh94] J. Rohn. Checking positive definiteness or stability of symmetric interval matrices is NP-hard. *Comment. Math. Univ. Carolin.*, 35:795–797, 1994.

[Tho93] C. Thomassen. The even cycle problem for planar digraphs. *J. Algorithms*, 15:61–75, 1993.

28 COMPUTATIONAL TOPOLOGY

Gert Vegter

INTRODUCTION

Topology studies point sets and their invariants under continuous deformations, invariants such as the number of connected components, holes, tunnels, or cavities. Metric properties such as the position of a point, the distance between points, or the curvature of a surface, are irrelevant to topology. A high level description of the main concepts and problems in topology is given in Section 28.1. Computational topology deals with the complexity of such problems, and with the design of efficient algorithms for their solution, in case these problems are tractable. These algorithms can deal only with spaces and maps that have a finite representation. To this end we consider simplicial complexes and maps (Section 28.2) and CW-complexes (Section 28.3). Section 28.4 deals with algebraic invariants of topological spaces, which are in general easier to compute than topological invariants. Mapping (embedding) a topological space 1–1 into another space may reveal some of its topological properties. Several types of embeddings are considered in Section 28.5. Section 28.6 deals with the classification of immersions of a space into another space. These maps are only *locally* 1–1, and hence more general than embeddings.

Many computational problems in topology are undecidable (in the sense of complexity theory). The mathematical literature of this century contains many (beautiful) topological algorithms, usually reducing to decision procedures, in many cases with exponential-time complexity. The quest for efficient algorithms for topological problems has started rather recently. Most of the problems in computational topology still await an efficient solution.

28.1 TOPOLOGICAL SPACES AND MAPS

Topology deals with the classification of spaces that are the same up to some equivalence relation. We introduce these notions, and describe some classes of topological problems.

GLOSSARY

Space: In this chapter a ***topological space*** (or ***space***, for short) is a subset of some Euclidean space $\mathbb{R}^d$, endowed with the topology of $\mathbb{R}^d$.

Map: A function $f : X \to Y$ from a space X to a space Y is a map if f is continuous.

Homeomorphism: A 1–1 map $h : X \to Y$, with a continuous inverse, is called a homeomorphism from X to Y (or: between X and Y).

Topological equivalence: Two spaces are topologically equivalent (or ***homeomorphic***) if there is a homeomorphism between them.

Embedding: A map $f : X \to Y$ is an embedding if f is a homeomorphism onto its image. We say that X can be (topologically) embedded in Y.

Homotopy of maps: Two maps $f_0, f_1 : X \to Y$ are homotopic if there is a map $F : X \times [0,1] \to Y$ such that $F(x,0) = f_0(x)$ and $F(x,1) = f_1(x)$, for all $x \in X$.

Homotopy equivalence: Two spaces X and Y are homotopy equivalent if there are maps $f : X \to Y$ and $g : Y \to X$ such that gf and fg are homotopic to the identity mappings on X and Y, respectively. Obviously topological equivalence implies homotopy equivalence.

Topological/homotopy invariant: A map ζ associating a number, or a group, $\zeta(X)$ to a space X, is a topological invariant (resp. homotopy invariant) if $\zeta(X_1)$ and $\zeta(X_2)$ are equal, or isomorphic, for topologically equivalent (resp. homotopy equivalent) spaces X_1 and X_2.

Contractability: A space is contractible if it is homotopy equivalent to a point.

Unit interval $\mathbb{I}$***:*** The interval $[0,1]$ in $\mathbb{R}$.

Ball: Open d-ball: $\mathbb{B}^d = \{(x_1, \cdots, x_d) \in \mathbb{R}^d \mid x_1^2 + \cdots x_d^2 < 1\}$. Closed d-ball: $\overline{\mathbb{B}}^d$ is the closure of $\mathbb{B}^d$.

Half ball: $\mathbb{B}^d_+ = \{(x_1, \cdots, x_d) \in \mathbb{R}^d \mid x_1^2 + \cdots x_d^2 < 1 \text{ and } x_d \geq 0\}$.

Sphere: $\mathbb{S}^d = \{(x_1, \cdots, x_{d+1}) \in \mathbb{R}^{d+1} \mid x_1^2 + \cdots x_{d+1}^2 = 1\}$ is the d-sphere. It is the boundary of the $(d+1)$-ball.

Manifold: A space X is a d-dimensional (topological) manifold (also: d-manifold) if every point of X has a neighborhood homeomorphic to $\mathbb{B}^d$. X is a d-manifold ***with boundary*** if every point has a neighborhood homeomorphic to $\mathbb{B}^d$ or $\mathbb{B}^d_+$.

Surface: A 2-dimensional manifold, with or without boundary. A ***closed surface*** is a surface without boundary.

Curve: A curve in X is a continuous map $\mathbb{I} \to X$. For $x_0 \in X$, a x_0-based ***closed curve*** c is a curve for which $c(0) = c(1) = x_0$.

BASIC TOPOLOGICAL PROBLEMS AND APPLICATIONS

Topological equivalence and classification: Decide whether a space belongs to (is topologically equivalent to an element of) a class of known objects.
Application: Object recognition in computer vision.

Homotopy equivalence: Decide whether two spaces are homotopy equivalent, or whether a curve in X is contractible (the ***contractibility problem***).
Applications: α-hull, skeletons; see [Ede94]. Concurrent computing; see [HS94a].

Embedding: Decide whether X can be embedded in Y. If so, construct an embedding.
Application: Graph drawing (Chapter 44), VLSI-layout, and wire routing.

Extension of maps: Let A be a subspace of X. Decide whether a map $f : A \to Y$ can be extended to X (i.e., whether there is a map $F : X \to Y$ whose restriction to A is f).

Lifting of maps: Let $f : A \to X$ and $p : Y \to X$ be maps. Decide whether there is a map $F : A \to Y$ such that $pF = f$.
Application: Inverse kinematics problems and tracking algorithms in robotics; see [Bak90] and Section 41.1.

28.2 SIMPLICIAL COMPLEXES

Computation requires finite representation of topological spaces. Representing a space by a simplicial complex corresponds to the idea of building the space from simplices. Simplicial complexes may be considered as combinatorial objects, with a straightforward data structure for their representation. See also Section 15.1.

GLOSSARY

Geometric simplex: A geometric k-simplex σ_k is the convex hull of a set A of $k+1$ independent points $a_0, \cdots, a_k$ in some Euclidean space $\mathbb{R}^d$ (so $d \geq k$). A is said to ***span*** the simplex σ_k. A simplex spanned by a subset A' of A is called a ***face*** of σ_k. The face is proper if $\emptyset \neq A' \neq A$. The ***dimension*** of the face is $|A'| - 1$. A 0-dimensional face is called a ***vertex***, a 1-dimensional face is called an ***edge***. The union σ_k^i, $0 \leq i \leq k$, of all faces of dimension at most i is called the ***i-skeleton*** of σ_k. In particular σ_k^0 is the set of vertices, and $\sigma_k^k = \sigma_k$. An ***orientation*** of σ_k is induced by an ordering of its vertices, denoted by $\langle a_0, \cdots, a_k \rangle$, as follows: For any permutation π of $0, \cdots, k$, the orientation $\langle a_{\pi(0)}, \cdots, a_{\pi(k)} \rangle$ is equal to $(-1)^{sign(\pi)} \langle a_0, \cdots, a_k \rangle$, where $sign(\pi)$ is the number of transpositions of π (so any simplex has two distinct orientations). If τ is a $(k-1)$-dimensional face of σ, obtained by omitting the vertex a_i, then the ***induced orientation*** on τ is $(-1)^i \langle a_0, \cdots, \hat{a}_i, \cdots, a_k \rangle$, where the hat indicates omission of a_i.

Geometric simplicial complex K: A finite set of simplices in some Euclidean space $\mathbb{R}^m$, such that (i) if σ is a simplex of K and τ is a face of σ, then τ is a simplex of K, and (ii) if σ and τ are simplices of K, then $\sigma \cap \tau$ is either empty or a common face of σ and τ. The ***dimension*** of K is the maximum of the dimensions of its simplices. The ***underlying space*** of K, denoted by $|K|$, is the union of all simplices of K, endowed with the subspace topology of $\mathbb{R}^m$. The ***i-skeleton*** of K, denoted by K^i, is the union of all simplices of K of dimension at most i. A ***subcomplex*** L of K is a subset of K that is a simplicial complex.

Combinatorial simplicial complex: A pair $\mathcal{K} = (V, \Sigma)$, where V contains finitely many elements, called vertices, and Σ is a collection of subsets of V, called (combinatorial) simplices, with the property that any subset of a simplex is a simplex. The dimension of a simplex is one less than the number of vertices it contains. The dimension of $\mathcal{K}$ is the maximum of the dimensions of its simplices.

Geometric realization: A geometric simplicial complex K in $\mathbb{R}^m$ is called a ***geometric realization*** (in $\mathbb{R}^m$) of the combinatorial simplicial complex $\mathcal{K} = (V, \Sigma)$ if there is a 1–1 correspondence $f : V \to K^0$, such that $A \subset V$ is a simplex of $\mathcal{K}$ iff $f(A)$ spans a simplex of K. Furthermore $\mathcal{K}$ is called the ***abstraction*** of K.

Triangulation: A triangulation of a topological space X is a pair (K, h), where K is a geometric simplicial complex and h is a homeomorphism from the underlying space $|K|$ to X.

Barycentric subdivision: The ***barycenter*** (center of mass) of a geometric k-simplex with vertices $a_0, \cdots, a_k$ in $\mathbb{R}^m$ is the point $1/(k+1)\sum_{i=0}^{k} a_i$. The barycentric subdivision of a geometric simplicial complex K is defined inductively: (i) the barycentric subdivision of the 0-skeleton σ^0 is σ^0 itself; (ii) if σ is an i dimensional face of K, $i > 0$, then σ is subdivided into the collection of simplices $C(b, \tau)$, for all simplices τ in the barycentric subdivision of the $(i-1)$-skeleton of σ. Here $C(b, \tau)$ is the convex hull of $b \cup \tau$ and b the barycenter of σ.

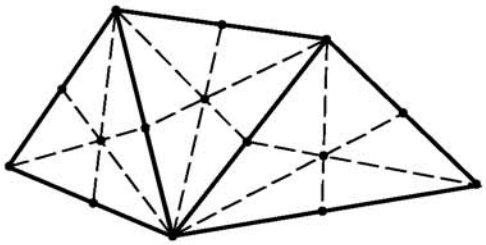


FIGURE 28.2.1
Barycentric subdivision.

The (first) barycentric subdivision of a simplicial complex K is the simplicial complex $s(K)$ obtained by barycentric subdivision of all simplexes of K; see Figure 28.2.1. The ith barycentric subdivision of K, $i > 1$, is defined inductively as $s(s^{i-1}(K))$. A simplicial complex L is called a ***refinement*** of K if $L = s^i(K)$, for some $i \geq 0$.

Simplicial map: A simplicial map between simplicial complexes K and L is a function $f : |K| \to |L|$ such that (i) if a is a vertex of K then $f(a)$ is a vertex of L; (ii) if $a_0, \cdots, a_k$ are vertices of a simplex of K, then the convex hull of $f(a_0), \cdots, f(a_k)$ is a simplex of L (whose dimension may be less than k); and (iii) f is linear on each simplex: if $x = \sum_{i=0}^{k} \lambda_i a_i$ is a point in a simplex with vertices $a_0, \cdots, a_k$, then $f(x) = \sum_{i=0}^{k} \lambda_i f(a_i)$.

Simplicial equivalence: Two simplicial complexes K and L are simplicially equivalent iff there are simplicial maps $f : |K| \to |L|$ and $g : |L| \to |K|$ such that gf is the identity on $|K|$ and fg is the identity on $|L|$.

Piecewise linear (PL)-equivalence: Two simplicial complexes K and L are called piecewise linearly equivalent (PL-equivalent, for short) if there is a refinement K' of K and L' of L such that K' and L' are simplicially equivalent.

Orientation of a simplicial manifold: An orientation of a simplicial complex K, whose underlying space is a d-manifold, is a choice of orientation for each simplex of K, such that, if τ is a $(d-1)$-face of two distinct d-simplices σ_1 and σ_2, then the orientation on τ induced by σ_1 is the opposite of the orientation induced by σ_2. The manifold is called ***orientable*** if it has a triangulation that has an orientation, otherwise it is ***nonorientable***.

Euler characteristic: (Combinatorial definition; cf. Section 28.4) The Euler characteristic of a simplicial d-complex K, denoted by $\chi(K)$, is the number $\sum_{i=0}^{d}(-1)^i \alpha_i$, where α_i is the number of i-simplices of K.

Polygonal schema for a surface: Let $\mathcal{M}_g(a_1, b_1, \cdots, a_g, b_g)$ be a regular $4g$-gon, whose successive edges are labeled $a_1, b_1, \overline{a}_1, \overline{b}_1, \cdots, a_g, b_g, \overline{a}_g, \overline{b}_g$. Edge x is directed counterclockwise, edge $\overline{x}$ clockwise. The space obtained by identifying edges x and $\overline{x}$, as indicated by their direction, is a closed oriented surface, denoted by $\mathbb{M}_g$; see e.g., [Sti93, Chapter 1.4]. This surface, called the orientable surface of genus g, is homeomorphic to a 2-sphere with g *handles.*

Let $\mathcal{N}_g(a_1, \cdots, a_g)$ be the regular $2g$-gon whose successive edges are labeled $a_1, a_1, \cdots, a_g, a_g$. Identifying edges in pairs, as indicated by their oriented labels, yields a closed nonorientable surface, denoted by $\mathbb{N}_g$. This surface, called the nonorientable surface of genus g, is homeomorphic to a 2-sphere with g *cross-caps.*

The labeled polygon $\mathcal{M}_g$ ($\mathcal{N}_g$) is called the polygonal schema of $\mathbb{M}_g$ ($\mathbb{N}_g$). $\mathbb{M}_1$ is the ***torus***, $\mathbb{N}_1$ is the ***projective plane***, $\mathbb{N}_2$ is the ***Klein bottle.***

Minimal triangulation: A triangulation of a surface is called minimal if it has no contractible edges (i.e., contracting an edge yields a subdivision that is not a triangulation).

EXAMPLES

1. A ***graph*** is a 1-dimensional simplicial complex. The complete graph with n vertices is the 1-skeleton of an $(n-1)$-simplex: $K_n = \sigma^1_{n-1}$.
2. Every connected, compact 1-manifold is topologically equivalent to $\mathbb{S}^1$ or $\mathbb{I}$.
3. The Delaunay triangulation of a set of points in general position in $\mathbb{R}^d$ is a simplicial complex.

BASIC PROPERTIES

1. Every triangulation of an orientable manifold has an orientation (i.e., the definition of orientability does not depend on the particular triangulation).
2. The Euler characteristic is a homotopy (and hence a topological) invariant (cf. Section 28.4).
3. A simplicial 2-complex is (topologically equivalent to) a closed surface iff every edge is incident with two faces, and the faces around a vertex can be ordered as $f_0, \cdots, f_{k-1}$ so that there is exactly one edge incident with both f_i and f_{i+1} (indices modulo k).
4. An oriented closed surface X is topologically equivalent to $\mathbb{S}^2$ if $\chi(X) = 2$, or to $\mathbb{M}_g$ if $\chi(X) \neq 2$, where g is uniquely determined by $\chi(X) = 2 - 2g$. A nonorientable closed surface X is topologically equivalent to $\mathbb{N}_g$, with $\chi(X) = 2 - g$. The number g is called the ***genus*** of the surface.
5. Every surface has finitely many minimal triangulations. (This number is 1 for $\mathbb{S}^2$, 2 for the projective plane, and 22 for the torus; cf. Section 18.2.)
6. A simplicial complex is a 3-manifold without boundary iff every 2-simplex is incident with exactly two 3-simplices and $\chi(M) = 0$. See [Fom91, p. 184].

7. Every combinatorial simplicial d-complex has a geometric realization in $\mathbb{R}^{2d+1}$.

8. Two geometric realizations K_1 and K_2 of a combinatorial simplicial complex $\mathcal{K}$ are simplicially equivalent (therefore the topology of K does not depend on the Euclidean space in which $\mathcal{K}$ is geometrically realized).

9. A simplicial map $f : |K| \to |L|$ is continuous. Hence both simplicial equivalence and PL-equivalence imply topological equivalence.

10. ***Hauptvermutung:*** Two simplicial complexes are PL-equivalent iff their underlying spaces are topologically equivalent. The Hauptvermutung is true if the underlying spaces are manifolds of dimension ≤ 3, and open for manifolds of dimension exceeding 3. It is false for general simplicial complexes, see Milnor [Mil61]. (*Reidemeister torsion* is a PL-invariant, but not a topological invariant [DFN90, pp. 156, 372].)

ALGORITHMS, DATA STRUCTURES, AND COMPLEXITY

Representation of spaces: The *Delaunay complex* D_X is a geometric simplicial complex which is, under some conditions, homotopy (or even topologically) equivalent to a given subspace X of some Euclidean space $\mathbb{R}^d$. See [ES94]. For applications of simplicial complexes to geometric modeling, see [Ede94].

Classification of surfaces: The Euler characteristic and orientability of a triangulated surface with n simplices can both be computed in $O(n)$ time.

Polygonal schema for a surface of genus $g > 0$: Given a triangulation of a closed orientable (nonorientable) surface of genus $g > 0$ with n triangles, there is a sequence of $O(n)$ elementary transformations (called ***cross-cap*** or ***handle normalizations***) that turns the triangulation into a polygonal schema of the form $\mathcal{M}_g$ ($\mathcal{N}_g$). This sequence of transformations can be computed in $O(n \log n)$ time [VY90].

Minimal triangulations of a surface: For a triangulation of a surface of genus g with n triangles, a sequence of $O(n)$ edge contractions leading to a minimal triangulation, can be computed in $O(n \log n)$ time [Sch91]. Therefore the classification problem for triangulated surfaces with n-triangles can be solved in $O(n)$ time; see property (4) above.

Isomorphism (simplicial equivalence): The isomorphism problem for simplicial 2-complexes is graph-isomorphism complete [OW+95]. It is unknown whether the graph-isomorphism problem is solvable in polynomial time (in the size of the graphs). See [vL90].

PL-equivalence: Deciding whether two arbitrary simplicial d-manifolds are PL-equivalent is unsolvable for $d \geq 4$ [Sti93, Chapter 9].

OPEN PROBLEMS

1. Design an algorithm that determines whether a simplicial 3-manifold is topologically equivalent to $\mathbb{S}^3$. This is a hard problem; see [VKF74] for partial results.

2. Design an algorithm that computes all minimal triangulations for a surface of genus g.

3. Determine the minimal size of a triangulation for a triangulable d-manifold [BK87, Sar87].

28.3 CELL COMPLEXES

Although simplicial complexes are convenient representations of topological spaces from an algorithmic point of view, they usually have many simplices. If a representation with a smaller number of cells is desirable, CW-complexes seem appropriate. See also Section 15.4.

GLOSSARY

Attaching cells to a space: Let X and Y be topological spaces, such that $X \subset Y$. We say that Y is obtained by attaching a (finite) collection of k-cells to X if $Y \setminus X$ is the disjoint union of a finite number of open k-balls $\{e_i^k \mid i \in I\}$, with the property that, for each i in the index set I, there is a map $f_i : \overline{\mathbb{B}}^k \to \overline{e}_i^k$, called the ***characteristic map*** of the cell e_i^k, such that $f_i(\mathbb{S}^{k-1}) \subset X$ and the restriction $f_i \mid \mathbb{B}^k$ is a homeomorphism $\mathbb{B}^k \to e_i^k$. (Note: $\overline{\mathbb{B}}^k$ need not be homeomorphic to $\overline{e}_i^k$.)

Cell complex (CW complex): A (finite) CW-decomposition of a topological space X is a finite sequence

$$\emptyset = X^{-1} \subset X^0 \subset X^1 \subset \cdots \subset X^d = X \tag{28.3.1}$$

such that (i) X^0 is a finite set of points, called the 0-cells of X; (ii) for $k > 0$, X^k is obtained from X^{k-1} by attaching a finite number of k-cells to X^{k-1}. The connected components of $X^k \setminus X^{k-1}$ are called the k-cells of X. The space X is called a (finite) CW-complex. The dimension of X is the maximal dimension of the cells of X. A finite CW-complex is called ***regular*** if the characteristic map of each cell is a homeomorphism. ("CW" stands for "Closure-finite with the Weak topology.")

EXAMPLES AND ELEMENTARY PROPERTIES

1. The d-sphere ($d > 0$) is a CW-complex, obtained by attaching a d-cell to a point p (so $X^k = \{p\}$, for $0 \leq k < d$, and $X^d = \mathbb{S}^d$). This CW-complex is not regular: the characteristic map of the d-cell maps the boundary of $\mathbb{B}^d$ to a single point.

2. The ***orientable surface*** $\mathbb{M}_g$ of genus $g > 1$ is a CW-complex with one 0-cell, $2g$ 1-cells, and one 2-cell. Let the 1-cells be $a_1, b_1, \cdots, a_g, b_g$, endowed with an orientation (direction). The characteristic map of the 2-cell is uniquely determined by attaching the labeled $4g$-gon $\mathcal{M}_g(a_1, b_1, \cdots, a_g, b_g)$ (cf. Section 28.2) to the 1-skeleton by mapping an edge to the 1-cell with the same

label, so that the directions of the edge and the 1-cell correspond. See [VY90]. The $2g$ 1-cells are curves on the surface, disjoint except at their common endpoint (which is the 0-cell). These curves are called ***canonical generators*** of the surface (see Section 28.4 for a justification of this nomenclature). The total number of cells is $2g + 2$, whereas the total number of simplices in a triangulation is at least $10g - 10 + \Theta(\sqrt{g})$ [JR80].

3. The ***nonorientable surface*** $\mathbb{N}_g$ of genus $g > 1$ is a CW-complex, with one 0-cell, g 1-cells, and one 2-cell. The characteristic map of the 2-cell is obtained from the polygonal schema represented by the $2g$-gon $\mathcal{N}_g(a_1, \cdots, a_g)$.

4. A geometric simplicial complex is a regular CW-complex.

5. The dual map of a triangulation of a surface is a regular CW-complex, but not a simplicial complex.

6. Examples of CW-complexes arising in computational geometry are: arrangements of hyperplanes in $\mathbb{R}^d$ (after addition of a point at infinity), the visibility complex [PV93], the free space of a polygonal robot moving amidst polygonal obstacles (see [SS83] and Chapter 40), and the zero-set of a generic polynomial defined on $\mathbb{S}^d \subset \mathbb{R}^{d+1}$.

ALGORITHMS AND DATA STRUCTURES

Representation: A data structure for the representation and manipulation of a finite, d-manifold CW-complex is described in [Bri93].

CW-decomposition of surfaces from triangulations: For a triangulated surface of genus g, with a total of n simplices, a set of canonical generators (cf. property (2)) can be computed in $O(gn)$ time, which is optimal in the worst case [VY90]. Each of the g or $2g$ canonical generators is represented by a polygonal curve whose vertices are on the 1-skeleton, while its other points are in the interior of a 2-simplex. In some cases the total number of edges of a single generator is $O(n)$. This method can be used to construct covering surfaces of m sheets in time $O(gnm)$ time and space; see also Section 28.4.

CW-decomposition in motion planning: A general method to solve motion planning problems is the construction of a cell decomposition (Equation 28.3.1) of the free space X of the robot, together with a ***retraction*** $r : X \to X^k$ of X onto a low-dimensional skeleton, such that there is a motion from initial position $x_0 \in X$ to final position $x_1 \in X$ iff there is a motion from $r(x_0)$ to $r(x_1)$. This may be regarded as a reduction of the degrees of freedom of the robot. Because in general the complexity of the motion planning problem is exponential in the number of degrees of freedom, this approach simplifies the problem. For more details on the cell decomposition method in motion planning, see Section 40.1.

28.4 ALGEBRAIC TOPOLOGY

In algebraic topology one associates homotopy invariant groups (homology and homotopy groups) to a space, and homotopy invariant homomorphisms to maps

between spaces. In passing from topology to algebra one may lose information since topologically distinct spaces may give rise to identical algebraic invariants. However, one gains on the algorithmic side, since the algebraic counterpart of an intractable topological problem may be tractable.

28.4.1 SIMPLICIAL HOMOLOGY GROUPS

Historically speaking, simplicial homology groups were among the first invariants associated with topological spaces. They are conceptually and algorithmically appealing. Modern algebraic topology usually deals with singular and cellular homology groups, which are more convenient from a mathematical point of view.

GLOSSARY

Ordered simplex: Let the vertices of a simplicial complex K be ordered $v_0, \cdots, v_m$. A k-simplex of K with vertices $v_{i_0}, \cdots, v_{i_k}$, $i_0 < \cdots < i_k$ is represented by the symbol $[v_{i_0}, \cdots, v_{i_k}]$, and called an ordered simplex.

Simplicial chain: If G is an abelian group, then an (ordered) simplicial k-chain is a formal sum of the form $\sum_j a_j \sigma_j$, with $a_j \in G$ and σ_j the symbol of a k-simplex in K. With the obvious definition for addition, the set of all (ordered) simplicial k-chains forms a (free) abelian group $C_k(K, G)$, called the group of (ordered) simplicial k-chains of K. If $G = \mathbb{Z}$, the group of integers, an element of $C_k(K, G)$ is called an ***integral k-chain.***

Boundary operator: The boundary operator $\partial_k : C_k(K, G) \to C_{k-1}(K, G)$ is defined as follows. For a single (ordered) k-simplex $\sigma = [v_{i_0}, \cdots, v_{i_k}]$, let $\partial_k \sigma = \sum_{h=0} (-1)^h [v_{i_0}, \cdots, \hat{v}_{i_h}, \cdots, v_{i_k}]$, and then let ∂_k be extended linearly, viz., $\partial_k(\sum_j a_j \sigma_j) = \sum_j a_j \partial_k \sigma_j$. The boundary operator is a homomorphism of groups. It satisfies $\partial_k \partial_{k+1} = 0$.

Simplicial k-cycles: $Z_k(K, G) = \ker \partial_k$ is called the group of (ordered) simplicial k-cycles.

Simplicial k-boundaries: $B_k(K, G) = \operatorname{im} \partial_{k+1}$ is called the group of (ordered) simplicial k-boundaries. Since the boundary of a boundary is 0, B_k is a subgroup of $Z_k(K, G)$.

Simplicial homology groups: The group $H_k(K, G) = Z_k(K, G)/B_k(K, G)$ is the kth (simplicial) homology group of K. This is a purely combinatorial object, since in fact it is defined for abstract simplicial complexes. If $G = \mathbb{Z}$, these groups are called ***integral homology groups***, usually denoted by $H_k(K)$. If G is a field (such as $\mathbb{R}$), then $H_k(K, G)$ is a vector space.

Homology groups of a triangulable topological space: $H_k(X, G) = H_k(K, G)$, if K is a simplicial complex triangulating X. This definition is independent of the triangulation K: if $h_i : K_i \to X$, $i = 1, 2$, are two triangulations of X, then $H_k(K_1, G) = H_k(K_2, G)$.

Betti numbers: The kth Betti number $\beta_k(K)$ of a simplicial complex K is the dimension of the real vector space $H_k(K, \mathbb{R})$. (For an alternative definition, see [Bre93, Chapter IV.1].)

Euler characteristic: The Euler characteristic $\chi(K)$ of a simplicial d-complex

K is defined by $\chi(K) = \sum_{i=0}^{d}(-1)^i\beta_i(K)$. This definition is equivalent to the one of Section 28.2.

EXAMPLES

1. The *n-sphere* ($n > 0$): $H_k(\mathbb{S}^n, \mathbb{Z}) = \mathbb{Z}$, if $k = 0$ or n, and 0 otherwise.
2. *Orientable surface*: For $g \geq 0$, $H_0(\mathbb{M}_g, G) = H_2(\mathbb{M}_g, G) = G$, $H_1(\mathbb{M}_g, G) = \oplus_{i=1}^{2g} G$, $H_k(\mathbb{M}_g, G) = 0$ for $k > 2$. Taking $G = \mathbb{R}$ we see that $\chi(\mathbb{M}_g) = 2 - 2g$.
3. *Nonorientable surface*: For $g \geq 0$, $H_0(\mathbb{N}_g, \mathbb{Z}) = \mathbb{Z}$, $H_1(\mathbb{N}_g, \mathbb{Z}) = \oplus_{i=1}^{g-1}\mathbb{Z} \oplus \mathbb{Z}_2$, $H_k(\mathbb{N}_g, \mathbb{Z}) = 0$ for $k \geq 2$. $H_0(\mathbb{N}_g, \mathbb{R}) = \mathbb{R}$, $H_1(\mathbb{N}_g, \mathbb{R}) = \oplus_{i=1}^{g-1}\mathbb{R}$, $H_2(\mathbb{N}_g, \mathbb{R}) = 0$. Hence, $\chi(\mathbb{N}_g) = 2 - g$.

BASIC PROPERTIES

1. Homology is a homotopy invariant: if X_1 and X_2 are homotopy equivalent, then $H_k(X_1) = H_k(X_2)$ for all k. In particular, Betti numbers and the Euler characteristic are homotopy invariants.
2. For a simplicial d-complex K: $H_k(K, G) = 0$ for $k > d$.
3. Let $\alpha_i(K)$ be the number of i-simplices of a simplicial d-complex K. Then $\chi(K) = \sum_{i=0}^{d}(-1)^i\alpha_i(K)$. This justifies the definition of χ in Section 28.2.

COMPUTING HOMOLOGY GROUPS

See Table 28.4.1 for the algorithmic complexity of computing the homology group of several important types of spaces.

TABLE 28.4.1 Complexity to compute homology groups.

TYPE OF SPACE	COMPLEXITY	SOURCE
Simplicial subcomplex of $\mathbb{S}^3$ of size n	$O(n\alpha(n))$	[DE93]
Sparse simplicial complex of size n	$O(n^2)$ (probabilistic)	[DC91]
Semi-algebraic set, defined by m poly's (deg $\leq d$) on $\mathbb{R}^n$, n fixed	polynomial in m, d	[SS83]

28.4.2 HOMOTOPY GROUPS

Homotopy groups usually provide more information than homology groups, but are generally harder to compute. The main object is the fundamental group, whose computation requires some combinatorial group theory.

GLOSSARY

Fundamental group: The space of x_0-based curves on X is endowed with a group structure by (group multiplication) $(u_1 \cdot u_2)(t) = u_1(2t)$, if $0 \leq t \leq \frac{1}{2}$, and $u_2(2t-1)$ if $\frac{1}{2} \leq t \leq 1$, and (inverse) $u^{-1}(t) = u(1-t)$.

This group structure can be extended to homotopy classes of x_0-based curves: If u, v are homotopic, then u^{-1} and v^{-1} are homotopic, and if u_i and v_i, $i = 1, 2$, are homotopic, then $u_1 \cdot u_2$ and $v_1 \cdot v_2$ are homotopic (homotopies respect the basepoint x_0). The group of homotopy classes of closed x_0-based curves is called the fundamental group (or, the ***first homotopy group***) of (X, x_0), and is denoted by $\pi_1(X, x_0)$. If X is connected, the definition is independent of the basepoint. Then the fundamental group is denoted by $\pi_1(X)$.

Combinatorial definition of the fundamental group: If X is a connected space with triangulation K and vertices $a_0, \cdots, a_m$, then the fundamental group has generators g_{ij}, one per ordered 1-simplex $[a_i, a_j]$, and relations $g_{ij}g_{jk}g_{ik}^{-1} = 1$, one for each ordered 2-simplex $[a_i, a_j, a_k]$ [Mau70, Chapter 3]. See [Sti93] for an introduction to combinatorial group theory.

k-th homotopy group: Let $s_0 \in \mathbb{S}^k$, for $k \geq 1$. The space of homotopy classes of basepoint-preserving maps $(\mathbb{S}^k, s_0) \to (X, x_0)$ can be endowed with a group structure. The group is called the kth homotopy group of (X, x_0), and is denoted by $\pi_k(X, x_0)$.

Word problem for a group G: Given a (finitely generated) group generated by $g_1, \ldots, g_k$ (the alphabet), and a finite set of relations of the form $g_1^{m_1} \cdots g_k^{m_k} = 1$ (rewrite rules) with $m_i \in \mathbb{Z}$, decide whether a given word of the form $g_1^{n_1} \cdots g_k^{n_k}$ represents the unit element 1.

Covering space: A continuous map $p : Y \to X$ is a covering map if every point $x \in X$ has a connected neighborhood U such that for each connected component V of $p^{-1}(x)$ the restriction of p to V is a homeomorphism $V \to U$. Y is called a covering space of X. If the cardinality n of $p^{-1}(U)$ is finite, Y is called an n-sheeted cover of X. This number is the same for all $x \in X$.

Universal covering space: A connected covering space Y of X is called universal if $\pi_1(Y) = 0$.

EXAMPLES

1. *The n-sphere* $(n > 0)$: $\pi_1(\mathbb{S}^n, s_0) = \mathbb{Z}$ if $n = 1$, and 0 otherwise.
2. *Orientable surface of genus* $g \geq 1$: $\pi_1(\mathbb{M}_g)$ is generated by $2g$ generators $a_1, b_1, \cdots, a_g, b_g$, with the single relation $a_1 b_1 a_1^{-1} b_1^{-1} \cdots a_g b_g a_g^{-1} b_g^{-1} = 1$.
3. *Nonorientable surface of genus* $g \geq 1$: $\pi_1(\mathbb{N}_g)$ is generated by g generators $a_1, \cdots, a_g$, with the single relation $a_1 a_1 \cdots a_g a_g = 1$.
4. *Universal covering space*: The universal covering space of $\mathbb{S}^1$ is $\mathbb{R}$, with covering map $p : \mathbb{R} \to \mathbb{S}^1$ defined by $p(t) = (\cos t, \sin t)$. The universal covering space of the projective plane $\mathbb{P}$ is $\mathbb{S}^2$, the covering map being antipodal identification. The plane is the universal covering space of $\mathbb{M}_g$ and $\mathbb{N}_g$, $g > 0$.

BASIC PROPERTIES

1. The homotopy groups are homotopy invariants.
2. The first integral homology group is the abelianized fundamental group.
3. The fundamental group of a simplicial complex is the fundamental group of its 2-skeleton.
4. For every finitely generated group G there is a finite simplicial 2-complex K and a 4-manifold M such that $\pi_1(K) = G$ and $\pi_1(M) = G$.
5. Homotopy invariants are topological invariants, but not vice versa. For example, the *lens spaces* $L(5,1)$ and $L(5,2)$ are not homotopy equivalent, but do have isomorphic homology and homotopy groups [Bre93, Chapter VI].
6. Let Y be a universal covering space of X with covering map $p : Y \to X$, and let $y_0 \in Y$ and $x_0 = p(y_0) \in X$. Every curve $c : \mathbb{I} \to X$ with $c(0) = x_0$ has a unique lift $\overline{c} : \mathbb{I} \to Y$ with $c(0) = y_0$. Furthermore, a closed curve c is contractible in X iff $\overline{c}$ is a closed curve in Y, i.e., $\overline{c}(1) = y_0$ [Sti93, Chapter 6]. This is the basis of Dehn's algorithm for the contractibility problem on surfaces (see below).

ALGORITHMS AND COMPLEXITY

Undecidability of homeomorphism problem: The word problem for general groups is undecidable. Hence the contractibility problem for general simplicial 2-complexes, and for manifolds of dimension ≥ 4, is undecidable [Sti93]. A slight variation even proves that the homeomorphism problem for 4-manifolds is undecidable.

Contractibility problem for surfaces: Determine whether a curve with k edges on a triangulated surface $\mathbb{M}_g$ of size n is contractible, and, if so, construct a contraction.

Schipper [Sch92a] and Dey [Dey94] implement Dehn's algorithm by constructing a finite portion of the covering surface of $\mathbb{M}_g$, for $g > 1$, and determining whether the lift of the curve to the covering space is closed. These algorithms can also be applied to solving the homotopy problem for curves on a surface. An algorithmic solution of the word problem for surface groups is described in [Sch92b].

TABLE 28.4.2 Algorithm complexities.

ALGORITHM	TIME	SPACE	SOURCE
Decision	$O(g^2k + gn)$	$O(g^2k + gn)$	[Sch92a]
Construction	$O(g^2kn)$	$O(g^2k + gn)$	[Sch92a]
Decision	$O(gk + n)$	$O(gk + n)$	[Dey94]
Construction	$O(gkn)$	$O(gk + n)$	[Dey94]

Representation problem: There is an algorithm that decides whether a homotopy class of curves contains a simple closed curve. The algorithm of [Chi72] can be turned into a polynomial-time algorithm using methods similar to those of [Sch92a] and [VY90]. (Poincaré had already given a condition for a *homology* class of a curve on a surface to contain a simple closed curve. This can also be turned into a polynomial algorithm along similar lines.)

Shortest paths: Hershberger and Snoeyink [HS94b] consider polygonal curves in the complement of a set of polygons in the plane. They construct part of the covering space to compute minimum length curves that are homotopy equivalent to a given curve, in $O(k+n)$ time, where k is the number of edges of the curve and n is the total number of vertices of the polygons. See Section 24.2.

28.5 EMBEDDING SIMPLICIAL COMPLEXES

Embeddability problems are important for their own sake, but also for computations. Especially important algorithmically is the problem of embedding a simplicial complex in a Euclidean space of lowest dimension. See also Section 18.1.

GLOSSARY

Simplicial embedding of a simplicial complex K in simplicial complex L: A simplicial map $f : |K| \to |L|$ that is a topological embedding.

Geometric embedding of a simplicial complex K in $\mathbb{R}^d$: A simplicial equivalence $f : K \to L$, where L is a geometric simplicial complex in $\mathbb{R}^d$.

Piecewise-linear (PL) embedding of a simplicial complex K in a simplicial complex L: A simplicial embedding of a refinement K' of K in a refinement L' of L. If L is a geometric simplicial complex in $\mathbb{R}^d$, we say that K can be PL-embedded in $\mathbb{R}^d$.

PL-*minimality:* A simplicial complex is PL-minimal in $\mathbb{R}^d$ if is not PL-embeddable in $\mathbb{R}^d$, but every proper subcomplex can be PL-embedded in $\mathbb{R}^d$.

Genus of a graph: The orientable (nonorientable) genus of a graph G is the minimal genus of an orientable (nonorientable) surface in which G is PL-embeddable.

Book: A book with p pages is a simplicial complex consisting of p triangles sharing a common edge (and nothing else).

Page number of a graph: Minimal number of pages of a book in which the graph is PL-embeddable.

28.5.1 PL-EMBEDDINGS

BASIC RESULTS

1. A simplicial d-complex that is topologically embeddable in $\mathbb{R}^{2d}$ is also PL-embeddable in $\mathbb{R}^{2d}$ [Web67].

2. For $d \geq 3$, a simplicial d-complex K is PL-embeddable in $\mathbb{R}^{2d}$ iff its *van Kampen obstruction class* $o(K) = 0$. ($o(K)$ is an element of the $2d$-th cohomology group of the symmetric product of K minus the diagonal; see [vK33, Sha57].) If K is a triangulation of a d-manifold, then $o(K) = 0$, so K can be embedded in $\mathbb{R}^{2d}$ [Whi44].

3. ***Kuratowski's theorem***: a graph G is PL-embeddable in the plane iff K_5 and $K_{3,3}$ are not PL-embeddable in G. The graphs K_5 and $K_{3,3}$ are called ***forbidden minors*** for planarity.

4. Every orientable triangulated surface can be PL-embedded in $\mathbb{R}^3$. Every nonorientable triangulated surface can be PL-embedded in $\mathbb{R}^4$, but not in $\mathbb{R}^3$ (for a simple proof of the latter, see [Mae93]).

5. Kuratowski's theorem can be rephrased by saying that K_5 and $K_{3,3}$ are the only PL-minimal 1-complexes in $\mathbb{R}^2$. For each $n \geq 2$ and each d, with $n+1 \leq d \leq 2n$, there are countably many nonhomeomorphic n-complexes that are all PL-minimal in $\mathbb{R}^d$ [Zak69].

6. There is a finite set of forbidden minors for PL-embeddability in a surface of fixed genus g [RS90].

7. The page-number of a graph is $O(g)$ [HI92].

ALGORITHMS AND COMPLEXITY

PL-embeddability of graphs: It can be decided in $O(n \log n)$ time whether a graph with n vertices is planar (PL-embeddable in the plane). In $O(n \log n)$ time a geometric embedding in the plane can be constructed [HT74].

Graph genus: The graph genus problem is NP-complete [Tho89].

OPEN PROBLEMS

1. Give an efficient algorithm that computes the van Kampen obstruction $o(K)$ for a simplicial d-complex K with a total of n simplices. Find an algorithm that constructs a PL-embedding (of reasonable complexity) for K in case $o(K) = 0$.

2. Design an efficient algorithm that determines whether a simplicial d-complex can be PL-embedded in $\mathbb{R}^k$, for $d \leq k < 2d$.

28.5.2 GEOMETRIC EMBEDDINGS

MAIN RESULTS

1. Every simplicial d-complex can be geometrically embedded in $\mathbb{R}^{2d+1}$.

2. Every simplicial 1-complex (graph) that is PL-embeddable in $\mathbb{R}^2$ can be geometrically embedded in $\mathbb{R}^2$ (***Fary's theorem***).

3. For each $d \geq 2$ there is a simplicial d-complex that is PL-embeddable in $\mathbb{R}^{d+1}$, but not geometrically embeddable in $\mathbb{R}^{d+1}$ [Duk70].

4. All minimal triangulations of the 2-sphere and the torus can be geometrically embedded in $\mathbb{R}^3$ [BW93]. All minimal triangulations of the projective plane can be geometrically embedded in $\mathbb{R}^4$ [BW93].

ALGORITHMS

Geometric embeddability of a graph: It can be decided in $O(n \log n)$ time whether a simplicial 1-complex (graph) with n cells (edges and vertices) can be geometrically embedded in the plane. If such an embedding exists, it can be constructed in $O(n \log n)$ time [HT74].

OPEN PROBLEMS

1. Can every minimal triangulation (see Section 28.2) of the surface of genus g be geometrically embedded in $\mathbb{R}^3$ (cf. [BW93])?

2. Design an efficient (polynomial-time) algorithm that determines whether a simplicial d-complex can be geometrically embedded in $\mathbb{R}^k$, for $d \leq k \leq 2d$.

3. Prove or disprove: If a simplicial d-complex is PL-embeddable in $\mathbb{R}^{2d}$, then it is geometrically embeddable in $\mathbb{R}^{2d}$.

4. Is there a constant c such that the cth barycentric subdivision of any simplicial complex K whose underlying space can be PL-embedded in $\mathbb{R}^d$, can be geometrically embedded in $\mathbb{R}^d$? Recall that there are examples of simplicial complexes that are PL-embeddable in $\mathbb{R}^d$, but not geometrically embeddable.

28.5.3 KNOTS

GLOSSARY

Knot: A PL-embedding of a polygon in $\mathbb{R}^3$.

Spanning surface of a knot: A PL-embedded orientable surface in $\mathbb{R}^3$, whose boundary is the knot (also called a ***Seifert surface***).

Trivial knot: A knot with a spanning surface that is PL-equivalent to a disk.

Genus of a knot: Minimum possible genus of a spanning surface. (The genus of a spanning surface is the genus of the closed orientable surface obtained by attaching a disk—cf. Section 28.3—along the boundary of the spanning surface. In particular, a trivial knot has genus 0.)

ALGORITHMS AND COMPLEXITY

1. There is a trivial polygonal knot with n vertices for which every spanning disk has size exponential in n [Sno90].
2. A spanning surface for a polygonal knot with n vertices can be constructed in $O(n^2)$ time (Seifert's construction [Liv93]).
3. There is an algorithm that solves the ***knot triviality*** problem, i.e., that decides whether a polygonal knot with n vertices is trivial, in time doubly-exponential in n (the Haken-Hemion algorithm; see [Hem92]).

OPEN PROBLEM

Knot triviality: Prove that the knot triviality problem is NP-hard or NP-complete (it is not known to be in NP).

28.6 IMMERSIONS

GLOSSARY

Immersion: Let K and L be simplicial complexes. A PL-map $f : |K| \to |L|$ is called an immersion if it is locally injective (i.e., every point $p \in |K|$ has a neighborhood in $|K|$ on which f is 1–1). We say that $|K|$ is immersed in $|L|$. An immersion of $|K|$ in $\mathbb{R}^d$ is defined similarly.

Regular equivalence of immersions: Two immersions f_0 and f_1 of $|K|$ in $|L|$ (or $\mathbb{R}^d$) are regularly equivalent if there is a homotopy F, between f_0 and f_1, defined on $|K| \times \mathbb{I}$, such that f_t, defined by $f_t(x) = F(x,t)$, is an immersion of $|K|$ in $|L|$ (or $\mathbb{R}^d$).

Winding number: Consider a polygon P with n vertices, immersed in the plane. Let its exterior angles $\theta_1, \cdots, \theta_n$, be measured with sign. The winding number of P is $w(P) = \frac{1}{2\pi} \sum_{i=1}^{n} \theta_i \in \mathbb{Z}$ (the total number of turns of its tangent vector). P may be considered as the image of a PL-immersion $c : \mathbb{S}^1 \to \mathbb{R}^2$, for which we define $w(c) = w(P)$.

BASIC RESULTS

1. Every PL-embedding is an immersion.
2. Every simplicial d-manifold can be immersed in $\mathbb{R}^{2d-1}$ [Whi44].
3. Two immersions $c_1, c_2 : \mathbb{S}^1 \to \mathbb{R}^2$ are regularly equivalent iff $w(c_1) = w(c_2)$ (a theorem of Whitney-Graustein).
4. There are two regular equivalence classes of immersions $\mathbb{S}^1 \to \mathbb{S}^2$, viz the curves that go once and twice along the equator of $\mathbb{S}^2$.

5. Smale [Sma58] associates with each immersion $c : \mathbb{S}^1 \to \mathbb{M}_g$ an element $W(c)$ of the fundamental group of the unit tangent bundle $S^1(\mathbb{M}_g)$ of $\mathbb{M}_g$ that is a complete invariant for the regular equivalence class of c. This element $W(c)$ may be considered the generalization of the winding number of an immersion of $\mathbb{S}^1$ in $\mathbb{R}^2$. For related definitions, see [Chi72, MC93].

6. All immersions of $\mathbb{S}^2$ in $\mathbb{R}^3$ are regularly equivalent. See [Sma60] and [Fra87, Phi66] for pictures and constructions.

7. More generally, there are 4^g regular equivalence classes of immersions of an oriented closed surface in $\mathbb{R}^3$ [JT66]. See [Phi66] for pictures of the 4 classes of immersions of the torus $\mathbb{M}_1$ in $\mathbb{R}^3$.

ALGORITHMS

Kinkfree deformations of immersed curves in $\mathbb{R}^2$: If $w(P_1) = w(P_2)$ for planar polygons P_1 and P_2 with a total of n vertices, there is a sequence of $O(n)$ "elementary" moves that realizes a regular equivalence between P_1 and P_2. This sequence can be computed in $O(n \log n)$ time [Veg89]. This algorithm can be adapted to construct a regular equivalence between two polygonal curves on $\mathbb{S}^2$.

Regular closed curves on $\mathbb{M}_g$, $g > 0$: There is an algorithm that determines in polynomial time whether two PL-immersions $\mathbb{S}^1 \to \mathbb{M}_g$ are regularly equivalent [Chi72, MC93].

OPEN PROBLEMS

1. *Regular deformations of curves on a surface*: Design an optimal algorithm that determines whether two PL-immersions $\mathbb{S}^1 \to \mathbb{M}_g$ are regularly equivalent, and, if so, construct such an equivalence.

2. *Immersions of $\mathbb{S}^2$ in $\mathbb{R}^3$*: Design an efficient algorithm that constructs a regular equivalence between two arbitrary PL-immersions of $\mathbb{S}^2$ in $\mathbb{R}^3$.

3. *Immersions of $\mathbb{M}_g$ in $\mathbb{R}^3$*: Design an algorithm that determines whether two immersions of $\mathbb{M}_g$ in $\mathbb{R}^3$ are regularly equivalent. Extend the method to the construction of such an equivalence.

28.7 SOURCES AND RELATED MATERIAL

FURTHER READING

[Sti93]: Low dimensional topology, including some knot theory and relationships with combinatorial group theory. Good starting point for exploration of topology; nice historical setting.

[Fom91]: User-friendly introduction to algebraic topology and the classification problem for manifolds.

[ST34]: A classic, dealing with combinatorial algebraic topology.

[Mau70]: Extensive treatment of simplicial complexes and simplicial algebraic topology.

[Bre93]: Modern textbook on algebraic topology, especially as related to topological aspects of manifold theory.

RELATED CHAPTERS

Chapter 15: Face numbers of polytopes and complexes
Chapter 18: Polyhedral maps
Chapter 29: Computational real algebraic geometry

REFERENCES

[Bak90] D.R. Baker. Some topological problems in robotics. *Math. Intelligencer*, 12:66–76, 1990.

[BK87] U. Brehm and W. Kühnel. Combinatorial manifolds with few vertices. *Topology*, 26:465–473, 1987.

[Bre93] G.E. Bredon. *Topology and Geometry*, volume 139 of *Grad. Texts in Math.* Springer-Verlag, New York, 1993.

[Bri93] E. Brisson. Representing geometric structures in d dimensions: Topology and order. *Discrete Comput. Geom.*, 9:387–426, 1993.

[BW93] U. Brehm and J.M. Wills. Polyhedral manifolds. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 535–554. Elsevier, Amsterdam, 1993.

[Chi72] D.R.J. Chillingworth. Winding numbers on surfaces, I and II. *Math. Ann.*, 196:218–249 and 199:131–153, 1972.

[DC91] B.R. Donald and D.R. Chang. On the complexity of computing the homology type of a triangulation. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 650–662, 1991.

[DE93] C.J.A. Delfinado and H. Edelsbrunner. An incremental algorithm for Betti numbers of simplicial complexes. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 232–239, 1993.

[Dey94] T.K. Dey. A new technique to compute polygonal schema for 2-manifolds with application to null-homotopy detection. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 277–284, 1994.

[DFN90] B.A. Dubrovin, A.T. Fomenko, and S.P. Novikov. *Modern Geometry—Methods and Applications, Part III*, volume 124 of *Grad. Texts in Math.* Springer-Verlag, New York, 1990.

[Duk70] R.A. Duke. Geometric embedding of complexes. *Amer. Math. Monthly*, 77:597–603, 1970.

[Ede94] H. Edelsbrunner. Modeling with simplicial complexes: Topology, geometry, and algorithms. In *Proc. 6th Canad. Conf. Comput. Geom.*, pages 36–44, 1994.

[ES94] H. Edelsbrunner and N.R. Shah. Triangulating topological spaces. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 285–292, 1994.

[Fom91] A. Fomenko. *Visual Geometry and Topology.* Springer-Verlag, New York, 1991.

[Fra87] G.K. Francis. *A Topological Picture Book.* Springer-Verlag, Berlin, 1987.

[Hem92] G. Hemion. *The Classifiaction of Knots and 3-dimensional Spaces.* Oxford University Press, New York, 1992.

[HI92] L.S. Heath and S. Istrail. The pagenumber of genus g graphs is $O(g)$. *J. Assoc. Comput. Mach.*, 39:479–501, 1992.

[HS94a] M. Herlihy and N. Shavit. Applications of algebraic topology to concurrent computation. *SIAM News*, pages 10–12, December 1994.

[HS94b] J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. *Comput. Geom. Theory Appl.*, 4:63–98, 1994.

[HT74] J. Hopcroft and R.E. Tarjan. Efficient planarity testing. *J. Assoc. Comput. Mach.*, 21:549–568, 1974.

[JR80] M. Jungerman and G. Ringel. Minimal triangulations of orientable surfaces. *Acta Math.*, 145:121–154, 1980.

[JT66] I. James and E. Thomas. On the enumeration of cross-sections. *Topology*, 5:95–114, 1966.

[Liv93] C. Livingston. *Knot Theory*, volume 24 of *The Carus Math. Monogr.* Math. Assoc. of Amer., Providence, 1993.

[Mae93] H. Maehara. Why is P^2 not embeddable in R^3? *Amer. Math. Monthly*, 100:862–864, 1993.

[Mau70] S.R.F. Maunder. *Algebraic Topology.* Van Nostrand Reinhold, London, 1970.

[MC93] M. Mcintyre and G. Cairns. A new formula for winding number. *Geom. Dedicata*, 46:149–160, 1993.

[Mil61] J. Milnor. Two complexes which are homeomorphic but combinatorially distinct. *Ann. of Math.*, 74:575–590, 1961.

[OW+95] C.Ó Dúnlaing, C. Watt, D. Wilkins, and C.K. Yap. Miscellaneous topological algorithms. Tech. Rep. ALCOM-II-430, Univ. of Dublin, Trinity College, 1995.

[Phi66] A. Phillips. Turning a surface inside out. *Sci. Amer.*, 214:112–120, May 1966.

[PV93] M. Pocchiola and G. Vegter. The visibility complex. *Internat. J. Comput. Geom. Appl.*, 6:279–308, 1996.

[RS90] N. Robertson and P.D. Seymour. Graph minors VIII: A Kuratowski theorem for general surfaces. *J. Combin. Theory Ser. B*, 48:255–288, 1990.

[Sar87] K.S. Sarkaria. Heawood inequalities. *J. Combin. Theory Ser. A*, 46:50–78, 1987.

[Sch91] H. Schipper. Generating triangulations of 2-manifolds. In *Computational Geometry—Methods, Algorithms and Applications: Proc. Internat. Workshop Comput. Geom. CG '91*, volume 553 of *Lecture Notes in Comput. Sci.*, pages 237–248. Springer-Verlag, New York, 1991.

[Sch92a] H. Schipper. Determining contractibility of curves. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 358–367, 1992.

[Sch92b] H. Schipper. The word problem: A geometric approach. In *Proc. 4th Canad. Conf. Comput. Geom.*, pages 59–65, 1992.

[Sha57] A. Shapiro. Obstructions to the imbedding of a complex in a Euclidean space. *Ann. of Math.*, 66:256–269, 1957.

[Sma58] S. Smale. Regular curves on Riemannian manifolds. *Trans. Amer. Math. Soc.*, 87:492–512, 1958.

[Sma60] S. Smale. A classification of immersions of the two-sphere. *Trans. Amer. Math. Soc.*, pages 281–290, 1960.

[Sno90] J. Snoeyink. A trivial knot whose spanning disks have exponential size. In *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, pages 139–147, 1990.

[SS83] J.T. Schwartz and M. Sharir. On the piano mover's problem: II. General techniques for computing topological properties of real algebraic manifolds. *Adv. Appl. Math.*, 4:298–351, 1983.

[ST34] H. Seifert and W. Threlfall. *Lehrbuch der Topologie.* Teubner, Leipzig, 1934. English translation, *A Textbook of Topology.* Academic Press, New York, 1980.

[Sti93] J. Stillwell. *Classical Topology and Combinatorial Group Theory*, volume 72 of *Grad. Texts in Math.* Springer-Verlag, New York, 1993.

[Tho89] C. Thomassen. The graph genus problem is NP-complete. *J. Algorithms*, 10:568–576, 1989.

[Veg89] G. Vegter. Kink-free deformations of polygons. In *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, pages 61–68, 1989.

[vK33] E.R. van Kampen. Komplexe in euklidischen Räumen. *Abh. Math. Sem. Hamb.*, 9:72–78 and 152–153, 1933.

[VKF74] I.A. Volodin, V.E. Kuznetsov, and A.T. Fomenko. The problem of discriminating algorithmically the standard three-dimensional sphere. *Russ. Math. Surveys*, 29:71–172, 1974.

[vL90] J. van Leeuwen. Graph algorithms. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, pages 525–631. Elsevier, Amsterdam, 1990.

[VY90] G. Vegter and C.K. Yap. Computational complexity of combinatorial surfaces. In *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, pages 102–111, 1990.

[Web67] C. Weber. Plongement des polyèdres dans le domaine métastable. *Comm. Math. Helv.*, 42:1–27, 1967.

[Whi44] H. Whitney. The self-intersections of a smooth n-manifold in $2n$-space. *Ann. of Math.*, 45:221–246, 1944.

[Yap87] C.K. Yap. Algorithmic motion planning. In J.T. Schwartz and C.K. Yap, editors, *Algorithmic and Geometric Aspects of Robotics*, volume I of *Advances in Robotics*, chapter 3, pages 95–143. Lawrence Erlbaum, Hillsdale, 1987.

[Zak69] J. Zaks. On minimal complexes. *Pacific J. Math.*, 28:721–727, 1969.

29 COMPUTATIONAL REAL ALGEBRAIC GEOMETRY

Bhubaneswar Mishra

INTRODUCTION

Computational real algebraic geometry studies various algorithmic questions dealing with the *real solutions* of a system of equalities, inequalities, and inequations of polynomials over the real numbers. This emerging field is largely motivated by the power and elegance with which it solves a broad and general class of problems arising in robotics, vision, computer-aided design, geometric theorem proving, etc.

The algorithmic problems that arise in this context are formulated as decision problems for the *first-order theory of reals* and the related problems of *quantifier elimination* (Section 29.1). The associated geometric structures are then examined via an exploration of the *semialgebraic sets* (Section 29.2). Algorithmic problems for semialgebraic sets are considered next. In particular, Section 29.3 discusses real algebraic numbers and their representation, relying on such classical theorems as Sturm's theorem and Thom's Lemma (Section 29.3). This discussion is followed by a description of semialgebraic sets using the concept of *cylindrical algebraic decomposition* (CAD) in both one and higher dimensions (Sections 29.4 and 29.5). This leads to brief descriptions of two algorithmic approaches for the decision and quantifier elimination problems (Section 29.6): namely, Collins's algorithm based on CAD, and some more recent approaches based on critical points techniques and on reducing the multivariate problem to easier univariate problems. These new approaches rely on the work of several groups of researchers: Grigor'ev and Vorobjov, Canny, Heintz et al., Renegar, and Basu et al. A few representative applications of computational algebra conclude this chapter (Section 29.7).

29.1 FIRST-ORDER THEORY OF REALS

The *decision problem* for the first-order theory of reals is to determine if a *Tarski sentence* in the first-order theory of reals is true or false. The *quantifier elimination problem* is to determine if there is a logically equivalent quantifier-free formula for an arbitrary Tarski formula in the first-order theory of reals. As a result of Tarski's work, we have the following theorem.

THEOREM 29.1.1 [Tar51]

- Let Ψ be a Tarski sentence. There is an effective decision procedure for Ψ.
- Let Ψ be a Tarski formula. There is a quantifier-free formula ϕ logically equivalent to Ψ. If Ψ involves only polynomials with rational coefficients, then so does the sentence ϕ.

Tarski formulas are formulas in a first-order language (defined by Tarski in 1930) constructed from equalities, inequalities, and inequations of polynomials over the reals. Such formulas may be constructed by introducing logical connectives and universal and existential quantifiers to the atomic formulas. *Tarski sentences* are Tarski formulas in which all variables are bound by quantification.

GLOSSARY

Term: A constant, variable, or term combining two terms by an arithmetic operator: $\{+, -, \cdot, /\}$. A constant is a real number. A variable assumes a real number as its value. A term contains finitely many such algebraic variables: $x_1, x_2, \ldots, x_n$.

Atomic formula: A formula comparing two terms by a binary relational operator: $\{=, \neq, >, <, \geq, \leq\}$.

Quantifier-free formula: An atomic formula, a negation of a quantifier-free formula given by the unary Boolean connective $\{\neg\}$, or a formula combining two quantifier-free formulas by a binary Boolean connective: $\{\Rightarrow, \wedge, \vee\}$. *Example*: The formula $(x^2 - 2 = 0) \wedge (x > 0)$ defines the (real algebraic) number $+\sqrt{2}$.

Tarski formula: If $\phi(y_1, \ldots, y_r)$ is a quantifier-free formula, then it is also a Tarski formula. All the variables y_i are ***free*** in ϕ. Let $\Phi(y_1, \ldots, y_r)$ and $\Psi(z_1, \ldots, z_s)$ be two Tarski formulas (with free variables y_i and z_i, respectively); then a formula combining Φ and Ψ by a Boolean connective is a Tarski formula with free variables $\{y_i\} \cup \{z_i\}$. Lastly, if $\mathcal{Q}$ stands for a quantifier (either universal $\forall$ or existential $\exists$) and if $\Phi(y_1, \ldots, y_r, x)$ is a Tarski formula (with free variables x and y), then

$$\Big(\mathcal{Q}\, x\Big)\, \Big[\Phi(y_1, \ldots, y_r, x)\Big]$$

is a Tarski formula with only the y's as free variables. The variable x is ***bound*** in $(\mathcal{Q}\, x)[\Phi]$.

Tarski sentence: A Tarski formula with no free variable.
Example: $(\exists\, x)\ (\forall\, y)\ [y^2 - x < 0]$. This Tarski sentence is false.

Prenex Tarski formula: A Tarski formula of the form

$$\Big(\mathcal{Q}\, x_1\Big)\, \Big(\mathcal{Q}\, x_2\Big)\, \cdots \Big(\mathcal{Q}\, x_n\Big)\, \Big[\phi(y_1, y_2, \ldots, y_r, x_1, \ldots, x_n)\Big],$$

where ϕ is quantifier-free. The string of quantifiers $(\mathcal{Q}\, x_1)\ (\mathcal{Q}\, x_2) \cdots (\mathcal{Q}\, x_n)$ is called the ***prefix*** and ϕ is called the ***matrix***.

Prenex form of a Tarski formula, Ψ: A prenex Tarski formula logically equivalent to Ψ. For every Tarski formula, one can find its prenex form using a simple procedure that works in four steps: (1) eliminate redundant quantifiers; (2) rename variables so that the same variable does not occur as free and bound; (3) move negations inward; and finally, (4) push quantifiers to the left.

Extension of a Tarski formula, $\Phi(y_1, \ldots, y_r)$ with free variables $\{y_1, \ldots, y_r\}$: The set of all $\langle \zeta_1, \ldots, \zeta_r \rangle \in \mathbb{R}^r$ such that

$$\Phi(\zeta_1, \ldots, \zeta_r) = \mathsf{True}.$$

THE DECISION PROBLEM

The general ***decision problem*** for the first-order theory of reals is to determine if a given Tarski sentence is true or false. A particularly interesting special case of the problem is when all the quantifiers are existential. We refer to the decision problem in this case as the ***existential problem*** for the first-order theory of reals.

The general decision problem was shown to be decidable by Tarski [Tar51]. However, the complexity of Tarski's original algorithm could only be given by a very rapidly-growing function of the input size (e.g., a function that could not be expressed as a bounded tower of exponents of the input size). The first algorithm with substantial improvement over Tarski's algorithm was due to Collins [Col75]; it has a doubly-exponential time complexity in the number of variables appearing in the sentence. Further improvements have been made by a number of researchers (Grigor'ev-Vorobjov [Gri88, GV88], Canny [Can88b, Can93], Heintz et al. [HRS89, HRS90], Renegar [Ren92a,b,c]) and most recently by Basu et al. [BPR].

In the following, we assume that our Tarski sentence is presented in its prenex form:

$$(\mathcal{Q}_1 \mathbf{x}^{[1]})\ (\mathcal{Q}_2 \mathbf{x}^{[2]})\ \cdots\ (\mathcal{Q}_\omega \mathbf{x}^{[\omega]})\ [\psi(\mathbf{x}^{[1]}, \ldots, \mathbf{x}^{[\omega]})],$$

where the $\mathcal{Q}_i$'s form a sequence of alternating quantifiers (i.e., $\forall$ or $\exists$, with every pair of consecutive quantifiers distinct), with $\mathbf{x}^{[i]}$ a partition of the variables

$$\bigcup_{i=0}^{\omega} \mathbf{x}^{[i]} = \{x_1, x_2, \ldots, x_n\} \triangleq \mathbf{x}, \quad \text{and} \quad |\mathbf{x}^{[i]}| = n_i,$$

and where ψ is a quantifier-free formula with atomic predicates consisting of polynomial equalities and inequalities of the form

$$g_i\left(\mathbf{x}^{[1]}, \ldots, \mathbf{x}^{[\omega]}\right) \gtreqless 0, \quad i = 1, \ldots, m.$$

Here, g_i is a multivariate polynomial (over $\mathbb{R}$ or $\mathbb{Q}$, as the case may be) of total degree bounded by d. There are a total of m such polynomials. The special case $\omega = 1$ reduces the problem to that of the existential problem for the first-order theory of reals.

If the polynomials of the basic equalities, inequalities, inequations, etc., are over the rationals, then we assume that their coefficients can be stored with at most L bits. Thus the arithmetic complexity can be described in terms of n, n_i, ω, m, and d, and the bit complexity will involve L as well.

Table 29.1.1 highlights a representative set of known bit-complexity results for the decision problem.

QUANTIFIER ELIMINATION PROBLEM

Formally, given a Tarski formula of the form,

$$\Psi(\mathbf{x}^{[0]}) = (\mathcal{Q}_1 \mathbf{x}^{[1]})\ (\mathcal{Q}_2 \mathbf{x}^{[2]})\ \cdots\ (\mathcal{Q}_\omega \mathbf{x}^{[\omega]})\ [\psi(\mathbf{x}^{[0]}, \mathbf{x}^{[1]}, \ldots, \mathbf{x}^{[\omega]})]\,,$$

where ψ is a quantifier-free formula, the ***quantifier elimination problem*** is to construct another quantifier-free formula, $\phi(\mathbf{x}^{[0]})$, such that $\phi(\mathbf{x}^{[0]})$ holds if and

TABLE 29.1.1 Selected time complexity results.

GENERAL OR EXISTENTIAL	TIME COMPLEXITY	SOURCE
General	$L^3(md)^{2^{O(\Sigma n_i)}}$	[Col75]
Existential	$L^{O(1)}(md)^{O(n^2)}$	[GV92]
General	$L^{O(1)}(md)^{(O(\sum n_i))^{4\omega-2}}$	[Gri88]
Existential	$L^{1+o(1)}\ (m)^{(n+1)}\ (d)^{O(n^2)}$	[Can88b, Can93]
General	$(L\log L\log\log L)(md)^{(2^{O(\omega)})\Pi n_i}$	[Ren92a,b,c]
Existential	$(L\log L\log\log L)m\ (m/n)^n\ (d)^{O(n)}$	[BPR94]
General	$(L\log L\log\log L)(m)^{\Pi(n_i+1)}(d)^{\Pi O(n_i)}$	[BPR94]

only if $\Psi(\mathbf{x}^{[0]})$ holds. Such a quantifier-free formula takes the form

$$\phi(\mathbf{x}^{[0]}) \equiv \bigvee_{i=1}^{I} \bigwedge_{j=1}^{J_i} \left(f_{i,j}(\mathbf{x}^{[0]}) \gtreqless 0 \right),$$

where $f_{i,j} \in \mathbb{R}[\mathbf{x}^{[0]}]$ is a multivariate polynomial with real coefficients.

Currently the best bounds are due to Basu et al. [BPR94], and may be summarized as follows:

$$\begin{aligned} I &\leq (m)^{\prod^{(n_i+1)}}(d)^{\prod^{O(n_i)}} \\ J_i &\leq (m)^{\prod_{i>0}{}^{(n_i+1)}}(d)^{\prod_{i>0}{}^{O(n_i)}}. \end{aligned}$$

The total degrees of the polynomials $f_{i,j}(\mathbf{x}^{[0]})$ are bounded by

$$(d)^{\prod_{i>0}{}^{O(n_i)}}.$$

Lower bound results for the quantifier elimination problem can be found in Davenport and Heintz [DH88]. They showed that for every n, there exists a Tarski formula Ψ_n with n quantifiers, of length $O(n)$, and of constant degree, such that any quantifier-free formula ψ_n logically equivalent to Ψ_n must involve polynomials of

$$\text{degree} = 2^{2^{\Omega(n)}} \quad \text{and} \quad \text{length} = 2^{2^{\Omega(n)}}.$$

Note that in the simplest possible case (i.e., $d = 2$ and $n_i = 2$), upper and lower bounds are doubly-exponential and match well. This result, however, does not imply a similar lower bound for the decision problems.

29.2 SEMIALGEBRAIC SETS

Every quantifier-free formula composed of polynomial inequalities and Boolean connectives defines a semialgebraic set. Thus, these semialgebraic sets play an important role in real algebraic geometry.

GLOSSARY

Semialgebraic set: A subset $S \subseteq \mathbb{R}^n$ defined by a set-theoretic expression involving a system of polynomial inequalities

$$S = \bigcup_{i=1}^{I} \bigcap_{j=1}^{J_i} \Big\{ \langle \xi_1, \ldots, \xi_n \rangle \in \mathbb{R}^n \mid \operatorname{sgn}(f_{i,j}(\xi_1, \ldots, \xi_n)) = s_{i,j} \Big\},$$

where the $f_{i,j}$'s are multivariate polynomials over $\mathbb{R}$ and the $s_{i,j}$'s are corresponding sets of signs in $\{-1, 0, +1\}$.

Real algebraic set: A subset $Z \subseteq \mathbb{R}^n$ defined by a system of algebraic equations.

$$Z = \Big\{ \langle \xi_1, \ldots, \xi_n \rangle \in \mathbb{R}^n \mid f_1(\xi_1, \ldots, \xi_n) = \cdots = f_m(\xi_1, \ldots, \xi_n) = 0 \Big\},$$

where the f_i's are multivariate polynomials over $\mathbb{R}$.

Semialgebraic map: A map $\theta : S \to T$, from a semialgebraic set $S \subseteq \mathbb{R}^m$ to a semialgebraic set $T \subseteq \mathbb{R}^n$, such that its graph $\{\langle s, \theta(s) \rangle \in \mathbb{R}^{m+n} : s \in S\}$ is a semialgebraic set in $\mathbb{R}^{m+n}$. Note that projection, being linear, is a semialgebraic map.

TARSKI-SEIDENBERG THEOREM

Equivalently, semialgebraic sets can be defined as

$$S = \Big\{ \langle \xi_1, \ldots, \xi_n \rangle \in \mathbb{R}^n \mid \psi(\xi_1, \ldots, \xi_n) = \mathsf{True} \Big\},$$

where $\psi(x_1, \ldots, x_n)$ is a quantifier-free formula involving n algebraic variables. As a direct corollary of Tarski's theorem on quantifier elimination, we see that extensions of Tarski formulas are also semialgebraic sets.

While real algebraic sets are quite interesting and would be natural objects of study in this context, *they are not closed under projection onto a subspace.* Hence they tend to be unwieldy. However, *semialgebraic sets are closed under projection.* This follows from a more general result: the famous ***Tarski-Seidenberg theorem*** which is an immediate consequence of quantifier elimination, since images are described by formulas involving only existential quantifiers.

THEOREM 29.2.1 *Tarski-Seidenberg Theorem*

Let S be a semialgebraic set in $\mathbb{R}^m$, and let $\theta : \mathbb{R}^m \to \mathbb{R}^n$ be a semialgebraic map. Then $\theta(S)$ is semialgebraic in $\mathbb{R}^n$.

In fact, semialgebraic sets can be defined simply as the smallest class of subsets of $\mathbb{R}^n$ containing real algebraic sets and closed under projection.

GLOSSARY

Connected component of a semialgebraic set: A maximal connected subset of a semialgebraic set. Semialgebraic sets have a finite number of connected components and these are also semialgebraic.

Semialgebraic decomposition of a semialgebraic set S: A finite collection $\mathcal{K}$ of disjoint connected semialgebraic subsets of S whose union is S. The collection of connected components of a semialgebraic set forms a semialgebraic decomposition. Thus, every semialgebraic set admits a semialgebraic decomposition.

Set of sample points for S: A finite number of points meeting every nonempty connected component of S.

Sign assignment: A vector of sign values of a set of polynomials at a point p. More formally, let $\mathcal{F}$ be a set of real multivariate polynomials in n variables. Any point $p = \langle \xi_1, \ldots, \xi_n \rangle \in \mathbb{R}^n$ has a ***sign assignment*** with respect to $\mathcal{F}$ as follows:

$$\mathrm{sgn}_{\mathcal{F}}(p) = \Big\langle \mathrm{sgn}(f(\xi_1, \ldots, \xi_n)) \mid f \in \mathcal{F} \Big\rangle.$$

A sign assignment induces an equivalence relation: Given two points $p, q \in \mathbb{R}^n$, we say

$$p \sim_{\mathcal{F}} q, \quad \text{if and only if} \quad \mathrm{sgn}_{\mathcal{F}}(p) = \mathrm{sgn}_{\mathcal{F}}(q).$$

Sign class of $\mathcal{F}$: An equivalence class in the partition of $\mathbb{R}^n$ defined by the equivalence relation $\sim_{\mathcal{F}}$.

Semialgebraic decomposition for $\mathcal{F}$: A finite collection of disjoint connected semialgebraic subsets $\{C_i\}$ such that each C_i is contained in some semialgebraic sign class of $\mathcal{F}$. That is, the sign of each $f \in \mathcal{F}$ is ***invariant*** in each C_i. The collection of connected components of the sign-invariant sets for $\mathcal{F}$ forms a semialgebraic decomposition for $\mathcal{F}$.

Cell decomposition for $\mathcal{F}$: A semialgebraic decomposition for $\mathcal{F}$ into finitely many disjoint semialgebraic subsets $\{C_i\}$ called ***cells***, such that each cell C_i is homeomorphic to $\mathbb{R}^{\delta(i)}$, $0 \leq \delta(i) \leq n$. $\delta(i)$ is called the ***dimension of the cell*** C_i, and C_i is called a ***$\delta(i)$-cell.***

Cellular decomposition for $\mathcal{F}$: A cell decomposition for $\mathcal{F}$ such that the closure $\overline{C_i}$ of each cell C_i is a union of cells C_j: $\overline{C_i} = \cup_j C_j$.

CONNECTED COMPONENTS OF SEMIALGEBRAIC SETS

A consequence of the Milnor-Thom result [Mil64, Tho65] gives a bound for the number (the zeroth ***Betti number***, $B_0(S)$) of connected components of a basic semialgebraic set S: the bound is polynomial in the number m and degree d of the polynomials defining S and singly-exponential in the number of variables, n. The current best bound for $B_0(S)$ is due to Pollack and Roy [PR93]: $B_0(S) = (O(md/n))^n$.

A key problem in computational real algebraic geometry is to compute at least one point in each connected component of each nonempty sign assignment. An elegant solution to this problem is obtained by Collins's ***cylindrical algebraic decomposition*** (CAD), which is, in fact, a cell decomposition. A related question is to provide a finitary representation for these sample points, e.g., each coordinate of the sample point may be a *real algebraic number.*

Currently, the best algorithm computing a finite set of points of bounded size that intersects *every connected component* of each nonempty sign condition is due to Basu et al. [BPR] and has an arithmetic time-complexity of $m(m/n)^n d^{O(n)}$.

29.3 REAL ALGEBRAIC NUMBERS

Real algebraic numbers are real roots of rational univariate polynomials and provide finitary representation for some of the basic objects (e.g., sample points). Furthermore, we note that (1) real algebraic numbers have effective finitary representation, (2) field operations and polynomial evaluation on real algebraic numbers are efficiently (polynomially) computable, and (3) conversions among various representations of real algebraic numbers are efficiently (polynomially) computable. The key machinery used in describing and manipulating real algebraic numbers relies upon techniques based on the Sturm-Sylvester theorem, Thom's lemma, resultant construction, and various bounds for real root separation.

GLOSSARY

Real algebraic number: A real root α of a univariate polynomial $p(t) \in \mathbb{Z}[t]$ with integer coefficients.

Polynomial for α: A univariate polynomial p such that α is a real root of p.

Minimal polynomial of α: A univariate polynomial p of minimal degree defining α as above.

Degree of a nonzero real algebraic number: The degree of its minimal polynomial. By convention, the degree of the 0 polynomial is $-\infty$.

OPERATIONS ON REAL ALGEBRAIC NUMBERS

Note that if α and β are real algebraic numbers, then so are $-\alpha$, α^{-1} (assuming $\alpha \neq 0$), $\alpha+\beta$, and $\alpha \cdot \beta$. These facts can be constructively proved using the algebraic properties of a resultant construction.

THEOREM 29.3.1

The real algebraic numbers form a field.

A real algebraic number α can be represented by a polynomial for α and a component that identifies the root. There are essentially three types of information that may be used for this identification: *order* (where we assume the real roots are indexed from left to right), *sign* (by a vector of signs), or *interval* (an interval that contains exactly one root).

A classical technique due to Sturm and Sylvester shows how to compute the number of real roots of a univariate polynomial $p(t)$ in an interval $[a, b]$. One important use of this classical theorem is to compute a sequence of relatively small (nonoverlapping) intervals that isolate the real roots of p.

GLOSSARY

Sturm sequence of a pair of polynomials $p(t)$ and $q(t) \in \mathbb{R}[t]$:

$$\overline{\text{STURM}}(p, q) = \left\langle \hat{r}_0(t), \hat{r}_1(t), \ldots, \hat{r}_s(t) \right\rangle,$$

where

$$\begin{aligned}
\hat{r}_0(t) &= p(t) \\
\hat{r}_1(t) &= q(t) \\
&\vdots \\
\hat{r}_{i-1}(t) &= \hat{q}_i(t)\,\hat{r}_i(t) - \hat{r}_{i+1}(t), \qquad \deg(\hat{r}_{i+1}) < \deg(\hat{r}_i) \\
&\vdots \\
\hat{r}_{s-1}(t) &= \hat{q}_s(t)\,\hat{r}_s(t).
\end{aligned}$$

Number of variations in sign of a finite sequence $\overline{c}$ of real numbers: Number of times the entries change sign when scanned sequentially from left to right; denoted $\mathrm{Var}(\overline{c})$.

For a vector of polynomials $\overline{P} = \langle p_1(t), \ldots, p_m(t)\rangle$ and a real number a:

$$\mathrm{Var}_a(\overline{P}) \;=\; \mathrm{Var}(\overline{P}(a)) \;=\; \mathrm{Var}(\langle p_1(a), \ldots, p_m(a)\rangle).$$

Formal derivative: $p'(t) = D(p(t))$, where $D\colon \mathbb{R}[t] \to \mathbb{R}[t]$ is the (formal) derivative map, taking t^n to nt^{n-1} and $a \in \mathbb{R}$ (a constant) to 0.

STURM-SYLVESTER THEOREM

THEOREM 29.3.2 *Sturm-Sylvester Theorem*

Let $p(t)$ and $q(t) \in \mathbb{R}[t]$ be two real univariate polynomials. Then, for any interval $[a, b] \subseteq \mathbb{R} \cup \{\pm\infty\}$ (where $a < b$):

$$\mathrm{Var}\Big[\overline{P}\Big]_a^b = c_p\Big[q > 0\Big]_a^b - c_p\Big[q < 0\Big]_a^b,$$

where

$$\begin{aligned}
\overline{P} &\triangleq \overline{\mathrm{STURM}}(p, p'q), \\
\mathrm{Var}\Big[\overline{P}\Big]_a^b &\triangleq \mathrm{Var}_a(\overline{P}) - \mathrm{Var}_b(\overline{P}),
\end{aligned}$$

and $c_p[\mathcal{P}]_a^b$ counts the number of distinct real roots (without counting multiplicity) of p in the interval (a, b) at which the predicate $\mathcal{P}$ holds.

Note that if we take $S_p \triangleq \overline{\mathrm{STURM}}(p, p')$ (i.e., $q = 1$) then

$$\begin{aligned}
\mathrm{Var}\Big[S_p\Big]_a^b &= c_p\Big[\mathsf{True}\Big]_a^b - c_p\Big[\mathsf{False}\Big]_a^b \\
&= \#\text{ of distinct real roots of } p \text{ in } (a, b).
\end{aligned}$$

COROLLARY 29.3.3

Let $p(t)$ and $q(t)$ be two polynomials with coefficients in a real closed field K. For any interval $[a, b]$ as before, we have

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & -1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_p\Big[q = 0\Big]_a^b \\ c_p\Big[q > 0\Big]_a^b \\ c_p\Big[q < 0\Big]_a^b \end{bmatrix} = \begin{bmatrix} \mathrm{Var}\Big[\overline{\mathrm{STURM}}(p, p')\Big]_a^b \\ \mathrm{Var}\Big[\overline{\mathrm{STURM}}(p, p'q)\Big]_a^b \\ \mathrm{Var}\Big[\overline{\mathrm{STURM}}(p, p'q^2)\Big]_a^b \end{bmatrix}.$$

These identities as well as some related algorithmic results (the so-called BKR-algorithm) are based on results of Ben-Or et al. [BKR86] and their extensions by others. Using this identity, it is a fairly simple matter to decide the sign conditions of a single univariate polynomial q at the roots of a univariate polynomial p. It is possible to generalize this idea to decide the sign conditions of a sequence of univariate polynomials $q_0(t)$, $q_1(t)$, ..., $q_n(t)$ at the roots of a single polynomial $p(t)$ and hence give an efficient (both sequential and parallel) algorithm for the decision problem for Tarski sentences involving univariate polynomials. Further applications in the context of general decision problems are described later.

GLOSSARY

Fourier sequence of a real univariate polynomial $p(t)$ of degree n:

$$\overline{\mathrm{FOURIER}}(p) = \Big\langle p^{(0)}(t) = p(t),\ p^{(1)}(t) = p'(t),\ \ldots, p^{(n)}(t)\ \Big\rangle,$$

where $p^{(i)}$ is the ith derivative of p with respect to t.

Sign-invariant region of $\mathbb{R}$ determined by a sign sequence $\overline{s}$ with respect to $\overline{\mathrm{FOURIER}}(p)$: The region $R(\overline{s})$ with the property that $\xi \in R(\overline{s})$ if and only if $\mathrm{sgn}(p^{(i)}(\xi)) = s_i$.

THOM'S LEMMA

LEMMA 29.3.4 *Thom's Lemma*

Every nonempty sign-invariant region $R(\overline{s})$ (determined by a sign sequence $\overline{s}$ with respect to $\overline{\mathrm{FOURIER}}(p)$) must be connected, i.e., consists of a single interval.

Let $\mathrm{sgn}_\xi(\overline{\mathrm{FOURIER}}(p))$ be the sign sequence obtained by evaluating the polynomials of $\overline{\mathrm{FOURIER}}(p)$ at ξ. Then as an immediate corollary of Thom's lemma, we have:

COROLLARY 29.3.5

Let ξ and ζ be two real roots of a real univariate polynomial $p(t)$ of positive degree $n > 0$. Then $\xi = \zeta$, if

$$\mathrm{sgn}_\xi(\overline{\mathrm{FOURIER}}(p')) = \mathrm{sgn}_\zeta(\overline{\mathrm{FOURIER}}(p')).$$

REPRESENTATION OF REAL ALGEBRAIC NUMBERS

Let $p(t)$ be a univariate polynomial of degree d with integer coefficients. Assume that the distinct real roots of $p(t)$ have been enumerated as follows:

$$\alpha_1 < \alpha_2 < \cdots < \alpha_{j-1} < \alpha_j = \alpha < \alpha_{j+1} < \cdots < \alpha_l,$$

where $l \leq d = \deg(p)$. Then we can represent any of its roots uniquely and in a finitary manner.

GLOSSARY

Order representation of an algebraic number: A pair consisting of its polynomial p and its index j in the monotone sequence enumerating the real roots of p: $\langle\alpha\rangle_o = \langle p, j\rangle$. *Example*: $\langle\sqrt{2}+\sqrt{3}\rangle_o = \langle x^4 - 10x^2 + 1, 4\rangle$.

Sign representation of an algebraic number: A pair consisting of its polynomial p and a sign sequence $\overline{s}$ representing the signs of its Fourier sequence evaluated at the root: $\langle\alpha\rangle_s = \langle p, \overline{s} = \mathrm{sgn}_\alpha(\overline{\mathrm{FOURIER}(p')})\rangle$. *Example*: $\langle\sqrt{2}+\sqrt{3}\rangle_s = \langle x^4 - 10x^2 + 1, (+1, +1, +1)\rangle$. The validity of this representation follows easily from Thom's Lemma.

Interval representation of an algebraic number: A triple consisting of its polynomial p and the two endpoints of an isolating interval, (l, r) $(l, r \in \mathbb{Q}, l < r)$, containing only α: $\langle\alpha\rangle_i = \langle p, l, r\rangle$. *Example*: $\langle\sqrt{2}+\sqrt{3}\rangle_i = \langle x^4 - 10x^2 + 1, 3, 7/2\rangle$.

29.4 UNIVARIATE DECOMPOSITION

In the one-dimensional case, a semialgebraic set is the union of finitely many intervals whose endpoints are real algebraic numbers. For instance, given a set of univariate defining polynomials:

$$\mathcal{F} = \Big\{ f_i(x) \in \mathbb{Q}[x] \quad | \quad i = 1, \ldots, m \Big\},$$

we may enumerate all the real roots of the f_i's (i.e., the real roots of the single polynomial $F = \prod f_i$) as

$$-\infty < \xi_1 < \xi_2 < \cdots < \xi_{i-1} < \xi_i < \xi_{i+1} < \cdots < \xi_s < +\infty,$$

and consider the following finite set $\mathcal{K}$ of elementary intervals defined by these roots:

$$\begin{aligned}&[-\infty, \xi_1),\ [\xi_1, \xi_1],\ (\xi_1, \xi_2),\ \ldots,\\ &\quad (\xi_{i-1}, \xi_i),\ [\xi_i, \xi_i],\ (\xi_i, \xi_{i+1}),\ \ldots,\ [\xi_s, \xi_s],\ (\xi_s, +\infty].\end{aligned}$$

Note that $\mathcal{K}$ is, in fact, a cellular decomposition for $\mathcal{F}$. Any semialgebraic set S defined by $\mathcal{F}$ is simply the union of a subset of elementary intervals in $\mathcal{K}$. Furthermore, for each interval $C \in \mathcal{K}$, we can compute a sample point α_C as follows:

$$\alpha_C = \begin{cases} \xi_1 - 1, & \text{if } C = [-\infty, \xi_1); \\ \xi_i, & \text{if } C = [\xi_i, \xi_i]; \\ (\xi_i + \xi_{i+1})/2, & \text{if } C = (\xi_i, \xi_{i+1}); \\ \xi_s + 1, & \text{if } C = (\xi_s, +\infty]. \end{cases}$$

Now, given a first-order formula involving a single variable, its validity can be checked by evaluating the associated univariate polynomials at the sample points. Using the algorithms for representing and manipulating real algebraic numbers, we see that the bit complexity of the decision algorithm is bounded by $(Lmd)^{O(1)}$. The resulting cellular decomposition has no more than $2md + 1$ cells.

Using variants of the theorem due to Ben-Or et al. [BKR86], Thom's lemma, and some results on parallel computations in linear algebra, one can show that this univariate decision problem is "well-parallelizable," i.e., the problem is solvable by uniform circuits of bounded depth and polynomially many "gates" (simple processors).

29.5 MULTIVARIATE DECOMPOSITION

A straightforward generalization of the standard univariate decomposition to higher dimensions is provided by Collins's cylindrical algebraic decomposition. In order to represent a semialgebraic set $S \subseteq \mathbb{R}^n$, we may assume recursively that we can construct a cell decomposition of its projection $\pi(S) \subseteq \mathbb{R}^{n-1}$ (also a semialgebraic set), and then decompose S as a union of the ***sectors*** and ***sections*** in the cylinders above each cell of the projection, $\pi(S)$. This also leads to a cell decomposition of S. One can further assign an algebraic sample point in each cell of S recursively in a straightforward manner.

If $\mathcal{F}$ is a set of polynomials defining the semialgebraic set $S \subseteq \mathbb{R}^n$, then at no additional cost, we may in fact compute a cell decomposition for $\mathcal{F}$ using the procedure described above. Such a decomposition leads to a *cylindrical algebraic decomposition* for $\mathcal{F}$.

GLOSSARY

Cylindrical algebraic decomposition (CAD): A recursively defined cell decomposition of $\mathbb{R}^n$ for $\mathcal{F}$. The decomposition is a cellular decomposition if the set of defining polynomials $\mathcal{F}$ satisfies certain nondegeneracy conditions.

In the recursive definition, the cells of n-dimensional CAD are constructed from an $(n-1)$-dimensional CAD: Every $(n-1)$-dimensional CAD cell C' has the property that the distinct real roots of $\mathcal{F}$ over C' vary continuously as a function of the points of C'.

Moreover, the following quantities remain invariant over a $(n-1)$-dimensional cell: (1) the total number of complex roots of each polynomial of $\mathcal{F}$; (2) the number of distinct complex roots of each polynomial of $\mathcal{F}$; and (3) the total number of common complex roots of every distinct pair of polynomials of $\mathcal{F}$.

These conditions can be expressed by a set $\Phi(\mathcal{F})$ of at most $O(md)^2$ polynomials in $n-1$ variables, obtained by considering *principal subresultant coefficients* (PSC's). Thus, they correspond roughly to *resultants* and *discriminants*, and ensure that the polynomials of $\mathcal{F}$ do not intersect or "fold" in a cylinder over an $(n-1)$-dimensional cell. The polynomials in $\Phi(\mathcal{F})$ are each of degree no more than d^2.

More formally, an $\mathcal{F}$-sign-invariant cylindrical algebraic decomposition of $\mathbb{R}^n$ is:

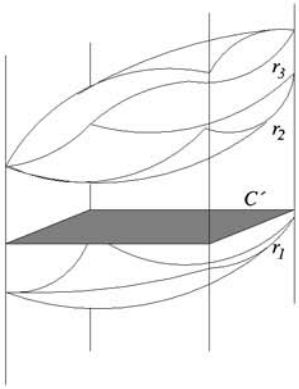


FIGURE 29.5.1
Sections and sectors "slicing" the cylinder over a lower dimensional cell.

- BASE CASE: $n = 1$. A univariate cellular decomposition of $\mathbb{R}^1$ as in the previous section.
- INDUCTIVE CASE: $n > 1$. Let $\mathcal{K}'$ be a $\Phi(\mathcal{F})$-sign-invariant CAD of $\mathbb{R}^{n-1}$. For each cell $C' \in \mathcal{K}'$, define an ***auxiliary polynomial*** $g_{C'}(x_1, \ldots, x_{n-1}, x_n)$ as the product of those polynomials of $\mathcal{F}$ that do not vanish over the $(n-1)$-dimensional cell, C'. The real roots of the auxiliary polynomial g'_C over C' give rise to a finite number (perhaps zero) of semialgebraic continuous functions, which partition the cylinder $C' \times (\mathbb{R} \cup \{\pm\infty\})$ into finitely many $\mathcal{F}$-sign-invariant "slices." The auxiliary polynomials are of degree no larger than md.
  Assume that the polynomial $g_{C'}(p', x_n)$ has l distinct real roots for each $p' \in C'$: $r_1(p'), r_2(p'), \ldots, r_l(p')$, each r_i being a continuous function of p'. The following sectors and sections are cylindrical over C' (see Figure 29.5.1):

$$\begin{aligned}
C_0^* &= \Big\{\langle p', x_n\rangle \mid p' \in C' \wedge x_n \in [-\infty, r_1(p'))\Big\},\\
C_1 &= \Big\{\langle p', x_n\rangle \mid p' \in C' \wedge x_n \in [r_1(p'), r_1(p')]\Big\},\\
C_1^* &= \Big\{\langle p', x_n\rangle \mid p' \in C' \wedge x_n \in (r_1(p'), r_2(p'))\Big\},\\
&\vdots\\
C_l^* &= \Big\{\langle p', x_n\rangle \mid p' \in C' \wedge x_n \in (r_l(p'), +\infty]\Big\}.
\end{aligned}$$

  The n-dimensional CAD is thus the union of all the sections and sectors computed over the cells of the $(n-1)$-dimensional CAD.

A straightforward recursive algorithm to compute a CAD follows from the above description.

CYLINDRICAL ALGEBRAIC DECOMPOSITION

If we assume that the dimension n is a fixed constant, then the preceding cylindrical algebraic decomposition algorithm is polynomial in $m = |\mathcal{F}|$ and $d = \deg(\mathcal{F})$. However, the algorithm can be easily seen to be doubly-exponential in n as the number of polynomials produced at the lowest dimension is $(md)^{2^{O(n)}}$, each of degree no larger than $d^{2^{O(n)}}$. The number of cells produced by the algorithm is also

doubly-exponential. This bound can be seen to be tight by a result due to Davenport and Heintz [DH88], and is related to their lower bound for the quantifier elimination problem (Section 29.1).

CONSTRUCTING SAMPLE POINTS

Cylindrical algebraic decomposition provides a sample point in every sign-invariant connected component for $\mathcal{F}$. However, the total number of sample points generated is doubly-exponential, while the number of connected components of all sign conditions is only singly-exponential. In order to avoid this high complexity (both algebraic and combinatorial) of a CAD, many recent techniques for constructing sample points use a single projection to a line instead of a sequence of cascading projections. For instance, if one chooses a height function carefully then one can easily enumerate its critical points and then associate at least two such critical points to every connected component of the semialgebraic set. From these critical points, it will be possible to create at least one sample point per connected component. Using Bézout's bound, it is seen that only a singly-exponential number of sample points is created, thus improving the complexity of the underlying algorithms.

However, in order to arrive at the preceding conclusion using critical points, one requires certain genericity conditions that can be achieved by symbolically deforming the underlying semialgebraic sets. These infinitesimal deformations can be handled by extending the underlying field to a field of *Puiseux series*. Many of the significant complexity improvements based on these techniques have been due to a careful choice of the symbolic perturbation schemes which results in keeping the number of perturbation variables small.

29.6 ALGORITHMIC APPROACHES

COLLINS'S APPROACH

The decision problem for the first-order theory of reals can be solved easily using a cylindrical algebraic decomposition. First consider the existential problem for a sentence with only existential quantifiers,

$$(\exists\, \mathbf{x}^{[0]})\ [\psi(\mathbf{x}^{[0]})].$$

This sentence is true if and only if there is a $q \in C$, a sample point in the cell C,

$$q \;=\; \alpha^{[0]} = \langle \alpha_1, \ldots, \alpha_n \rangle \;\in\; \mathbb{R}^n,$$

such that $\psi(\alpha^{[0]})$ is true. Thus we see that the decision problem for the purely existential sentence can be solved by simply evaluating the matrix ψ over the finitely many sample points in the associated CAD. This also implies that the existential quantifiers could be replaced by finitely many disjunctions ranging over all the sample points. Note that the same arguments hold for any semialgebraic decomposition with at least one sample point per sign-invariant connected component.

In the general case, one can describe the decision procedure by means of a search process that proceeds *only on* the coordinates of the sample points in the

cylindrical algebraic decomposition. This follows because a sample point in a cell acts as a representative for any point in the cell as far as the sign conditions are concerned.

Consider a Tarski sentence

$$(\mathcal{Q}_1 \mathbf{x}^{[1]})\ (\mathcal{Q}_2 \mathbf{x}^{[2]})\ \cdots\ (\mathcal{Q}_\omega \mathbf{x}^{[\omega]})\ [\psi(\mathbf{x}^{[1]}, \ldots, \mathbf{x}^{[\omega]}],$$

with $\mathcal{F}$ the set of polynomials appearing in the matrix ψ. Let $\mathcal{K}$ be a cylindrical algebraic decomposition of $\mathbb{R}^n$ for $\mathcal{F}$. Since the cylindrical algebraic decomposition produces a sequence of decompositions:

$$\mathcal{K}_1 \text{ of } \mathbb{R}^1,\ \ \mathcal{K}_2 \text{ of } \mathbb{R}^2,\ \ \ldots,\ \ \mathcal{K}_n \text{ of } \mathbb{R}^n,$$

such that the each cell $C_{i-1,j}$ of $\mathcal{K}_i$ is cylindrical over some cell C_{i-1} of $\mathcal{K}_{i-1}$, the search progresses by first finding cells C_1 of $\mathcal{K}_1$ such that

$$(\mathcal{Q}_2 x_2)\ \cdots\ (\mathcal{Q}_n x_n)\ [\psi(\alpha_{C_1}, x_2, \ldots, x_n)] = \mathsf{True}.$$

For each C_1, the search continues over cells C_{12} of $\mathcal{K}_2$ cylindrical over C_1 such that

$$(\mathcal{Q}_3 x_3)\ \cdots\ (\mathcal{Q}_n x_n)\ [\psi(\alpha_{C_1}, \alpha_{C_{12}}, x_3, \ldots, x_n)] = \mathsf{True},$$

etc. Finally, at the bottom level the truth properties of the matrix ψ are determined by evaluating at all the coordinates of the sample points.

This produces a tree structure, where each node at the $(i-1)$-th level corresponds to a cell $C_{i-1} \in \mathcal{K}_{i-1}$ and its children correspond to the cells $C_{i-1,j} \in \mathcal{K}_i$ that are cylindrical over C_{i-1}. The leaves of the tree correspond to the cells of the final decomposition $\mathcal{K} = \mathcal{K}_n$. Because we only have finitely many sample points, the universal quantifiers can be replaced by finitely many conjunctions and the existential quantifiers by disjunctions. Thus, we label every node at the $(i-1)$-th level "AND" (respectively, "OR") if $\mathcal{Q}_i$ is a universal quantifier $\forall$ (respectively, $\exists$) to produce a so-called AND-OR tree. The truth of the Tarski sentence is thus determined by simply evaluating this AND-OR tree.

A quantifier elimination algorithm can be devised by a similar reasoning and a slight modification of the CAD algorithm described earlier.

NEW APPROACHES USING CRITICAL POINTS

In order to avoid the cascading projections inherent in Collins's algorithm, the new approaches employ a single projection to a one-dimensional set by using critical points in a manner described earlier. As before, we start with a sentence with only existential quantifiers,

$$(\exists\, \mathbf{x}^{[0]})\ [\psi(\mathbf{x}^{[0]})].$$

Let $\mathcal{F} = \{f_1, \ldots, f_m\}$ be the set of polynomials appearing in the matrix ψ.

Under certain genericity conditions, it is possible to produce a set of sample points such that every sign-invariant connected component of the decomposition induced by $\mathcal{F}$ contains at least one such point. Furthermore, these sample points are described by a set of univariate polynomial sequences, where each sequence is of the form

$$p(t), q_0(t), q_1(t), \ldots, q_n(t),$$

and encodes a sample point $\langle \frac{q_1(\alpha)}{q_0(\alpha)}, \ldots, \frac{q_n(\alpha)}{q_0(\alpha)} \rangle$. Here α is a root of p. Now the decision problem for the existential theory can be solved by deciding the sign conditions of the sequence of univariate polynomials

$$f_1(q_1/q_0, \ldots, q_n/q_0), \ \ldots, \ f_m(q_1/q_0, \ldots, q_n/q_0),$$

at the roots of the univariate polynomial $p(t)$. Note that we have now reduced a multivariate problem to a univariate problem and can solve this by the BKR approach.

In order to keep the complexity reasonably small, one needs to ensure that the number of such sequences is small and that these polynomials are of low degree. Assuming that the polynomials in $\mathcal{F}$ are in general position, one can achieve this and compute the polynomials p and q_i (for example, by the u-resultant method in Renegar's algorithm).

If the genericity conditions are violated, one needs to symbolically deform the polynomials and carry out the computations on these polynomials with additional perturbation parameters. The Basu-Pollack-Roy (BPR) algorithm differs from Renegar's algorithm primarily in the manner in which these perturbations are made so that their effect on the algorithmic complexity is controlled.

Next consider an existential Tarski formula of the form

$$(\exists \, \mathbf{x}^{[0]}) \ [\psi(\mathbf{y}, \mathbf{x}^{[0]})],$$

where $\mathbf{y}$ represents the free variables. If we carry out the same computation as before over the ambient field $\mathbb{R}(\mathbf{y})$, we get a set of *parameterized* univariate polynomial sequences, each of the form

$$p(\mathbf{y}, t), q_0(\mathbf{y}, t), q_1(\mathbf{y}, t), \ldots, q_n(\mathbf{y}, t).$$

For a fixed value of $\mathbf{y}$, say $\bar{y}$, the polynomials

$$p(\bar{y}, t), q_0(\bar{y}, t), q_1(\bar{y}, t), \ldots, q_n(\bar{y}, t)$$

can then be used as before to decide the truth or falsity of the sentence

$$(\exists \, \mathbf{x}^{[0]}) \ [\psi(\bar{y}, \mathbf{x}^{[0]})].$$

Also, one may observe that the *parameter space* $\mathbf{y}$ can be partitioned into semialgebraic sets so that all the necessary information can be obtained by computing at sample values $\bar{y}$.

This process can be extended to ω blocks of quantifiers, by replacing each block of variables by a finite number of cases, each involving only one new variable; the last step uses a CAD method for these ω-many variables.

29.7 APPLICATIONS

Computational real algebraic geometry finds applications in robotics, vision, computer-aided design, geometric theorem proving, and other fields. Important problems in robotics include the kinematic modeling, the inverse kinematic solution, the computation of the workspace and workspace singularities, and the planning of an

obstacle-avoiding motion of a robot in a cluttered environment—all arising from the algebro-geometric nature of robot kinematics. In solid modeling, graphics, and vision, almost all applications involve the description of surfaces, the generation of various auxiliary surfaces such as blending and smoothing surfaces, the classification of various algebraic surfaces, the algebraic or geometric invariants associated with a surface, the effect of various affine or projective transformations of a surface, the description of surface boundaries, and so on.

To give examples of the nature of the solutions demanded by various applications, we discuss a few representative problems from robotics, engineering, and computer science.

ROBOT MOTION PLANNING

Given the initial and desired configurations of a robot (composed of rigid subparts) and a set of obstacles, find a collision-free continuous motion of the robot from the initial configuration to the final configuration.

The algorithm proceeds in several steps. The first step translates the problem to ***configuration space***, a parameter space modeled as a low-dimensional algebraic manifold (assuming that the obstacles and the robot subparts are bounded by piecewise algebraic surfaces). The second step computes the set of configurations that avoid collisions and produces a semialgebraic description of this so-called "free space" (subspaces of the configuration space). Since the initial and final configurations correspond to two points in the configuration space, we simply have to test whether they lie in the same connected component of the free space. If so, they can be connected by a piecewise algebraic path. Such a path gives rise to an obstacle-avoiding motion of the robot(s). This path planning process can be carried out using Collins's CAD [SS83], yielding an algorithm with doubly-exponential time complexity (Theorem 40.1.1). A singly-exponential time complexity algorithm (the *roadmap algorithm*) has been devised by Canny [Can88a] (Theorem 40.1.2). The main idea of Canny's algorithm is to determine a one-dimensional connected subset (called the "roadmap") of each connected component of the free space. Once these roadmaps are available, they can be used to link up two points in the same connected component. The main geometric idea is to construct roadmaps starting from the critical sets of some projection function. The basic roadmap algorithm has been improved and extended by several researchers over the last decade (Heintz et al., Gournay and Risler, Grigor'ev and Vorobjov, and Canny).

OFFSET SURFACE CONSTRUCTION IN SOLID MODELING

Given a polynomial $f(x, y, z)$, whose zeros define an algebraic surface in three-dimensional space, compute the envelope of a family of spheres of radius r whose centers lie on the surface f. Such a surface is called a (two-sided) ***offset surface*** of f.

Let $p = \langle x, y, z \rangle$ be a point on the offset surface and $q = \langle u, v, w \rangle$ be a ***footprint*** of p on f; that is, q is the point at which a normal from p to f meets f. Let $\vec{t_1} = \langle t_{1,1}, t_{1,2}, t_{1,3} \rangle$ and $\vec{t_2} = \langle t_{2,1}, t_{2,2}, t_{2,3} \rangle$ be two linearly independent tangent

vectors to f at the point q. Then, we see that the system of polynomial equations

$$\begin{aligned}(x-u)^2+(y-v)^2+(z-w)^2-r^2 &= 0,\\ f(u,v,w) &= 0,\\ (x-u)t_{1,1}+(y-v)t_{1,2}+(z-w)t_{1,3} &= 0,\\ (x-u)t_{2,1}+(y-v)t_{2,2}+(z-w)t_{2,3} &= 0,\end{aligned}$$

describes a surface in the (x, y, z, u, v, w) six-dimensional space, which, when projected into the three-dimensional space with coordinates (x, y, z), gives the offset surface in an implicit form. The offset surface is computed by simply eliminating the variables u, v, w from the preceding set of equations.

This approach (the ***envelope method***) of computing the offset surface has several problematic features: the method does not deal with self-intersection in a clean way and, sometimes, generates additional points not on the offset surface. For a discussion of these and several other related problems in solid modeling, see [Hof89] and Chapter 47 of this Handbook.

GEOMETRIC THEOREM PROVING

Given a geometric statement consisting of a finite set of hypotheses and a conclusion,

$$\begin{aligned}\text{Hypotheses} &: f_1(x_1,\ldots,x_n)=0,\ldots,f_r(x_1,\ldots,x_n)=0\\ \text{Conclusion} &: g(x_1,\ldots,x_n)=0\end{aligned}$$

decide whether the conclusion $g = 0$ is a consequence of the hypotheses $((f_1 = 0) \wedge \cdots \wedge (f_r = 0))$.

Thus we need to determine whether the following universally quantified first-order sentence holds:

$$\Big(\forall\, x_1,\ldots,x_n\Big)\left[\Big((f_1=0)\wedge\cdots\wedge(f_r=0)\Big)\ \Rightarrow\ g=0\right].$$

One way to solve the problem is by first translating it into the form: decide if the following existentially quantified first-order sentence is unsatisfiable:

$$\Big(\exists\, x_1,\ldots,x_n,z\Big)\left[(f_1=0)\wedge\cdots\wedge(f_r=0)\wedge(gz-1)=0\right].$$

When the underlying domain is assumed to be the field of real numbers, then we may simply check whether the following multivariate polynomial (in $x_1, \ldots, x_n, z$) has no real root:

$$f_1^2+\cdots+f_r^2+(gz-1)^2.$$

If, on the other hand, the underlying domain is assumed to be the field of complex numbers (an algebraically closed field), then other tools from computational algebra are used (e.g., techniques based on Hilbert's Nullstellensatz). In the general setting, some techniques based on Ritt-Wu characteristic sets have proven very powerful. See [Cho88].

For another approach to geometric theorem proving, see Section 48.4.

29.8 SOURCES AND RELATED MATERIAL

SURVEYS

All results not given an explicit reference may be traced in these surveys.

[Mis93]: A textbook for algorithmic algebra covering Gröbner bases, characteristic sets, resultants, and real algebra. Chapter 8 gives many details of the classical results in computational real algebra.

[CJ95]: An anthology of key papers in computational real algebra and real algebraic geometry. Contains reprints of the following papers cited in this chapter: [BPR, Col75, Ren91, Tar51].

[AB88]: A special issue of the *J. Symbolic Comput.* on computational real algebraic geometry. Contains several papers ([DH88, Gri88, GV88] cited here) addressing many key research problems in this area.

[BR90]: A very accessible and self-contained textbook on real algebra and real algebraic geometry.

[BCR87]: A self-contained textbook in French on real algebra and real algebraic geometry. An English edition is under preparation.

[HRR91]: A survey of many classical and recent results in computational real algebra.

[Cha94]: A survey of the connections among computational geometry, computational algebra, and computational real algebraic geometry.

[Tar51]: Primary reference for Tarski's classical result on the decidability of elementary algebra.

[Col75]: Collins's work improving the complexity of Tarski's solution for the decision problem [Tar51]. Also, introduces the concept of cylindrical algebraic decomposition (CAD).

[Ren91]: A survey of some recent results, improving the complexity of the decision problem and quantifier elimination problem for the first-order theory of reals. This is mostly a summary of the results first given in a sequence of papers by Renegar [Ren92a,b,c].

[Lat91]: A comprehensive textbook covering various aspects of robot motion planning problems and different solution techniques. Chapter 5 includes a description of the connection between the motion planning problem and computational real algebraic geometry.

[SS83]: A classic paper in robotics showing the connection between the robot motion planning problem and the connectivity of semialgebraic sets using CAD. Contains several improved algorithmic results in computational real algebra.

[Can88a]: Gives a singly-exponential time algorithm for the robot motion planning problem and provides complexity improvement for many key problems in computational real algebra.

[Hof89]: A comprehensive textbook covering various computational algebraic tech-

niques with applications to solid modeling. Contains a very readable description of Gröbner bases algorithms.

[Cho88]: A monograph on geometric theorem proving using Ritt-Wu characteristic sets. Includes computer-generated proofs of many classical geometric theorems.

RELATED CHAPTERS

Chapter 40: Algorithmic motion planning
Chapter 41: Robotics
Chapter 47: Solid modeling
Chapter 48: Geometric applications of the Grassmann-Cayley algebra

REFERENCES

[AB88] D. Arnon and B. Buchberger, editors, *Algorithms in Real Algebraic Geometry.* Special Issue: *J. Symbolic Comput.*, 5(1-2), 1988.

[BPR94] S. Basu, R. Pollack, and M.-F. Roy. On the combinatorial and algebraic complexity of quantifier elimination (Extended Abstract). In *Proc. 35th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 632–641, 1994.

[BPR] S. Basu, R. Pollack, and M.-F. Roy. A new algorithm to find a point in every cell defined by a family of polynomials. In B. Caviness and J. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, *Texts Monographs Symbol. Comput.*, Springer-Verlag, Vienna/New York, to appear.

[BR90] R. Benedetti and J.-J. Risler. *Real Algebraic and Semi-Algebraic Sets.* Hermann, Éditeurs des Sciences et des Arts, Paris, 1990.

[BKR86] M. Ben-Or, D. Kozen, and J. Reif. The complexity of elementary algebra and geometry. *J. Comput. Syst. Sci.*, 32:251–264, 1986.

[BCR87] J. Bochnak, M. Coste, and M.-F. Roy. *Géométrie Algébrique Réelle.* Springer-Verlag, New York, 1987.

[Can88a] J.F. Canny. *The Complexity of Robot Motion Planning.* Ph.D. Thesis, MIT, Cambridge, 1988.

[Can88b] J.F. Canny. Some algebraic and geometric computations in PSPACE. In *Proc. 20th Annu. ACM Sympos. Theory Comput.*, pages 460–467, 1988.

[Can93] J.F. Canny. Improved algorithms for sign determination and existential quantifier elimination. *Comput. J.*, 36:409–418, 1993.

[Cha94] B. Chazelle. Computational geometry: A retrospective. In *Proc. 26th Annu. ACM Sympos. Theory Comput.*, pages 75–94, 1994.

[Cho88] S.C. Chou. *Mechanical Geometry Theorem Proving.* Reidel, Dordrecht, 1988.

[CJ95] B.F. Caviness and J.R. Johnson, editors. *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Springer-Verlag, Vienna, 1995.

[Col75] G. Collins. Quantifier elimination for real closed fields by Cylindrical Algebraic Decomposition. *Second GI Conf. on Automata Theory Formal Lang.*, volume 33 of *Lecture Notes in Comput. Sci.*, pages 134–183. Springer-Verlag, Berlin, 1975. Also in [CJ95].

[DH88] J.H. Davenport and J. Heintz. Real quantifier elimination is doubly exponential. *J. Symbolic Comput.*, 5:29–35, 1988.

[Gri88] D. Grigor'ev. The complexity of deciding Tarski algebra. *J. Symbolic Comput.*, 5:65–108, 1988.

[GV88] D. Grigor'ev and N.N. Vorobjov. Solving systems of polynomial inequalities in subexponential time. *J. Symbolic Comput.*, 5:37–64, 1988.

[GV92] D. Grigor'ev and N.N. Vorobjov. Counting connected components of a semialgebraic set in subexponential time. *Comput. Complexity*, 2:133–186, 1992.

[HRR91] J. Heintz, T. Recio, and M.-F. Roy. Algorithms in real algebraic geometry and applications to computational geometry. In J.E. Goodman, R. Pollack, and W. Steiger, editors, *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, pages 137–164. Amer. Math. Soc., Providence, 1991.

[HRS89] J. Heintz, M.-F. Roy, and P. Solernó. On the complexity of semi-algebraic sets. In *Proc. Internat. Fed. Info. Process. 89*, pages 293–298. North-Holland, San Francisco, 1989.

[HRS90] J. Heintz, M.-F. Roy, and P. Solernó. Sur la complexité du principe de Tarski-Seidenberg. *Bull. Soc. Math. France*, 118:101–126, 1990.

[Hof89] C.M. Hoffmann. *Geometric and Solid Modeling.* Morgan Kaufmann, San Mateo, 1989.

[Lat91] J.-C. Latombe. *Robot Motion Planning.* Kluwer, Boston, 1991.

[Mil64] J. Milnor. On the Betti numbers of real algebraic varieties. *Proc. of Amer. Math. Soc.*, 15:275–280, 1964.

[Mis93] B. Mishra. *Algorithmic Algebra.* In *Texts and Monographs in Comput. Sci.*, Springer-Verlag, New York, 1993.

[PR93] R. Pollack and M.-F. Roy. On the number of cells defined by a set of polynomials. *C.R. Acad. Sci. Paris Sér. I Math.*, 316:573–577, 1993.

[Ren91] J. Renegar. Recent progress on the complexity of the decision problem for the reals. In J.E. Goodman, R. Pollack, and W. Steiger, editors, *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, pages 287–308. Amer. Math. Soc., Providence, 1991. Also in [CJ95].

[Ren92a] J. Renegar. On the computational complexity and geometry of the first-order theory of the reals: Part I. *J. Symbolic Comput.*, 13:255–299, 1992.

[Ren92b] J. Renegar. On the computational complexity and geometry of the first-order theory of the reals: Part II. *J. Symbolic Comput.*, 13:301–327, 1992.

[Ren92c] J. Renegar. On the computational complexity and geometry of the first-order theory of the reals: Part III. *J. Symbolic Comput.*, 13:329–352, 1992.

[SS83] J.T. Schwartz and M. Sharir. On the piano movers' problem: II. General techniques for computing topological properties of real algebraic manifolds. *Adv. Appl. Math.*, 4:298–351, 1983.

[Tar51] A. Tarski. *A Decision Method for Elementary Algebra and Geometry.* Univ. of California Press, Berkeley, 1951. Also in [CJ95].

[Tho65] R. Thom. *Sur l'Homologie des Variétés Réelles.* Differential and Combinatorial Topology, Princeton Univ. Press, Princeton, 255–265, 1965.

GEOMETRIC DATA STRUCTURES AND SEARCHING

30 POINT LOCATION

Jack Snoeyink

INTRODUCTION

A basic question for computer applications that employ geometric structures (e.g., for computer graphics, geographic information systems, robotics, and databases) is: "Where am I?" Given a set of disjoint geometric objects, the ***point-location problem*** asks for the object containing a query point. Instances of the problem vary in the dimension and type of objects and whether the set is static or dynamic. Solutions vary in preprocessing time, space used, and query time.

Point location has inspired several techniques for structuring geometric data, which we survey in this chapter. We begin with point location in one dimension (Section 30.1) or in one polygon (Section 30.2). For point location in the plane, we look at two optimal static methods (Section 30.3), the current best dynamic methods (Section 30.4), other methods in current practice (Section 30.5), and two other methods of general import (Section 30.6). There are fewer results on point location in higher dimensions (Section 30.7); randomized methods are covered briefly.

30.1 ONE-DIMENSIONAL POINT LOCATION

The simplest nontrivial instance of point location is list searching. The objects are points $x_1 \leq \cdots \leq x_n$ on the real line, presented in arbitrary order, and the intervals between them, (x_i, x_{i+1}) for $1 \leq i < n$. The answer to a query q is the name of the object containing q.

The list searching problem already illustrates several aspects of general point location problems.

GLOSSARY

Preprocessing/queries: If one assumes that many queries will ask for the same input, then resources can profitably be spent building data structures to facilitate the search.

Query time: Computation time to answer a single query, given a point location data structure. Usually a worst-case upper bound, expressed as a function of the number of objects in the structure, n.

Preprocessing time: Time required to build a point location structure for n objects.

Space: Memory used by the point location structure for n objects.

Decomposable problem: A problem whose answer can be obtained from the answers to the same problem on the sets of an arbitrary partition of the in-

put [Ben79, BS80]. As initially stated, one-dimensional point location is not decomposable—taking subsets of the points gives different intervals. If, however, we choose to name each interval by its lower endpoint, then we can report (the lower endpoint of) the interval containing a query from the intervals in the subproblems—simply report the highest "lower endpoint" from the answers to queries on subsets. Most point location problems can be made decomposable and are easier to solve in such a form.

Dynamic point location: Maintains a location data structure as points are inserted and deleted. The one-dimensional point location structures can be made dynamic without changing their asymptotic performances.

Randomized point location: Structures whose preprocessing algorithms may make random choices in an attempt to avoid poor performance caused by pathological input data. Preprocessing and query times are reported as expectations over these random choices. Randomized algorithms make no assumptions on the input distribution—they assume worst-case input—but by using a sample they obtain information about the input distribution, and can achieve good expected performance with simple algorithms.

LIST SEARCH AS ONE-DIMENSIONAL POINT LOCATION

Table 30.1.1 reports query time, preprocessing time, and space for five search methods. Linear search requires no additional data structure if the problem is decomposable. Binary search trees or randomized search trees [AS89, Pug90] require a total order and an ability to do comparisons. An adversary argument shows that these comparison-based query algorithms require $\Omega(\log n)$ comparisons.

If the points are restricted to integers $[1, \ldots, U]$, then van Emde Boas has shown how hashing techniques can be applied in stratified search trees to answer a query in $O(\log \log U)$ time. A useful method in practice is to partition the input range into b equal-sized buckets, and to answer a query by searching the bucket containing the query.

TABLE 30.1.1 List search as one-dimensional point location.

TECHNIQUE	QUERY	PREPROC	SPACE
Linear search	$O(n)$	none	data only
Binary search	$O(\log n)$	$O(n \log n)$	$O(n)$
Randomized tree	$O(\log n)$ expec	$O(n \log n)$ expec	$O(n)$
van Emde Boas tree [vEKZ77]	$O(\log \log U)$	$O(n)$ expec	$O(n)$
Bucketing	$O(n)$	$O(n + b)$	$O(n + b)$

30.2 POINT-IN-POLYGON

The second simplest form of point location is to determine whether a query point q lies inside a given n-sided polygon P [Hai94]. Without preprocessing the polygon,

one may use parity of the winding or crossing numbers: count intersections of a ray from q with the boundary of polygon P. Point q is inside P iff the number is odd. A query takes $O(n)$ time.

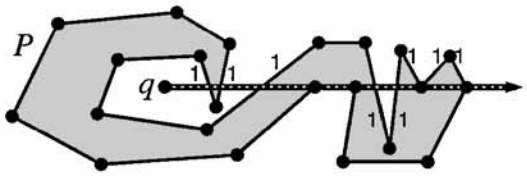


FIGURE 30.2.1
Counting degenerate crossings: eight crossings imply $q \notin P$.

One must count carefully in degenerate cases when the ray passes through a vertex or edge of P. When the ray is horizontal, as in Figure 30.2.1, then edges of P can be considered to contain their lower but not their upper endpoints. Edges inside the ray can be ignored. This is consistent with the count obtained by perturbing the ray infinitesimally upward. Stewart [Ste91] further considered instances in which vertex and edge positions may be imprecise.

To obtain sublinear query times, preprocess the polygon P using the more general techniques of the next sections.

30.3 PLANAR POINT LOCATION: OPTIMAL METHODS

Theoretical research has produced a number of planar point location methods that are optimal for comparison-based models: $O(n \log n)$ time to preprocess a planar subdivision with n vertices for $O(\log n)$ time queries using $O(n)$ space. Preprocessing time reduces to linear if the input is given in an appropriate format, and some preprocessing schemes have been parallelized (see Chapter 36). We focus on the data structuring techniques used to reach optimality.

In a planar subdivision, point location can be made decomposable by storing with each edge the name of the face immediately above. If one knows for each subproblem the edge below a query, then one can determine the edge directly below and report the containing face, even for an arbitrary partition into subproblems.

GLOSSARY

Planar subdivision: A partitioning of a region of the plane into point ***vertices,*** line segment ***edges,*** and polygonal ***faces.***

Size of a planar subdivision: The number of vertices, usually denoted by n. Euler's relation bounds the numbers of edges $e \leq 3n - 6$ and faces $f \leq 2n - 4$; often the constants are suppressed by saying that the number of vertices, edges, and faces are all $O(n)$.

Monotone subdivision: A planar subdivision whose faces are x-monotone polygons: i.e., the intersection of any face with any vertical line is connected.

Triangulation/trapezoidation: Planar subdivisions whose faces are triangles (resp. trapezoids whose parallel sides are all parallel).

Dual graph: A planar subdivision can be viewed as a graph with vertices joined by edges. The dual graph has a node for each face and an arc joining two faces if they share a common edge.

SLABS AND PERSISTENCE

By drawing a vertical line through every vertex, as shown in Figure 30.3.1(a), we obtain vertical ***slabs*** in which point location is almost one-dimensional. Two binary searches suffice to answer a query: one on x-coordinates for the slab containing q, and one on edges that cross that slab. Query time is $O(\log n)$, but space may be quadratic if all edges are stored with the slabs that they cross.

The location structures for adjacent slabs are similar—the segments with right endpoint on the slab boundary are removed and those with left endpoint are added. A balanced binary search tree could be used to construct each slab in turn using $O(n \log n)$ total preprocessing time and a linear number of node updates. To store all slabs in linear space, Sarnak and Tarjan [ST86] add to this the idea of *persistence*.

Rather than modifying a node to update the tree, copy the $O(\log n)$ nodes on the path from the root to the node to be modified. Then modify the copies. This ***node-copying persistence*** preserves the former tree and gives access to a new tree (through the new root) that represents the adjacent slab. The total space for n trees is $O(n \log n)$. Figure 30.3.1(a) provides an illustration. The initial tree contains 8 and 1. (Recall that edges are named by the face immediately above.) Then 2, 3, and 7 are added, 8 is copied during rebalancing, but node 1 is not changed. When 6 is added, 7 is copied in the rebalancing, but the two subtrees holding 1, 2, 3, and 8 are not changed.

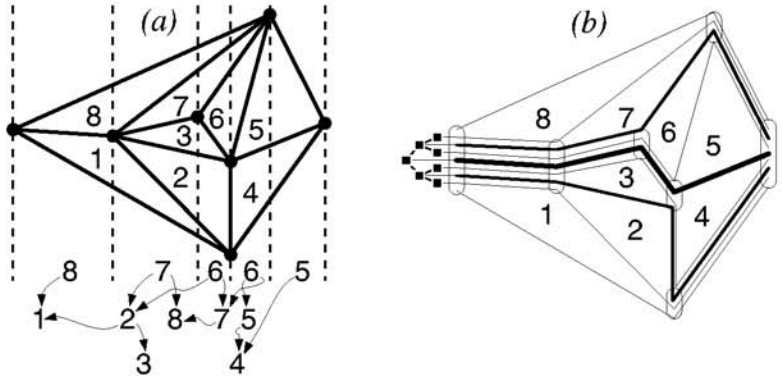
Limited node copying reduces the space to linear. Give each node spare left and right pointers and left and right time-stamps. Build a balanced tree for the initial slab. When a pointer is to be modified, use a spare and time-stamp it, if there is a spare available. Future searches can use the time-stamp to determine whether to follow the pointer or the spare. Otherwise, copy the node and modify its ancestor to point to the copy. If the slab location structures are maintained with $O(1)$ rotations per update, then the amortized cost of copying is also $O(1)$ per update.

Preprocessing takes $O(n \log n)$ time to sort by x coordinates and build either persistent data structure. To compare constants with other methods, the data structure has about 12 entries per edge because of extra pointers and copying. Searches take about $4 \log_2 N$ comparisons, where N is the number of edges that can intersect a vertical line; this is because there are two comparisons per node and "$O(1)$ rotation" tree-balancing routines are balanced only to within a factor of two.

SEPARATING CHAINS AND FRACTIONAL CASCADING

If a subdivision is monotone, then its faces can be totally ordered consistent with aboveness; in other words, we can number faces $1, \ldots, f$ so that any vertical line encounters lower numbers below higher numbers. The ***separating chain*** between the faces $< k$ and those $\geq k$ is a monotone chain of edges [LP77]. Figure 30.3.1(b) shows all separating chains for a subdivision; the middle chain, $k = 5$, is shown darkest.

FIGURE 30.3.1
Optimal static methods: (a) *Slab (persistent);* (b) *separating chain (fractional cascading).*



A balanced binary tree of separating chains can be used for point location: if query point q is above chain i and below chain $i+1$, then q is in face i. To preserve linear space we need to avoid the duplication of edges in chains that can be seen in Figure 30.3.1(b).

Note that the separating chains that contain an edge are defined by consecutive integers; we can store the first and last with each edge. Then form a binary tree in which each subtree stores the separating chains from some interval—at each node, store the edges of the median chain that have not been stored higher in the tree, and recursively store the intervals below and above the median in the left and right subtrees respectively. The root, for example, stores all edges of the middle chain. Since no edge is stored twice, this data structure takes $O(n)$ space.

As we search the tree for a query point q, we keep track of the edges found so far that are immediately above and below q. (Initially, no edges have been found.) Now, the root of the subtree to search is associated with a separating chain. If that chain does not contain one of the edges that we know is above or below q, then we search the x-coordinates of edges stored at the node and find the one on the vertical line through q. We then compare against the separating chain and recursively search the left or right subtree. Thus, this separating chain method [LP77] inspects $O(\log n)$ tree nodes at a cost of $O(\log n)$ each, giving $O(\log^2 n)$ query time.

To reduce the query time, we can use fractional cascading [CG86, EGS86] for efficient search in multiple lists. We have a tree at whose every node we need to search a list by x-coordinates. Pass every fourth x-coordinate from a child list to its parent, and establish connections so that knowing one's position in the parent list gives one's position in the child to within four nodes. This increases the data structure size by a constant, but means that all x-coordinate searches after the first take constant time.

Preprocessing takes $O(n)$ time on a monotone subdivision; arbitrary planar subdivisions can be made monotone by plane sweep in $O(n \log n)$ time. One can trade off space and query time in fractional cascading, but typical constants are 8 entries per edge for a query time of $4 \log_2 n$.

30.4 PLANAR POINT LOCATION: DYNAMIC

In dynamic planar point location, the subdivision can be updated by adding or deleting vertices and edges. Unlike the static case, algorithms that match the performance of one-dimensional point location have not yet been found. Again, we focus on the data structures used by the best methods, summarized in Table 30.4.1.

GLOSSARY

Updates: A dynamic planar subdivision is most commonly updated by inserting or deleting a vertex or edge. Update time usually refers to the worst-case time for a single insertion or deletion.

Chain insertion/deletion: Some methods support insertion or deletion of a chain of k vertices and edges, so that this is faster than doing k insertions or k deletions.

Vertex expansion/contraction: Updating a planar subdivision by splitting a vertex into two vertices joined by an edge, or the inverse: contracting an edge and merging the two endpoints into one. This operation, supported by the "primal/dual spanning tree" (discussed below), is important for point location in three-dimensional subdivisions.

Amortized update time: When times are reported as amortized, then an individual operation may be expensive, but the total time for k operations, starting from an empty data structure, will take at most k times the amortized bound.

TABLE 30.4.1 Dynamic point location results.

TECHNIQUE	QUERY	UPDATE	SPACE	UPDATES SUPPORTED
Interval tree [CJ92]	$O(\log^2 n)$	$O(\log n)$	$O(n)$	ins/del edge & chain
with frac casc [BJM94]	$O(\log n \log\log n)$	$O(\log^2 n)$	$O(n)$	amort del, ins faster
Trapezoid method [CT92]	$O(\log n)$	$O(\log^2 n)$	$O(n \log n)$	ins/del vertex & edge
Pr/dual span tree [GT91]	$O(\log^2 n)$	$O(\log n)$	$O(n)$	ins/del edge & chain, expand/contract vertex
amortized	$O(\log n \log\log n)$	$O(1)$	$O(n)$	

DYNAMIC INTERVAL TREE

An ***interval tree*** storing segments can be defined recursively: the root stores segments that cross a given vertical line ℓ; segments to the left (right) are stored in an interval tree that is the left (right) child of the root. To locate a query point q, one must search down the interval tree, answering the following subproblem at $O(\log n)$ nodes: Given a set of segments S that intersect a common line ℓ, which segment is immediately below q?

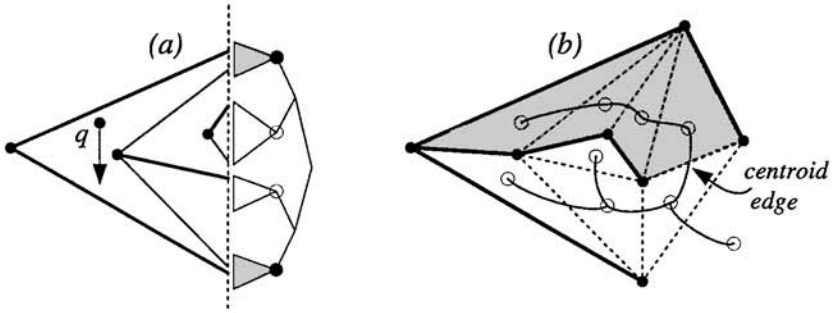
Cheng and Janardan [CJ92] solve this subproblem by a priority-tree search, which allows them to use the interval tree for dynamic point location. In an interval

tree node, store the segments in a binary search tree ordered along ℓ, and store in each subtree a pointer to the "priority segment" with endpoint farthest left of ℓ. (Priority must also be stored on the right.) At each level of the search tree, only two candidate subtrees may contain the segment below q—the ones whose priority segments are immediately above and below q. Figure 30.4.1(a) illustrates a case in which the search continues in the two shaded subtrees.

Performing this search in each node of the interval tree leads to $O(\log^2 n)$ query time using $O(n)$ space. Constants are moderate, with only 4 or 5 entries per edge and 6 comparisons per search step. Updates take $O(\log n)$ time with larger constants; they must maintain tree balance and segment priorities.

Baumgarten et al. [BJM94] use fractional cascading on blocks of $O(\log^2 n)$ segments in each interval tree node to speed up queries to $O(\log n \log\log n)$, at the cost of slowing insertions to $O(\log n \log\log n)$ amortized, and deletions to $O(\log^2 n)$. This is a surprising development because fractional cascading requires a global order that is difficult to establish in interval tree techniques.

FIGURE 30.4.1
Dynamic methods: (a) *Priority search (interval tree);* (b) *primal/dual spanning tree.*



TRAPEZOID METHOD

Preparata's [Pre81] static "trapezoid method" uses $O(n \log n)$ space, but can be made dynamic. Start with a vertical trapezoid that contains a planar subdivision. If a trapezoid interior intersects a segment s, but does not contain its endpoints, then use s to cut the trapezoid into two. Otherwise, if a trapezoid interior contains subdivision vertices, then split the trapezoid vertically by a line through the median vertex. This leads to a tree with depth (and query time) $4 \log n$. Experiments [EKA84] suggest that this method performs well, despite its worst-case size.

Chiang and Tamassia [CT92, CPT96] have the best dynamization, based on *link-cut* trees [ST83], which support in $O(\log n)$ time the operation of linking two trees by adding an edge, and the inverse, cutting an edge to make two trees.

PRIMAL/DUAL SPANNING TREE

A monotone subdivision has a ***monotone spanning tree*** in which all root-to-leaf

paths are monotone. Each edge not in the tree closes a cycle and defines a monotone polygon.

In any planar graph whose faces are simple polygons, the duals of edges not in the spanning tree form a dual spanning tree of faces, as in Figure 30.4.1(b). Goodrich and Tamassia [GT91] use a centroid decomposition of the dual tree to guide comparisons with monotone polygons in the primal tree. The centroid edge, which breaks the dual tree into two nearly-equal pieces, is indicated in Figure 30.4.1(b). The primal edge creates the shaded monotone polygon; if the query is inside then we recursively explore the corresponding piece of the dual tree. Using link-cut trees, the centroid decomposition can be maintained in logarithmic time per update, giving a dynamic point-location structure with $O(\log^2 n)$ query time.

In the static setting, fractional cascading can turn this into an optimal point location method. Dynamic fractional cascading can be used to reduce the dynamic query time and to obtain $O(1)$ amortized update time.

The dual nature of the structure supports insertion and deletion of dual edges, which correspond to expansion and contraction of vertices. These are needed to support static three-dimensional point location via persistence. Furthermore, a k-vertex monotone chain can be inserted/deleted in $O(\log n + k)$ time.

OPEN PROBLEMS

1. Improve dynamic planar point location to simultaneously attain $O(n)$ space and $O(\log n)$ query and update time, or establish a lower bound.
2. Can persistent data structures be made dynamic? The fact that data are copied seems to work against maintaining a data structure under insertions and deletions.

30.5 PLANAR POINT LOCATION: COMMON PRACTICE

Programming complexity and nonnegligible asymptotic constants mean that optimal point location techniques are used less than might be expected.

PICK HARDWARE

Graphic workstations employ special "pick hardware" that draws objects on the screen and returns a list of objects that intersect a query pixel. The hardware imposes a minimum time of about 1/30th of a second on a pick operation, but hundreds of thousands of polygons may be considered in this time.

BUCKETING AND SPATIAL INDEX STRUCTURES

Because data in practical applications tend to be evenly distributed, bucketing techniques are far more effective [AEI+85, EKA84] than worst-case analysis would predict. For problems in two and three dimensions, a uniform grid will often trim data to a manageable size.

Adaptive data structures for more general spatial indexing, such as k-d trees, quadtrees, BANG files, R-trees, and their relatives [Sam90], can be used as filters for point location—these techniques are common in databases and geographic information systems.

SUBDIVISION WALKING

Applications that store planar subdivisions with their adjacency relations, such as geographic information systems, can walk through the regions of the subdivision from a known position p to the query q.

To walk a subdivision with $O(n)$ edges, compute the intersections of $\overline{pq}$ with the current region and determine if q is inside. If not, let q' denote the intersection point closest to q. Advance to the region incident to q' that contains a point in the interior of $\overline{q'q}$ and repeat. In the worst case, this walk takes $O(n)$ time. The application literature typically claims $O(\sqrt{n})$ time, which is the average number of intersections with a line under the assumption that vertices and edges of the subdivision are evenly distributed. When combined with bucketing techniques (for example, maintaining a regular grid of known positions and starting from the closest to answer a query), walking is an effective, practical location method.

Guibas and Stolfi's [GS85] algorithm for Delaunay triangulation walks from edge to edge, and thus depends on an acyclicity theorem (Sections 20.4 and 22.1) that, as Heckbert points out, does not hold for arbitrary triangulations and can lead to cycles. A robust walk should remember its starting point and handle vertices on the traversed segment as if they had been perturbed consistently.

OPEN PROBLEM

What are the characteristics of an average planar subdivision? This unanswerable question is the primary impediment to analyzing effective point-location schemes that have poor worst-case performance (e.g., bucketing). Perhaps, however, point location schemes can be analyzed in terms of subdivision characteristics (vertex and face degrees, edge lengths, and vertex distribution) so that appropriate methods can be chosen for different types of data.

30.6 PLANAR POINT LOCATION: TWO EARLY METHODS

The first two optimal planar point location methods, which happen to be impractical because of their large constant factors, are still important for other applications.

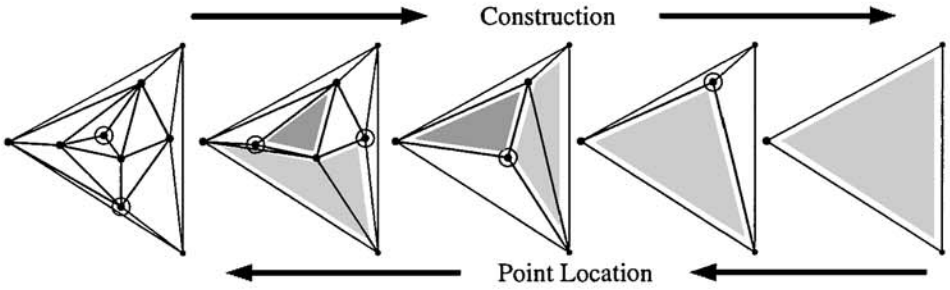
DOBKIN-KIRKPATRICK HIERARCHY

Kirkpatrick's point location scheme [Kir83] has become an important tool for problems on convex polyhedra (see Chapter 33). This scheme creates a hierarchy of subdivisions in which all faces, including the outer face, are triangles.

In every planar triangulation, one can find (in linear time) an independent set of low-degree vertices that consists of a constant fraction of all vertices. In Fig-

ure 30.6.1 these are circled and, in the next picture, are removed and the shaded hole is retriangulated if necessary. Repeating this "coarsening" operation a logarithmic number of times gives a constant-size triangulation.

FIGURE 30.6.1
Hierarchical triangulation.



To locate the triangle containing a query point q, start by finding the triangle in the coarsest triangulation, at right in Figure 30.6.1. Knowing the hole (shaded) that this triangle came from, one need only replace the missing vertex and check the incident triangles to locate q in the previous, finer triangulation.

Given a triangulation, preprocessing takes $O(n)$ time, but the hidden constants on time and space are large. For example, choosing the independent set by greedily taking vertices in order of increasing degree up to 12 guarantees almost 1/7th of the vertices, which leads to a data structure with $14n$ triangles in which a query can take more than $50 \log_2 n$ comparisons.

PLANAR SEPARATOR THEOREM

The first optimal point location scheme is based on Lipton and Tarjan's ***planar separator theorem*** [LT80] that every planar graph of n nodes has a set of $O(\sqrt{n})$ nodes that partition it into roughly equal pieces. When applied to the dual graph of a planar subdivision, the nodes are a small set of faces that partition the remainder of the faces: simple quadratic-space methods can be used to determine which set of the partition needs to be searched recursively.

Goodrich [Goo95] gave a linear-time construction of a family of planar separators in his parallel triangulation algorithm. The fact that embedded graphs have small separators continues to be important in theoretical work.

30.7 LOCATION IN HIGHER DIMENSIONS

In higher dimensions, known point location methods do not achieve both linear space and logarithmic query time. Linear space can be attained by relatively straightforward linear search, such as the point-in-polygon test.

Logarithmic time, or $O(d \log n)$ time, can be obtained by projection [DL76]:

project the $d-2$ faces of a subdivision to an arrangement in $d-1$ dimensions and recursively build a point location structure for the arrangement in the projection. Knowing the cell in the projection gives a list of the possible faces that project to that cell, so an additional logarithmic search can return the answer. The worst-case space required is $O(n^{2^d})$.

Because point location is decomposable, batching can trade space for time: preprocessing n/k groups of k facets into structures with $S(k)$ space and $Q(k)$ time gives, in total, $O(nS(k)/k)$ space and $O(nQ(k)/k)$ query time.

Clever ways of batching can lead to better structures. Randomized methods can often reduce the dependence on dimension from doubly- to singly-exponential, since random samples can be good approximations to a set of geometric objects. They can also be used with objects that are implicitly defined.

We should mention that convex polyhedra can be preprocessed using the Dobkin-Kirkpatrick hierarchy described in the previous section so that the point-in-convex-polyhedron test does take $O(n)$ space and $O(\log n)$ query time.

THREE-DIMENSIONAL STATIC POINT LOCATION

Dynamic location structures can be used for static spatial point location in one higher dimension by employing persistence. If one swept a plane through a subdivision of three-space into polyhedra, one could see the intersection as a dynamic planar subdivision in which vertices (intersections of the sweep plane with edges) move along linear trajectories. Whenever the sweep plane passes through a vertex in space, vertices in the plane may join and split.

Goodrich and Tamassia's primal/dual method supports the necessary operations to maintain a point location structure for the sweeping plane. Using node-copying to make the structures persistent gives an $O(n \log n)$ space structure that can answer queries in $O(\log^2 n)$ time. Preprocessing takes $O(n \log n)$ time.

RECTILINEAR SUBDIVISIONS

Restricting attention to rectilinear (orthogonal) subdivisions permits better results via data structures for orthogonal range search. The ***skewer tree***, a multidimensional interval tree, gives static point location among n rectangular prisms with $O(n)$ space and $O(\log^{d-1} n)$ query time after $O(n \log n)$ preprocessing [EHH86]. In dimensions two and three, stratified trees and perfect hashing [DKM+94] can be used to obtain $O((\log \log U)^{d-1})$ query times in a fixed universe $[1, \ldots, U]$, or $O(\log n)$ query time in general.

POINT LOCATION AMONG ALGEBRAIC VARIETIES

Chazelle and Sharir [CS90] consider point location in a general setting, among n algebraic varieties of constant maximum degree b in d-dimensional Euclidean space. They augment Collins's cylindrical algebraic decomposition to obtain an $O(n^{2^{d-1}})$-space, $O(\log n)$-query time structure after $O(n^{2^{d+6}})$ preprocessing. Hidden constants depend on the degrees of projections and intersections, which can be b^{4^d}.

This method provides a general technique to obtain subquadratic solutions to optimization problems that minimize a function $\{F(a,b) \mid a \in A, b \in B\}$, where $F(a,b)$ has a constant-size algebraic description. For a fixed b, F is algebraic in a. Thus, small batches of points from B can be preprocessed in subquadratic time, and each a can be tested against each batch, again in subquadratic time.

OPEN PROBLEMS

1. Find an optimal method for static (or dynamic) point location in a three-dimensional subdivision with n vertices and $O(n)$ faces: $O(n)$ space and $O(\log n)$ query time.
2. In a subdivision of a d-dimensional rectangular prism into n prisms, is there an optimal $O(\log n)$-query, $O(n)$-space point location method? The constants hidden by the big-O may depend on d. Under a pointer model of computation, this is already open for $d = 3$.

RANDOMIZED POINT LOCATION

TABLE 30.7.1 Randomized point location in arrangements.

TECHNIQUE	OBJECTS	QUERY	PREPROC	SPACE
Random sample [Cla87]	hyperplanes	$O(c^d \log n)$ exp	$O(n^{d+1+\epsilon})$ exp	$O(n^{d+\epsilon})$
Derandomized [CF94]	hyperplanes	$O(c^d \log n)$	$O(n^{2d+1})$	$O(n^d)$
Random sample [MS91]	dyn hpl $d \leq 4$	$O(\log n)$ exp	$O(n^{d+\epsilon})$ exp	$O(n^{d+\epsilon})$
Epsilon nets [Mei93]	hyperplanes	$O(d^5 \log n)$ exp	$O(n^{d+1+\epsilon})$ exp	$O(n^{d+\epsilon})$

The techniques of Chapter 34 can lead to good point location methods when a random sample of a set of objects can be used to approximate the whole. Arrangements of hyperplanes in dimension d are a good example. A random sample of hyperplanes divides space into cells intersected by few hyperplanes; recursively sampling in each cell gives a point location structure for the arrangement. Table 30.7.1 lists the performance of some randomized point location methods for hyperplanes. Query time can be traded for space by choosing larger random samples.

The randomized incremental construction algorithms of Chapter 34 are simple because they naturally build randomized point location structures along with the objects that they aim to construct [Mul93, Sei93]. These have good "tail bounds" and work well as insertion-only location structures.

Randomized point location structures can be made fully dynamic by lazy deletion and randomized rebuild techniques [dBDS94, MS91]; they maintain good expected performance if random elements are chosen for insertion and deletion. That is, the sequence of insertions and deletions may be specified, but the elements are to be chosen independently of their roles in the data structure.

IMPLICIT POINT LOCATION

In some applications of point location, the objects are not given explicitly. A planar motion planning problem may ask whether a start and a goal point are in the same cell of an arrangement of constraint segments or curves, without having explicit representations of all cells.

Consider a simple example: an arrangement of n lines, which defines nearly n^2 bounded cells. Without storing all cells, we can determine whether two points p and q are in the same cell by preprocessing $\sqrt{n}$ subarrangements of $\sqrt{n}$ lines ($O(n\sqrt{n})$ cells in all) and making sure that p and q are together in each subarrangement. If the lines are put into batches by slope, then within the same asymptotic time, an algorithm can return the pair of lines defining the lowest vertex as a unique cell name.

Implicit location methods are often seen as special cases of range queries (Chapter 31) or vertical ray shooting [Aga91]. Table 30.7.2 lists results on implicit location among line segments, which depend upon tools discussed in Chapters 31, 32, and 34, specifically random sampling, ϵ-net theory, and spanning trees with low stabbing number.

TABLE 30.7.2 Implicit point location results for arrangements of n line segments.

TECHNIQUE	QUERY	PREPROC	SPACE
Span tree lsn [Aga92]	$O(\sqrt{n}\log^2 n)$	$O(n^{3/2}\log^{\omega} n)$	$O(n\log^2 n)$
Batch sp tree [AvK94]	$O\left((n/\sqrt{s})\log^2(n/\sqrt{s})+\log n\right)$	$O\left(sn(\log(n/\sqrt{s})+1)^{2/3}\right)$	$n\sqrt{\log n}\le s\le n^2$

30.8 SOURCES AND RELATED MATERIAL

SURVEYS

Further references may be found in these surveys.

[Pre90]: A survey of planar point-location algorithms.

[Hai94, Wei94]: Point-in-polygon algorithms in *Graphics Gems IV*, with code.

[DS87, EOS84, O'R88]: General surveys that include point location.

RELATED CHAPTERS

Chapter 21: Arrangements
Chapter 22: Triangulations
Chapter 23: Polygons

Chapter 31: Range searching
Chapter 32: Ray shooting and lines in space
Chapter 34: Randomized algorithms
Chapter 36: Parallel algorithms in geometry
Chapter 42: Computer graphics

REFERENCES

[AEI+85] Ta. Asano, M. Edahiro, H. Imai, M. Iri, and K. Murota. Practical use of bucketing techniques in computational geometry. In G.T. Toussaint, editor, *Computational Geometry*, pages 153–195. North-Holland, Amsterdam, 1985.

[Aga91] P.K. Agarwal. Geometric partitioning and its applications. In J.E. Goodman, R. Pollack, and W. Steiger, editors, *Computational Geometry: Papers from the DIMACS Special Year*. Amer. Math. Soc., Providence, 1991.

[Aga92] P.K. Agarwal. Ray shooting and other applications of spanning trees with low stabbing number. *SIAM J. Comput.*, 21:540–570, 1992.

[AS89] C.R. Aragon and R. Seidel. Randomized search trees. In *Proc. 30th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 540–545, 1989.

[AvK94] P.K. Agarwal and M. van Kreveld. Implicit point location in arrangements of line segments, with an application to motion planning. *Internat. J. Comput. Geom. Appl.*, 4:369–383, 1994.

[Ben79] J.L. Bentley. Decomposable searching problems. *Inform. Process. Lett.*, 8:244–251, 1979.

[BJM94] H. Baumgarten, H. Jung, and K. Mehlhorn. Dynamic point location in general subdivisions. *J. Algorithms*, 17:342–380, 1994.

[BS80] J.L. Bentley and J.B. Saxe. Decomposable searching problems I: Static-to-dynamic transformation. *J. Algorithms*, 1:301–358, 1980.

[CF94] B. Chazelle and J. Friedman. Point location among hyperplanes and unidirectional ray-shooting. *Comput. Geom. Theory Appl.*, 4:53–62, 1994.

[CG86] B. Chazelle and L.J. Guibas. Fractional cascading I: A data structuring technique. *Algorithmica*, 1:133–162, 1986.

[CJ92] S.W. Cheng and R. Janardan. New results on dynamic planar point location. *SIAM J. Comput.*, 21:972–999, 1992.

[Cla87] K.L. Clarkson. New applications of random sampling in computational geometry. *Discrete Comput. Geom.*, 2:195–222, 1987.

[CPT96] Y.-J. Chiang, F.P. Preparata, and R. Tamassia. A unified approach to dynamic point location, ray shooting, and shortest paths in planar maps. *SIAM J. Comput.*, 25:207–233, 1996.

[CS90] B. Chazelle and M. Sharir. An algorithm for generalized point location and its application. *J. Symbolic Comput.*, 10:281–309, 1990.

[CT92] Y.-J. Chiang and R. Tamassia. Dynamization of the trapezoid method for planar point location in monotone subdivisions. *Internat. J. Comput. Geom. Appl.*, 2:311–333, 1992.

[dBDS94] M. de Berg, K. Dobrindt, and O. Schwarzkopf. On lazy randomized incremental construction. In *Proc. 26th Annu. ACM Sympos. Theory Comput.*, pages 105–114, 1994.

[DKM+94] M. Dietzfelbinger, A. Karlin, K. Mehlhorn, F. Meyer auf der Heide, H. Rohnert, and R.E. Tarjan. Dynamic perfect hashing: Upper and lower bounds. *SIAM J. Comput.*, 23:738–761, 1994.

[DL76] D.P. Dobkin and R.J. Lipton. Multidimensional searching problems. *SIAM J. Comput.*, 5:181–186, 1976.

[DS87] D.P. Dobkin and D.L. Souvaine. Computational geometry: A user's guide. In J.T. Schwartz and C.K. Yap, editors, *Advances in Robotics 1: Algorithmic and Geometric Aspects of Robotics*, pages 43–93. Lawrence Erlbaum, Hillsdale, 1987.

[EGS86] H. Edelsbrunner, L.J. Guibas, and J. Stolfi. Optimal point location in a monotone subdivision. *SIAM J. Comput.*, 15:317–340, 1986.

[EHH86] H. Edelsbrunner, G. Haring, and D. Hilbert. Rectangular point location in d dimensions with applications. *Comput. J.*, 29:76–82, 1986.

[EKA84] M. Edahiro, I. Kokubo, and Ta. Asano. A new point-location algorithm and its practical efficiency: Comparison with existing algorithms. *ACM Trans. Graph.*, 3:86–109, 1984.

[EOS84] H. Edelsbrunner, M.H. Overmars, and R. Seidel. Some methods of computational geometry applied to computer graphics. *Comput. Vision Graph. Image Process.*, 28:92–108, 1984.

[Goo95] M.T. Goodrich. Planar separators and parallel polygon triangulation. *J. Comput. Syst. Sci.*, 51:374–389, 1995.

[GS85] L.J. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Trans. Graph.*, 4:74–123, 1985.

[GT91] M.T. Goodrich and R. Tamassia. Dynamic trees and dynamic point location. In *Proc. 23rd Annu. ACM Sympos. Theory Comput.*, pages 523–533, 1991.

[Hai94] E. Haines. Point in polygon strategies. In P. Heckbert, editor, *Graphics Gems IV*, pages 24–46. Academic Press, Boston, 1994.

[Kir83] D.G. Kirkpatrick. Optimal search in planar subdivisions. *SIAM J. Comput.*, 12:28–35, 1983.

[LP77] D.T. Lee and F.P. Preparata. Location of a point in a planar subdivision and its applications. *SIAM J. Comput.*, 6:594–606, 1977.

[LT80] R.J. Lipton and R.E. Tarjan. Applications of a planar separator theorem. *SIAM J. Comput.*, 9:615–627, 1980.

[Mei93] S. Meiser. Point location in arrangements of hyperplanes. *Inform. Comput.*, 106:286–303, 1993.

[MS91] K. Mulmuley and S. Sen. Dynamic point location in arrangements of hyperplanes. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 132–141, 1991.

[Mul93] K. Mulmuley. *Computational Geometry: An Introduction through Randomized Algorithms.* Prentice Hall, Englewood Cliffs, 1993.

[O'R88] J. O'Rourke. Computational geometry. *Annu. Rev. Comput. Sci.*, 3:389–411, 1988.

[Pre81] F.P. Preparata. A new approach to planar point location. *SIAM J. Comput.*, 10:473–482, 1981.

[Pre90] F.P. Preparata. Planar point location revisited. *Internat. J. Found. Comput. Sci.*, 1:71–86, 1990.

[Pug90] W. Pugh. Skip lists: A probabilistic alternative to balanced trees. *Commun. ACM*, 35:668–676, 1990.

[Sam90] H. Samet. *The Design and Analysis of Spatial Data Structures.* Addison-Wesley, Reading, 1990.

[Sei93] R. Seidel. Backwards analysis of randomized geometric algorithms. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms Combin.*, pages 37–68. Springer-Verlag, Berlin, 1993.

[ST83] D.D. Sleator and R.E. Tarjan. A data structure for dynamic trees. *J. Comput. Syst. Sci.*, 26:362–391, 1983.

[ST86] N. Sarnak and R.E. Tarjan. Planar point location using persistent search trees. *Commun. ACM*, 29:669–679, 1986.

[Ste91] A.J. Stewart. Robust point location in approximate polygons. In *Proc. 3rd Canad. Conf. Comput. Geom.*, pages 179–182, Vancouver, 1991.

[vEKZ77] P. van Emde Boas, R. Kaas, and E. Zijlstra. Design and implementation of an efficient priority queue. *Math. Systems Theory*, 10:99–127, 1977.

[Wei94] K. Weiler. An incremental angle point in polygon test. In P. Heckbert, editor, *Graphics Gems IV*, pages 16–23. Academic Press, Boston, 1994.

31 RANGE SEARCHING

Pankaj K. Agarwal

INTRODUCTION

Range searching is one of the central problems in computational geometry, because it arises in many applications and a wide variety of geometric problems can be formulated as range-searching problems. A typical range-searching problem has the following form. Let S be a set of n points in $\mathbb{R}^d$, and let $\mathcal{R}$ be a family of subsets of $\mathbb{R}^d$; elements of $\mathcal{R}$ are called ***ranges***. *We wish to preprocess S into a data structure so that for a query range R, the points in $S \cap R$ can be reported or counted efficiently.* Typical examples of ranges include rectangles, halfspaces, simplices, and balls. If we are only interested in answering a single query, it can be done in linear time, using linear space, by simply checking for each point $p \in S$ whether p lies in the query range. However, most of the applications call for querying the same set S several times (perhaps with periodic insertions and deletions), in which case we would like to answer a query faster by preprocessing S into a data structure.

Range counting and ***range reporting*** are just two instances of range-searching queries. Other examples include ***emptiness queries***, where one wants to determine whether $S \cap R = \emptyset$, and ***extremal queries***, where one wants to return a point with a certain property (e.g., returning a point with the largest x_1-coordinate). In order to encompass all different types of range-searching queries, a general range-searching problem can be defined as follows. Let $(\mathbf{S}, +)$ be a semigroup. For each point $p \in S$, we assign a weight $w(p) \in \mathbf{S}$. For a query range $R \in \mathcal{R}$, we wish to compute $\sum_{p \in S \cap R} w(p)$. For example, range-counting queries can be answered by setting $w(p) = 1$ for every $p \in S$ and choosing the semigroup to be $(\mathbb{N}, +)$, range-emptiness queries by setting $w(p) = 1$ and choosing the semigroup to be $(\{0, 1\}, \vee)$, and range-reporting queries by setting $w(p) = \{p\}$ and choosing the semigroup to be $(2^S, \cup)$.

Most range-searching data structures construct a family of "canonical" subsets of S, and for each canonical subset C, they store the weight $w(A) = \sum_{p \in A} w(p)$. For a query range r, the data structure searches for a small subfamily of disjoint canonical subsets, $A_1, \ldots, A_k$, such that $\bigcup_{i=1}^{k} A_i = r \cap S$, and then computes $\sum_{i=1}^{k} w(A_i)$. In order to expedite the search, the structure also stores some auxiliary information. Typically, the canonical subsets are organized in a (multi-level) tree-like data structure, each of whose nodes v is associated with a canonical subset A; v stores the weight $w(A)$ and some auxiliary information. A query is answered by searching the tree in a top-down fashion, using the auxiliary information to guide the search.

MODEL OF COMPUTATION

The performance of a data structure is measured by the time spent in answering a query, called the ***query time*** and denoted by $Q(n, d)$; by the size of the data structure, denoted by $S(n, d)$; and by the time constructing the data structure,

called the ***preprocessing time*** and denoted by $P(n, d)$. Since the data structure is constructed only once, its query time and size are more important than its preprocessing time. If a data structure supports insertion and deletion operations, the ***update time*** is also relevant. We should remark that the query time of a range-reporting query on any reasonable machine depends on the output size, so the query time for a range-reporting query consists of two parts—***search time***, which depends only on n and d, and ***reporting time***, which depends on n, d, and the output size. We will use k to denote the output size.

We assume that d is a small fixed constant, and that the big-O notation hides constants depending on d. The dependence on d of the performance of all the data structures mentioned here is exponential, which makes them unsuitable for large values of d. We assume that each memory cell can store $\log n$ bits. The upper bounds will be given on ***pointer-machine*** or ***RAM*** models. The main difference between the two models is that on the pointer machine a memory cell can be accessed only through a series of pointers while in the RAM model any memory cell can be accessed in constant time. Most of the lower bounds will be given in the so-called ***semigroup model***, which was originally introduced by Fredman [Fre81] and which is much weaker than the pointer machine or the RAM model. In the semigroup model, a data structure is regarded as a set of precomputed sums in the underlying semigroup. The size of the data structure is the number of sums stored, and the query time is the number of semigroup operations performed (on the precomputed sums) to answer a query; the query time ignores the cost of various auxiliary operations, e.g., the cost of determining which of the precomputed sums should be added to answer a query. A weakness of the semigroup model is that it does not allow subtractions even if the weights of points belong to a group. Therefore, we will also consider the ***group model***, in which both additions and subtractions are allowed.

The size of any range-searching data structure is at least linear, for it has to store each point (or its weight) at least once, and the query time on any reasonable model of computation (e.g., pointer machine, or RAM) is $\Omega(\log n)$ even for $d = 1$. Therefore, one would like to develop a linear-size data structure with logarithmic query time. Although near-linear-size data structures are known for orthogonal range searching in any fixed dimension that can answer a query in polylogarithmic time, no similar bounds are known for range searching with more complex ranges (e.g., simplices, disks). In such cases, one seeks a tradeoff between the query time and the size of the data structure—how fast can a query be answered using $n \log^{O(1)} n$ space, how much space is required to answer a query in $\log^{O(1)} n$ time, or what kind of tradeoff between the size and the query time can be achieved.

This chapter is organized as follows. In Section 31.1 we review the orthogonal range-searching data structures, and in Section 31.2 we review simplex range-searching data structures. Section 31.3 surveys other variants and extensions of range searching. In Section 31.4, we study intersection-searching problems, which can be regarded as a generalization of range searching. Finally, Section 31.5 deals with various optimization queries.

31.1 ORTHOGONAL RANGE SEARCHING

In d-dimensional orthogonal range searching, the ranges are d-rectangles, each of the form $\prod_{i=1}^{d} [a_i, b_i]$ where $a_i, b_i \in \mathbb{R}$. This is an abstraction of ***multi-key search-***

ing. For example, the points of S may correspond to employees of a company, each coordinate corresponding to a key such as age, salary, experience, etc. Queries of the form, e.g., "report all employees between the ages of 30 and 40 who earn more than $\$30,000$ and who have worked for more than 5 years," can be formulated as orthogonal range-reporting queries. Because of its numerous applications, orthogonal range searching has been studied extensively for the last 25 years. In this section we review recent data structures and lower bounds.

GLOSSARY

EPM: A pointer machine with + operation.

APM: A pointer machine with basic arithmetic and shift operations.

Faithful semigroup: A semigroup $(\mathbf{S}, +)$ is called faithful if for each $n > 0$, for any $T_1, T_2 \subseteq \{1, \ldots, n\}$ such that $T_1 \neq T_2$, and for every sequence of integers $\alpha_i, \beta_j > 0$ $(i \in T_1, j \in T_2)$, there are $s_1, s_2, \ldots, s_n \in \mathbf{S}$ such that

$$\sum_{i \in T_1} \alpha_i s_i \neq \sum_{j \in T_2} \beta_j s_j.$$

Notice that $(\mathbb{R}, +)$ is a faithful semigroup, but $(\{0,1\}, \text{XOR})$ is not a faithful semigroup.

Dominance query: A point $p \in \mathbb{R}^d$ is dominated by another point $q \in \mathbb{R}^d$ if $x_i(p) \leq x_i(q)$ for every $1 \leq i \leq d$. A dominance query is to compute the sum of the weights of all points in S that are dominated by a query point.

UPPER BOUNDS

Most orthogonal range-searching data structures are based on ***range trees***, introduced by Bentley [Ben80]. For $d = 1$, the range tree of S is an array storing S in a nondecreasing order. For $d > 1$, the range tree of S is a minimum-height binary tree with n leaves, whose ith leftmost leaf stores the point of S with the ith smallest x_1-coordinate. For an interior node v of T, let $S(v)$ denote the set of points stored at the leaves in the subtree rooted at v, let a_v (resp. b_v) be the smallest (resp. largest) x_1-coordinate of points in $S(v)$, and let $S^*(v)$ denote the projection of $S(v)$ onto the hyperplane $x_1 = 0$. The interior node v stores a_v, b_v, and a $(d-1)$-dimensional range tree constructed on $S^*(v)$. For any fixed dimension d, the size of T is $O(n \log^{d-1} n)$, and it can be constructed in time $O(n \log^{d-1} n)$. The range-reporting query for a rectangle $q = \prod_{i=1}^{d} [a_i, b_i]$ can be answered as follows. If $d = 1$, the query can be answered by a binary search. For $d > 1$, we traverse the range tree as follows. Suppose we are at a node v. If v is a leaf, then we report the point if it lies inside q. If v is an interior node and the interval $[a_v, b_v]$ does not intersect $[a_1, b_1]$, there is nothing to do. If $[a_v, b_v] \subseteq [a_1, b_1]$, we recursively search in the $(d-1)$-dimensional range tree stored at v, with the rectangle $\prod_{i=2}^{d} [a_i, b_i]$. Otherwise, we recursively visit both children of v. The query time of this procedure is $O(\log^d n + k)$, which can be improved to $O(\log^{d-1} n + k)$, using the ***fractional-cascading*** technique (Section 30.3). A range tree can also answer a range-counting query in time $O(\log^{d-1} n)$.

A range tree can be regarded as a ***multi-level*** tree structure in the sense that each node of the tree stores another tree structure as a second level structure; each level of the tree increases the size of the tree by a logarithmic factor. As we will see below, such multi-level tree structures play a key role in developing efficient data structures for range searching and related problems.

The best-known data structures for orthogonal range searching are due to Chazelle [Cha86, Cha88b], who used compressed range trees and other techniques (such as filtering search) to improve the storage and query time. His results in the plane, under various models of computation, are summarized in Table 31.1.1; the preprocessing time of each data structure is $O(n \log n)$.

TABLE 31.1.1 Best upper bounds known on planar orthogonal range searching.

PROBLEM	MODEL	$S(n)$	$Q(n)$
Counting	RAM	n	$\log n$
	APM	n	$\log n$
	EPM	n	$\log^2 n$
Reporting	RAM	n	$\log n + k \log^{\epsilon}(2n/k)$
	RAM	$n \log\log n$	$\log n + k \log\log(4n/k)$
	RAM	$n \log^{\epsilon} n$	$\log n + k$
	APM	n	$k \log(2n/k)$
	EPM	n	$k \log^2(2n/k)$
	EPM	$\frac{n \log n}{\log\log n}$	$\log n + k$
Semigroup	arithmetic	m	$\frac{n \log n}{\log(2m/n)}$
	RAM	n	$\log^{2+\epsilon} n$
	RAM	$n \log\log n$	$\log^2 n \log\log n$
	RAM	$n \log^{\epsilon} n$	$\log^2 n$
	APM	n	$\log^3 n$
	EPM	n	$\log^4 n$

All the results mentioned in Table 31.1.1 can be extended to higher dimensions at a cost of a $\log^{d-2} n$ factor in the preprocessing time, storage, and query-search time. Table 31.1.2 summarizes a few additional results on higher-dimensional orthogonal range-searching results.

Overmars [Ove88] showed that if S is a subset of a $u \times u$ grid U in the plane and the vertices of query rectangles are also a subset of U, then a range-reporting query can be answered in time $O(\sqrt{\log u} + k)$, using $O(n \log n)$ storage and preprocessing; or in $O(\log\log u + k)$ time, using $O(n \log n)$ storage and $O(u^3 \log u)$ preprocessing.

Range-tree-based data structures for orthogonal range searching can be extended to handle c-oriented ranges. The performance of such a data structure is the same as that of a c-dimensional, orthogonal range-searching structure. If the ranges are homothets of a given triangle, or translates of a convex polygon with a constant number of edges, a two-dimensional range-reporting query can be answered in $O(\log n + k)$ time using linear space [CE87]. If the ranges are octants in $\mathbb{R}^3$, a range-reporting query can be answered in either $O((k+1)\log n)$ or $O(\log^2 n + k)$ time using linear space [CE87].

TABLE 31.1.2 Known upper bounds on higher-dimensional orthogonal range reporting.

$S(n)$	$Q(n)$	NOTES
$n \log^{d-1+\epsilon} n$	$\left(\frac{\log n}{\log \log n}\right)^{d-1} + k$	pointer machine
m	$\left(\frac{\log n}{\log 2m/n}\right)^{d-1}$	semigroup model
$\frac{n \log^{d-1} n}{\log \log n}$	$\frac{\log^{d-1} n}{\log \log n} + k$	fusion trees [Wil92]
$n \log^{d-1} n$	$\log^{d-2} n \log^* n + k$	P^*-trees [SR95]

LOWER BOUNDS

Fredman [Fre80, Fre81] was the first to prove nontrivial lower bounds on orthogonal range searching, but he considered the framework in which the points could be inserted and deleted dynamically. He showed that a mixed sequence of n insertions, deletions, and queries takes $\Omega(n \log^d n)$ time. These bounds were extended by Willard to a group model, under some restrictive assumptions. Chazelle proved lower bounds for the static version of orthogonal range searching, which almost match the best upper bounds known [Cha90b]. The following theorem summarizes his main result.

THEOREM 31.1.1 *Chazelle* [Cha90b]

Let $(\mathbf{S}, \oplus)$ be a faithful semigroup, let d be a constant, and let n and m be parameters. Then there exists a set S of n weighted points in $\mathbb{R}^d$, with weights from $\mathbf{S}$, such that the worst-case query time, under the semigroup model, for an orthogonal range-searching data structure that uses m units of storage is $\Omega((\log n/\log(2m/n))^{d-1})$.

Theorem 31.1.1 even holds for answering dominance queries. In fact, this lower bound applies to answering the dominance query for a randomly chosen query point; in this sense the above theorem gives a lower bound on the average-case complexity of the query time. It should pointed out that Theorem 31.1.1 assumes the weights of points in S to be a part of the input. That is, the data structure is not tailored to a special set of weights, and it should work for any set of weights. It is conceivable that a faster algorithm can be developed for answering orthogonal range-counting queries, exploiting the fact that the weight of each point is 1 in this case. None of the known algorithms are able to exploit this fact, however.

A rather surprising result of Chazelle [Cha90a] shows that the size of any data structure on a pointer machine that answers a d-dimensional range-reporting query in $O(\log^c n + k)$ time, for any constant c, is $\Omega(n(\log n/\log\log n)^{d-1})$. Notice that this lower bound is greater than the known upper bound on the RAM model (see Table 31.1.1).

These lower bounds do not hold for off-line orthogonal range searching, where given a set of n weighted points in $\mathbb{R}^d$ and a set of n rectangles, one wants to compute the weight of points in each rectangle. Recently, Chazelle [Cha95] proved that the off-line version takes $\Omega(n(\log n/\log\log n)^{d-1})$ time in the semigroup model and $\Omega(n \log\log n)$ time in the group model.

RELATED PROBLEMS

- *Partial-sum problem:* Preprocess a d-dimensional array A with n entries in an additive semigroup into a data structure so that, for a d-dimensional rectangle $q = [a_1, b_1] \times \cdots \times [a_d, b_d]$, the sum $\sigma(A, q) = \sum_{(k_1, \ldots, k_d) \in q} A[k_1, \ldots, k_q]$ can be computed efficiently. In the off-line version, given A and m rectangles $q_1, \ldots, q_m$, we wish to compute $\sigma(A, q_i)$ for every $i \leq m$. For any fixed $d \geq 1$, a partial-sum query can be answered in $O(\alpha(n) \log^{d-2} n)$ time using $O(n \log^{d-1} n)$ space [CR89]. Chazelle and Rosenberg [CR89] also showed that the off-line version of the partial-sum problem, even for $d = 1$, takes $\Omega(n + m\alpha(m, n))$ time; here $\alpha(m, n)$ is the inverse Ackerman function (Section 40.4). If points are allowed to be inserted into S, the query time is $\Omega((\log n / \log \log n)^d)$, for any fixed dimension d.
- *Rectangle-rectangle searching:* Preprocess a set S of n rectangles in $\mathbb{R}^d$ so that for a query rectangle q, the rectangles of S that intersect q can be reported (or counted) efficiently. The bounds mentioned in Table 31.1.1 hold for this problem also.

OPEN PROBLEMS

1. No nontrivial lower bounds are known for answering emptiness queries.
2. Chazelle's lower bound for range reporting on the pointer-machine model does not hold if the query time is allowed to be of the form $O((k + 1) \log^c n)$.
3. Find better lower bounds under the group model.
4. Find efficient data structures when d is "large," e.g., when $d \approx \log n$.

31.2 SIMPLEX RANGE SEARCHING

In the last few years, simplex range searching has received considerable attention because, apart from its direct applications, simplex range-searching data structures have provided fast algorithms for numerous other geometric problems.

Unlike orthogonal range searching, no simplex range-searching data structure is known that can answer a query in polylogarithmic time using near-linear storage. In fact, the lower bounds stated below indicate that there is very little hope of obtaining such a data structure, since the query time of a linear-size data structure, under the semigroup model, is roughly at least $n^{1-1/d}$ (thus only saving a factor of $n^{1/d}$ over the naive approach). Because the size and query time of any data structure have to be at least linear and logarithmic, respectively, we consider these two ends of the spectrum: (i) how fast a simplex range query can be answered using a linear-size data structure; and (ii) how large the size of a data structure should be in order to answer a query in logarithmic time. Combining these two extreme cases leads to a space/query-time tradeoff.

GLOSSARY

Range space: A range space is a set system $\Sigma = (X, R)$ where X is a set of objects and R is a family of subsets of X. The elements of R are called ranges of Σ. Σ is called a ***finite range space*** if X is finite. See Section 34.4 for examples.

ϵ-net: A subset $N \subseteq X$ is called an ϵ-net of a finite range space Σ if $N \cap r \neq \emptyset$ for every $r \in R$ with $|r| \geq \epsilon |X|$.

VC-dimension: The VC-dimension of a range space $\Sigma = (X, R)$ is d if there is no subset $A \subseteq X$ of size $d+1$ such that $\{A \cap r \mid r \in R\} = 2^A$; cf. Section 10.12.

Arrangements: The arrangement of a set H of hyperplanes in $\mathbb{R}^d$ is the subdivision of $\mathbb{R}^d$ into cells of dimension k, for $0 \leq k \leq d$, each cell of dimension $k < d$ being a maximal connected set contained in the intersection of a fixed subset of H and not intersecting any other hyperplane of H. See Chapter 21.

$1/r$-cutting: Let H be a set of n hyperplanes in $\mathbb{R}^d$ and let $1 \leq r \leq n$ be a parameter. A $(1/r)$-cutting of H is a set of (relatively open) disjoint simplices covering $\mathbb{R}^d$ so that each simplex intersects at most n/r hyperplanes of H.

Duality: The dual of a point $(a_1, \ldots, a_d) \in \mathbb{R}^d$ is the hyperplane $x_d = -a_1 x_1 - \cdots - a_{d-1} x_{d-1} + a_d$, and the dual of a hyperplane $x_d = b_1 x_1 + \cdots + b_d$ is the point $(b_1, \ldots, b_{d-1}, b_d)$.

Ham sandwich theorem: Let $P_1, P_2, \ldots, P_d$ be d finite sets of points in $\mathbb{R}^d$. There exists a hyperplane h that simultaneously bisects $P_1, P_2, \ldots, P_d$. See [Ede87], as well as Section 11.2.

TABLE 31.2.1 Lower bounds for simplex range searching.

RANGE	MODEL	$S(n)$	$Q(n)$
Simplex	semigroup ($d = 2$)	m	$\frac{n}{\sqrt{m}}$
Simplex	semigroup ($d > 2$)	m	$\frac{n}{m^{1/d} \log n}$
Simplex	group	$n \log n$	$\log n$
Simplex	pointer (reporting)	m	$\frac{n^{1-\epsilon}}{m^{1/d}} + k$
Halfspace	semigroup	m	$\left(\frac{n}{\log n}\right)^{\frac{d^2+1}{d^2+d}} \cdot \frac{1}{m^{1/d}}$

LOWER BOUNDS

In a series of papers, Chazelle proved nontrivial lower bounds on simplex range searching, using various elegant mathematical techniques; see Table 31.2.1. The following theorem is perhaps the most interesting result on lower bounds.

THEOREM 31.2.1 *Chazelle* [Cha89]

Let $(\mathbf{S}, \oplus)$ be a faithful semigroup, and let n and m be parameters. There exists a set S of n weighted points in $\mathbb{R}^d$, with weights from $\mathbf{S}$, such that the worst-case query

time of any simplex range-searching data structure, under the semigroup model, using m units of storage is $\Omega(n/\sqrt{m})$ for $d = 2$, and $\Omega(n/(m^{1/d} \log n))$ for $d \geq 3$.

The above theorem holds even if the query ranges are wedges or strips. Like Theorem 31.1.1, this theorem also assumes that the weights of S are a part of the input. Hence, it does not yield a lower bound for the simplex emptiness or for the simplex range-counting problem. As we will see below, faster data structures are known for answering halfspace emptiness queries.

LINEAR-SIZE DATA STRUCTURES

Most of the linear-size data structures for simplex range searching are based on ***partition trees***, originally introduced by Willard [Wil82]. His partition tree is a 4-way tree, constructed as follows. Let us assume that n is of the form 4^k for some integer k, and that the points of S are in general position. If $k = 0$, the tree consists of a single node that stores the coordinates of the only point in S. Otherwise, using the ham sandwich theorem, we find two lines ℓ_1, ℓ_2 such that each quadrant Q_i, for $1 \leq i \leq 4$, induced by ℓ_1, ℓ_2 contains exactly $n/4$ points. The root stores the equations of ℓ_1, ℓ_2 and the value of n. For each quadrant, we recursively construct a partition tree for $S \cap Q_i$ and attach it as the ith subtree of the root. The total size of the data structure is linear, and it can be constructed in $O(n \log n)$ time. A halfplane range-counting query can be answered as follows. Let h be a query halfplane. We traverse T, starting from the root, and maintain a global count. At each node v storing n_v nodes in its subtree, the algorithm performs the following step: If the line ∂h intersects the quadrant Q_v associated with v, we recursively visit the children of v. If $Q_v \cap h = \emptyset$, we do nothing. Otherwise, we add n_v to the global count. The quadrants associated with the four children of an interior node of T are induced by two lines, so ∂h intersects at most three quadrants, which implies that the query procedure does not explore the subtree of one of the children. Hence, the query time of this procedure is $O(n^{\log_4 3}) = O(n^{.792})$. A similar procedure can answer a simplex range-counting query within the same time bound, and a simplex range-reporting query in time $O(n^{.792} + k)$.

A breakthrough in simplex range searching was made by Haussler and Welzl [HW87]. They formulated range searching in an abstract setting and, using elegant probabilistic methods, gave a randomized algorithm to construct a linear-size partition tree with $O(n^\alpha)$ query time, where $\alpha = 1 - \frac{1}{d(d-1)+1} + \epsilon$ for any $\epsilon > 0$. The constant of proportionality hidden in the big-O notation depends on the value of ϵ. The major contribution of their paper is the abstract framework and the notion of ϵ-nets. The following theorem gives a slightly stronger version of their main result.

THEOREM 31.2.2 *Komlós et al.* [KPW92]

For any finite range space (X, R) of VC-dimension d and for $0 < \epsilon, \delta < 1$, if N is a subset of X obtained by

$$\frac{d}{\epsilon}\left(\log\frac{1}{\epsilon} + 2\log\log\frac{1}{\epsilon} + 3\right)$$

random independent draws, then N is an ϵ-net of (X, R) with probability at least $1 - e^{-\delta}$.

Theorem 31.2.2 implies that any finite range space of VC-dimension d has an

ϵ-net of size $(1 + o(1))(d/\epsilon) \log 1/\epsilon$. ϵ-nets have turned out to be a powerful tool in developing divide-and-conquer algorithms for several geometric problems, and in learning theory.

The best-known linear-size data structure for simplex range searching, which almost matches the lower bounds mentioned above, is by Matoušek [Mat93b]. He showed that a simplex range-counting (resp. range-reporting) query in $\mathbb{R}^d$ can be answered in time $O(n^{1-1/d})$ (resp. $O(n^{1-1/d} + k)$). His algorithm is based on a stronger version of the following theorem.

THEOREM 31.2.3 *Matoušek* [Mat92]
Let S be a set of n points in $\mathbb{R}^d$, and let $1 < r \leq n/2$ be a given parameter. Then there exists a family of pairs

$$\Pi = \{(S_1, \Delta_1), \ldots, (S_m, \Delta_m)\}$$

such that $S_i \subseteq S$ lies inside simplex Δ_i, $n/r \leq |S_i| \leq 2n/r$, $S_i \cap S_j = \emptyset$ for $i \neq j$, and every hyperplane crosses at most $cr^{1-1/d}$ simplices of Π; here c is a constant. If r is a constant, then Π can be constructed in $O(n)$ time.

Using this theorem, a partition tree T can be constructed as follows. Each interior node v of T is associated with a subset $S_v \subseteq S$ and a simplex Δ_v containing S_v; the root of T is associated with S and $\mathbb{R}^d$. Choose r to be a sufficiently large constant. If $|S| \leq 4r$, T consists of a single node, and it stores all points of S. Otherwise, we construct a family of pairs $\Pi = \{(S_1, \Delta_1), \ldots, (S_m, \Delta_m)\}$ using Theorem 31.2.3. The root u stores the value of n. We recursively construct a partition tree T_i for each S_i and attach T_i as the ith subtree of u. The root of T_i also stores Δ_i. The total size of the data structure is linear, and it can be constructed in time $O(n \log n)$. A simplex range-counting query can be answered in a manner similar to that used for Willard's partition tree. Since any hyperplane intersects at most $cr^{1-1/d}$ simplices of Π, the query time is $O(n^{1-1/d} \cdot n^{\log_r c})$; the $\log_r c$ factor can be reduced to any arbitrarily small positive constant ϵ by choosing r sufficiently large. Although the query time can be improved to $O(n^{1-1/d} \log^c n)$ by choosing r to be n^ϵ, a stronger version of Theorem 31.2.3, which was proved in [Mat93b], and some other sophisticated techniques are needed to obtain $O(n^{1-1/d})$ query time.

If the points in S lie on a k-dimensional algebraic surface of constant degree, a simplex range-counting query can be answered in time $O(n^{1-\gamma})$ using linear space, where $\gamma = 1/\lfloor (d+k)/2 \rfloor$.

TABLE 31.2.2 Halfspace range searching.

d	$S(n)$	$Q(n)$	NOTES
$d = 2$	n	$\log n + k$	reporting
$d = 3$	$n \log n$	$\log n + k$	reporting
	n	$\log n$	emptiness
$d > 3$	$n \log \log n$	$n^{1-1/\lfloor d/2 \rfloor} \log^c n$	reporting
	n	$n^{1-1/d} 2^{O(\log^* n)}$	emptiness

Because the query time of a linear-size simplex range-searching data structure is only an $n^{1/d}$ factor faster than the naive method, and all the known data structures are rather intricate, researchers have developed practical data structures that work well most of the time. For example, Arya and Mount [AM95b] have developed a linear-size data structure for answering approximate range-counting queries, approximate in the sense that the points lying within distance $\delta \cdot \text{diam}(\Delta)$ of the boundary of the query simplex Δ may or may not be counted. Its query time is $O(\log n + 1/\delta^{d-1})$. Other practical data structures include k-d trees and quad trees [Ede87].

We conclude this subsection by noting that better bounds can be obtained for halfspace range reporting, using *filtering search*; see Table 31.2.2.

DATA STRUCTURES WITH LOGARITHMIC QUERY TIME

For the sake of simplicity, we first consider the halfspace range-counting problem. Using a standard duality transform, this problem can be reduced to the following: *Given a set H of n hyperplanes, determine the number of hyperplanes of H lying above a query point.* Since the same subset of hyperplanes lies above all points in a single cell of $A(H)$, the arrangement of H, we can answer a halfspace range-counting query by locating the cell of $A(H)$ that contains the point dual to the hyperplane bounding the query halfspace. The following theorem of Chazelle [Cha93] yields an $O((n/\log n)^d)$-size data structure, with $O(\log n)$ query time, for halfspace range counting.

THEOREM 31.2.4 *Chazelle* [Cha93]

Let H be a set of n hyperplanes and $r \leq n$ a parameter. Set $k = \lceil \log_2 r \rceil$. There exist k cuttings $\Xi_1, \ldots, \Xi_k$ so that Ξ_i is a $(1/2^i)$-cutting of size $O(2^{id})$, each simplex of Ξ_i is contained in a simplex of Ξ_{i-1}, and each simplex of Ξ_{i-1} contains a constant number of simplices of Ξ_i. Moreover, $\Xi_1, \ldots, \Xi_k$ can be computed in time $O(nr^{d-1})$.

The above approach can be extended to the simplex range-counting problem as well. That is, store the solution of every combinatorially distinct simplex (two simplices are combinatorially distinct if they do not contain the same subset of S). Since there are $\Theta(n^{d(d+1)})$ combinatorially distinct simplices, such an approach will require $\Omega(n^{d(d+1)})$ storage. Chazelle et al. [CSW92] showed that the size can be reduced to $O(n^{d+\epsilon})$, for any $\epsilon > 0$, using a multi-level data structure, with each level composed of a halfspace range-counting data structure. The space bound can be reduced to $O(n^d)$ by increasing the query time to $O(\log^{d+1} n)$ [Mat93b]. Halfspace range-reporting queries can be answered in $O(\log n + k)$ time, using $O(n^{\lfloor d/2 \rfloor + \epsilon})$ space.

A space/query-time tradeoff can be attained by combining the linear-size and logarithmic query-time data structures. The known results on this tradeoff are summarized in Table 31.2.3.

OPEN PROBLEMS

1. Bridge the gap between the known upper and lower bounds in the group model. Even in the semigroup model there is a small gap between the known bounds.

TABLE 31.2.3 Space/query-time tradeoff.

RANGE	MODE	$Q(m,n)$
Simplex	reporting	$\frac{n}{m^{1/d}} \log^{d+1} \frac{m}{n} + k$
Simplex	counting	$\frac{n}{m^{1/d}} \log^{d+1} \frac{m}{n}$
Halfspace	reporting	$\frac{n}{m^{1/\lfloor d/2 \rfloor}} \log^{c} n + k$
Halfspace	emptiness	$\frac{n}{m^{1/\lfloor d/2 \rfloor}} \log^{c} n$
Halfspace	counting	$\frac{n}{m^{1/d}} \log \frac{m}{n}$

2. Can a halfspace range-reporting query be answered in $O(n^{1-1/\lfloor d/2 \rfloor} + k)$ time using linear space?
3. Find simpler range-searching data structures for the cases in which points satisfy certain properties, e.g., when the points are uniformly distributed.

31.3 VARIANTS AND EXTENSIONS

In this section we review some extensions of range-searching data structures, including semialgebraic range searching, dynamization, and secondary-memory data structures.

GLOSSARY

Semialgebraic set: A subset of $\mathbb{R}^d$ obtained as a finite Boolean combination of sets of the form $\{f \geq 0\}$, where f is a d-variate polynomial (see Chapter 29).

Tarski cells: A real semialgebraic set defined by a constant number of polynomials, each of constant degree.

SEMIALGEBRAIC RANGE SEARCHING

So far we have assumed that the ranges were bounded by hyperplanes, but in many applications one has to deal with ranges bounded by nonlinear functions. For example, a query of the form, "for a given point p and a real number r, find all points of S lying within distance r from p," is a range-searching problem in which the ranges are balls.

As shown below, ball range searching in $\mathbb{R}^d$ can be formulated as an instance of the halfspace range searching in $\mathbb{R}^{d+1}$. So a ball range-reporting (resp. range-counting) query in $\mathbb{R}^d$ can be answered in time $O(n/m^{1/\lceil d/2 \rceil} \log^c n + k)$ (resp. $O(n/m^{1/(d+1)} \log(m/n))$), using $O(m)$ space; somewhat better performance can be obtained using a more direct approach (Table 31.3.1). However, relatively little is known about range-searching data structures for more general ranges.

A natural class of more general ranges is the family of Tarski cells. It suffices to consider the ranges bounded by a single polynomial because the ranges bounded by multiple polynomials can be handled using multi-level data structures. We assume that the ranges are of the form

$$\Gamma_f(a) = \{x \in \mathbb{R}^d \mid f(x,a) \geq 0\},$$

where f is a $(d+p)$-variate polynomial specifying the type of ranges (disks, cylinders, cones, etc.), and a is a p-tuple specifying a specific range of the given type (e.g., a specific disk). We will refer to the range-searching problem in which the ranges are from the set Γ_f as ***Γ_f-range searching.***

TABLE 31.3.1 Semialgebraic range counting.

d	RANGE	$S(n)$	$Q(n)$	NOTES
$d=2$	disk	$n \log n$	$\sqrt{n \log n}$	
	Tarski cell	n	$n^{1/2+\epsilon}$	partition tree
$d \geq 3$	Tarski cell	n	$n^{1-\frac{1}{2d-3}+\epsilon}$	partition tree
	Tarski cell	n	$n^{1-\frac{1}{\lambda}+\epsilon}$	linearization

One approach to answering Γ_f-range queries is ***linearization.*** We represent the polynomial $f(x,a)$ in the form

$$f(x,a) = \psi_0(a) + \psi_1(a)\varphi_1(x) + \cdots + \psi_k(a)\varphi_k(x)$$

where $\varphi_1, \ldots, \varphi_k, \psi_0, \ldots, \psi_k$ are real functions. A point $x \in \mathbb{R}^d$ is mapped to the point

$$\varphi(x) = (\varphi_1(x), \varphi_2(x), \ldots, \varphi_k(x)) \in \mathbb{R}^k.$$

Then a range $\gamma_f(a) = \{x \in \mathbb{R}^d \mid f(x,a) \geq 0\}$ is mapped to a halfspace

$$\varphi^{\#}(a) : \{y \in \mathbb{R}^k \mid \psi_0(a) + \psi_1(a)y_1 + \cdots + \psi_k(a)y_k \geq 0\};$$

k is called the ***dimension*** of linearization. Agarwal and Matoušek [AM94] have described an algorithm for computing a linearization of smallest dimension. A Γ_f-range query can now be answered using a k-dimensional halfspace range-searching data structure.

For example, a circle with center (a_1, a_2) and radius a_3 in the plane can be regarded as a set of the form $\gamma_f(a)$, where $a = (a_1, a_2, a_3)$ and f is a 5-variate polynomial that can be written as

$$f(x_1, x_2, a_1, a_2, a_3) = [a_3^2 - a_1^2 - a_2^2] + [2a_1x_1] + [2a_2x_2] - [x_1^2 + x_2^2].$$

Thus, setting

$$\begin{array}{llll} \psi_0(a) = a_3^2 - a_1^2 - a_2^2, & \psi_1(a) = 2a_1, & \psi_2(a) = 2a_2, & \psi_3(a) = -1 \\ & \varphi_1(x) = x_1, & \varphi_2(x) = x_2, & \varphi_3(x) = x_1^2 + x_2^2, \end{array}$$

we get a linearization of dimension 3. In general, balls in $\mathbb{R}^d$ admit a linearization of dimension $d+1$; cylinders in $\mathbb{R}^3$ admit a linearization of dimension 9. One of

the most widely used linearizations in computational geometry is based on *Plücker coordinates*, which map a line in $\mathbb{R}^3$ to a point in $\mathbb{R}^5$ (see Section 32.1).

Agarwal and Matoušek [AM94] have also proposed another approach to answer Γ_f-range queries, by extending Theorem 31.2.3 to Tarski cells and by constructing partition trees using this extension. Table 31.3.1 summarizes the known results on Γ_f-range-counting queries. In the table, λ is the dimension of linearization.

DYNAMIZATION

All the data structures discussed above assume that S is fixed, but in many applications one needs to update S dynamically—insert a new point into S or delete a point from S. One cannot hope to perform insert/delete operations on a data structure in less than $P(n)/n$ time, where $P(n)$ is the preprocessing time of the data structure. If we allow only insertions (i.e., a point cannot be deleted from the structure), the static data structure can be modified, using standard optimization techniques [BS80, Ove83], so that a point can be inserted in time $O(P(n)\log n/n)$ and a query can be answered in time $O(Q(n)\log n)$, where $Q(n)$ is the query time of the original static data structure; in some cases the logarithmic overhead in the query or update time can be avoided. Although these techniques do not extend to deletions, many range-searching data structures, such as orthogonal and simplex range-searching structures, can handle deletions with polylogarithmic or n^ϵ overhead in query and update time, by exploiting the fact that a point is stored at roughly $S(n)/n$ nodes [AS93]. Table 31.3.2 summarizes the known results on dynamic 2D orthogonal range-searching data structures; these results can be extended to higher dimensions at a cost of $\log^{d-2} n$ factor in the storage, in the query time, and in the update time.

TABLE 31.3.2 Dynamic 2D orthogonal range searching.

MODE	$S(n)$	$Q(n)$	$U(n)$
Counting	n	$\log^2 n$	$\log^2 n$
Counting	n	$k\log^2(2n/k)$	$\log^2 n$
Counting	n	$n^\epsilon + k$	$\log^2 n$
Reporting	$n\log n$	$\log n \log\log n + k$	$\log n \log\log n$
Reporting	$\dfrac{n\log n}{\log\log n}$	$\dfrac{\log^{2+\epsilon} n}{\log\log n} + k$	$\dfrac{\log^2 n}{\log\log n}$
Semigroup	n	$\log^4 n$	$\log^4 n$

Because an arbitrary sequence of deletions is difficult to handle in general, researchers have examined whether a random sequence of insertions and deletions can be handled efficiently. Mulmuley [Mul91] has shown that there exists a dynamic halfspace range-reporting data structure that can process a random update sequence of length m in expected time $O(m^{\lfloor d/2\rfloor+\epsilon})$, and can answer a halfspace range-reporting query in time $O(k\log n)$. Agarwal and Matoušek [AM95a] developed a dynamic data structure for halfspace range reporting that can process an arbitrary update sequence efficiently; its update time is $O(n^{\lfloor d/2\rfloor-1+\epsilon})$, and it can

answer a query in time $O(\log n + k)$. If we allow only $O(n \log n)$ space, then the query and update time become $O(n^{1-1/\lfloor d/2 \rfloor + \epsilon} + k)$ and $O(\log n)$, respectively.

SECONDARY-MEMORY STRUCTURES

If the input point set is rather large and does not fit into main memory, then the data structure is stored in secondary memory, and portions of it are moved to the main memory as required. In this case the bottleneck is the time spent in transferring the data between main and secondary memory. We assume that the data are stored in the secondary memory in blocks of size B, where B is a parameter. Each access to the secondary memory transfers one block (i.e., B words), and we count this as one I/O operation. The size of a data structure is the number of blocks required to store it, and the query (resp. preprocessing) time is defined as the number of I/O operations required to answer a query (resp. to construct the structure). I/O-efficient orthogonal range-searching structures have received much attention recently, but most of the results are only known for $d \leq 3$ [SR95, VV96].

Table 31.3.3 summarizes the known results on secondary-memory structures for orthogonal range searching; here $\beta(n) = \log \log \log_B n$. Extending the lower bounds by Chazelle for orthogonal range searching, Subramanian and Ramaswamy [SR95] proved that any secondary-memory data structure that answers a range-reporting query in time $O(\log_B^c n + k/B)$ requires $\Omega((n/B) \log(n/B)/ \log \log_B n)$ storage.

TABLE 31.3.3 Secondary-memory structures for orthogonal range searching.

d	RANGE	$Q(n)$	$S(n)$
$d = 1$	interval	$\log_B n + k/B$	n/B
$d = 2$	quadrant	$\log_B n + k/B$	$(n/B) \log \log B$
	3-sided rect	$\log_B n + k/B + \log^* B$	n/B
	3-sided rect	$\log_B n + k/B$	$(n/B) \log B \log \log B$
	rectangle	$\log_B n + k/B + \log^* B$	$(n/B) \log(n/B)/ \log \log_B n$
$d = 3$	octant	$\beta(n, B) \log_B n + k/B$	$(n/B) \log(n/B)$
	rectangle	$\beta(n, B) \log_B n + k/B$	$(n/B) \log^4(n/B)$

OPEN PROBLEMS

1. Can a ball range-counting query be answered in $O(\log n)$ time using $O(n^2)$ space?
2. Can a Γ_f-range-counting query be answered in time $O(n^{1-1/d+\epsilon})$ using near-linear space?
3. A solution to the following problem, which is interesting in its own right, would result in a better semialgebraic range-searching data structure: Given

a set Γ of n algebraic surfaces in $\mathbb{R}^d$, each of constant maximum degree, decompose the cells of the arrangement of Γ into $O(n^d)$ Tarski cells.

4. If the hyperplanes bounding the query halfspaces satisfy some property—e.g., all of them are tangent to a given sphere—can a halfspace range-counting query be answered more efficiently?

5. Efficient secondary-memory structures for higher-dimensional orthogonal range searching and for simplex range searching are lacking.

6. Find efficient range-searching data structures for a set of moving points.

31.4 INTERSECTION SEARCHING

A general intersection-searching problem can be formulated as follows: *Given a set S of objects in $\mathbb{R}^d$, a semigroup $(\mathbf{S}, +)$, and a weight function $w : S \to \mathbf{S}$; we wish to preprocess S into a data structure so that for a query object γ, we can compute the weighted sum $\sum w(p)$, where the sum is taken over all objects of S that intersect γ.* Range searching is a special case of intersection-searching in which S is a set of points.

An intersection-searching problem can be formulated as a semialgebraic range-searching problem by mapping each object $p \in S$ to a point $\varphi(p)$ in a parametric space $\mathbb{R}^k$ and every query range γ to a semialgebraic set $\psi(\gamma)$ so that p intersects γ if and only if $\varphi(p) \in \psi(\gamma)$. For example, let both S and the query ranges be sets of segments in the plane. Each segment $e \in S$ with left and right endpoints (p_x, p_y) and (q_x, q_y), respectively, can be mapped to a point $\varphi(e) = (p_x, p_y, q_x, q_y)$ in $\mathbb{R}^4$, and a query segment γ can be mapped to a semialgebraic region $\psi(\gamma)$ so that γ intersects e if and only if $\psi(\gamma) \in \varphi(e)$. A shortcoming of this approach is that k, the dimension of the parametric space, is typically much larger than d, and that therefore this does not lead to an efficient data structure. The efficiency can be significantly improved by expressing the intersection test as a conjunction of simple primitive tests (in low dimensions), and then using a multi-level data structure to perform these tests. For example, a segment γ intersects another segment e if the endpoints of e lie on the opposite sides of the line containing γ and vice-versa. We can construct a two-level data structure—the first level sifts the subset $S_1 \subseteq S$ of all the segments that intersect the line supporting the query segment, and the second level reports those segments of S_1 whose supporting lines separate the endpoints of γ. Each level of this structure can be implemented using a two-dimensional simplex range-searching searching structure, and hence a reporting query can be answered in $O(n/\sqrt{m}\log^c n + k)$ time using $O(m)$ space.

It is beyond the scope of this chapter to cover all intersection-searching problems. Instead, we discuss a selection of basic problems that have been studied extensively. All intersection-counting data structures described here can answer intersection-reporting queries at an additional cost proportional to the output size. In some cases an intersection-reporting query can be answered faster. Moreover, using intersection-reporting data structures, intersection-detection queries can be answered in time proportional to their query-search time. Finally, all the data structures described in this section can be dynamized at an expense of an $O(n^\epsilon)$ factor in the storage and query time.

POINT INTERSECTION SEARCHING

Preprocess a set S of objects (e.g., balls, halfspaces, simplices, Tarski cells) in $\mathbb{R}^d$ into a data structure so that all the objects of S containing a query point can be reported (or counted) efficiently. This is the inverse of the range-searching problem, and it can also be viewed as locating a point in the subdivision induced by the objects in S. Table 31.4.1 gives some of the known results.

TABLE 31.4.1 Point intersection searching.

d	OBJECTS	$S(n)$	$Q(n)$	NOTES
$d=2$	disks	m	$(n/\sqrt{m})^{4/3}$	counting
	disks	$n \log n$	$\log n + k$	reporting
	triangles	m	$\frac{n}{\sqrt{m}} \log^3 n$	counting
	fat triangles	$n \log^2 n$	$\log^3 n + k$	reporting
	Tarski cells	$n^{2+\epsilon}$	$\log n$	counting
$d=3$	functions	$n^{1+\epsilon}$	$\log n + k$	reporting
$d \geq 3$	simplices	m	$\frac{n}{m^{1/d}} \log^{d+1} n$	counting
	balls	$n^{d+\epsilon}$	$\log n$	counting
	balls	m	$\frac{n}{m^{1/\lceil d/2 \rceil}} \log^c n + k$	reporting
	Tarski cells	$n^{2d-3+\epsilon}$	$\log n$	counting

Point location in arrangements of surfaces, especially determining whether a query point lies above a given set of regions of the form $x_{d+1} \geq f(x_1, \ldots, x_d)$, has found many applications in computational geometry. For example, the ***collision-detection*** problem—given a set O of obstacles and a robot B, determine whether a placement p of B is free (Chapter 40)—can be formulated as a point intersection-detection query amid a set of regions. If B has k degrees of freedom, then a placement of B can be represented as a point in $\mathbb{R}^k$, and the set of placements of B that intersect an obstacle $O_i \in m$ is a region $K_i \subseteq \mathbb{R}^k$. If B and the obstacles are semialgebraic sets, then each K_i is also a semialgebraic set. A placement p of B is ***free*** if and only if p does not intersect any of the K_i's. Hence, a collision-detection query can be answered by preprocessing the K_i's for answering point intersection-detection queries.

SEGMENT INTERSECTION SEARCHING

Preprocess a set of objects in $\mathbb{R}^d$ into a data structure so that all the objects of S intersected by a query segment can be reported (or counted) efficiently. See Table 31.4.2 for some of the known results on segment intersection searching. For the sake of clarity, we have omitted polylogarithmic factors from the query-search time whenever it is of the form n/m^{α}.

A special case of segment intersection searching, in which the objects are horizontal segments in the plane and query ranges are vertical segments, has been

TABLE 31.4.2 Segment intersection searching.

d	OBJECTS	$S(n)$	$Q(n)$	NOTES
$d=2$	simple polygon	n	$(k+1)\log n$	reporting
	segments	m	$n/\sqrt{m}$	counting
	circles	$n^{2+\epsilon}$	$\log n$	counting
	circular arcs	m	$n/m^{1/3}$	counting
$d=3$	planes	m	$n/m^{1/3}$	counting
	triangles	m	$n/m^{1/4}$	counting
	spheres	m	$n/m^{1/4}$	counting
	spheres	$n^{3+\epsilon}$	$(k+1)\log^2 n$	reporting

widely studied. In this case a query can be answered in time $O(\log n + k)$ using $O(n \log n)$ space and preprocessing. If we also allow insertions and deletions, the query and update time are $O(\log n \log\log n + k)$ and $O(\log n \log\log n)$ [MN90]; if we allow only insertions, the query and update time become $O(\log n + k)$ and $O(\log n)$ [IA87].

A problem related to segment intersection searching is the ***stabbing problem.*** *Given a set S of objects in $\mathbb{R}^d$, determine whether a query k-flat ($0 < k < d$) intersects all objects of S.* Such queries can also be answered efficiently using semialgebraic range-searching data structures. A line-stabbing query amid a set of triangles in $\mathbb{R}^3$ can be answered in $O(\log n)$ time using $O(n^{2+\epsilon})$ storage. See Section 32.2 for details.

COLORED INTERSECTION SEARCHING

Preprocess a given set S of colored objects in $\mathbb{R}^d$ (i.e., each object in S is assigned a color) so that we can report (or count) the colors of the objects that intersect the query range. This problem arises in many contexts where one wants to answer intersection-searching queries for nonconstant-size input objects. For example, given a set $P = \{P_1, \ldots, P_m\}$ of m simple polygons, one may wish to report all the simple polygons that intersect a query segment; the goal is to return the indices, and not the description, of these polygons. If we color the edges of P_i with color i, the problem reduces to colored segment intersection searching in a set of segments.

If an intersection-detection query for S with respect to a range γ can be answered in $Q(n)$ time, then the colored intersection-reporting query with γ can be answered in time $O((1 + k\log(n/k))Q(n))$. Therefore intersection-searching data structures with logarithmic query-time can easily be modified for colored intersection reporting, but very little is known about linear-size colored intersection-searching data structures, except in a few special cases.

Gupta et al. [GJS94] have shown that the colored halfplane-reporting queries in the plane can be answered in $O(\log^2 n + k)$ using $O(n \log n)$ space. Agarwal and van Kreveld [AvK93] presented a linear-size data structure with $O(n^{1/2+\epsilon} + k)$ query time for colored segment intersection-reporting queries amid a set of segments in the plane, assuming that the segments of the same color form a connected planar graph, or if they form the boundary of a simple polygon; these data structures can also handle insertions of new segments. Gupta et al. [GJS94, GJS95] present

segment intersection-reporting structures for many other special cases.

OPEN PROBLEMS

1. Find faster algorithms for point-intersection searching in Tarski cells.
2. Design faster segment intersection-detection structures for (possibly intersecting) Jordan arcs in the plane, and for triangles and spheres in $\mathbb{R}^3$.
3. Find a linear-size, $O(\sqrt{n}\log^c n + k)$ query-time data structure for colored triangle range reporting.

31.5 OPTIMIZATION QUERIES

In optimization queries, we want to return an object that satisfies certain conditions with respect to a query range. The most common example of optimization queries is, perhaps, ray-shooting queries. Other examples include segment-dragging and linear-programming queries.

RAY-SHOOTING QUERIES

Preprocess a set S of objects in $\mathbb{R}^d$ into a data structure so that the first object (if one exists) intersected by a query ray can be reported efficiently. This problem arises in ray tracing, hidden-surface removal, radiosity, and other graphics problems. Recently, efficient solutions to many other geometric problems have also been developed using ray-shooting data structures.

A general approach to the ray-shooting problem, using segment intersection-detection structures and Megiddo's parametric-searching technique (Chapter 37), was proposed by Agarwal and Matoušek [AM93]. The basic idea of their approach is as follows. Suppose we have a segment intersection-detection data structure for S, based on partition trees. Let ρ be a query ray. Their algorithm maintains a segment $\vec{ab} \subseteq \rho$ so that the first intersection point of $\vec{ab}$ with S is the same as that of ρ. If a lies on an object of S, it returns a. Otherwise, it picks a point $c \in ab$ and determines, using the segment intersection-detection data structure, whether the interior of the segment ac intersects any object of S. If the answer is YES, it recursively finds the first intersection point of $\vec{ac}$ with S; otherwise, it recursively finds the first intersection point of $\vec{cb}$ with S. Using parametric searching, the points c at each stage can be chosen in such a way that the algorithm terminates after $O(\log n)$ steps.

In some cases the query time can be improved by a polylogarithmic factor using a more direct approach.

Table 31.5.1 gives a summary of known ray-shooting results. For the sake of clarity, we have ignored the polylogarithmic factors in the query time whenever it is of the form n/m^{α}. The ray-shooting structures for d-dimensional convex polyhedra assume that the source point of the query ray lies inside the polytope. All the ray-shooting data structures mentioned in Table 31.5.1 can be dynamized at a cost of a

TABLE 31.5.1 Ray shooting.

d	OBJECTS	$S(n)$	$Q(n)$
$d = 2$	simple polygon	n	$\log n$
	s disjoint simple polygons	n	$\sqrt{s}$
		$(s^2 + n) \log s$	$\log s \log n$
	s convex polygons	$sn \log s$	$\log s \log n$
	segments	m	$n/\sqrt{m}$
	circlular arcs	n	$n/m^{1/3}$
	disjoint arcs	n	$\sqrt{n}$
$d = 3$	convex polytope	n	$\log n$
	c-oriented polytopes	n	$\log n$
	s convex polytopes	$s^2 n^{2+\epsilon}$	$\log^2 n$
	halfplanes	m	$n/\sqrt{m}$
	terrain	m	$n/\sqrt{m}$
	triangles	m	$n/m^{1/4}$
	spheres	$n^{3+\epsilon}$	$\log^2 n$
$d > 3$	hyperplanes	m	$n/m^{1/d}$
	hyperplanes	$\frac{n^d}{\log^{d-\epsilon} n}$	$\log n$
	convex polytope	m	$n/m^{1/\lfloor d/2 \rfloor}$
	convex polytope	$\frac{n^{\lfloor d/2 \rfloor}}{\log^{\lfloor d/2 \rfloor - \epsilon} n}$	$\log n$

polylogarithmic or n^ϵ factor in the query time. Goodrich and Tamassia [GT93] have developed a dynamic ray-shooting data structure for connected planar subdivisions, with $O(\log^2 n)$ query and update time.

Like simplex range searching, many practical data structures have been proposed that, despite having poor worst-case performance, work well in practice. One common approach is to construct a subdivision of $\mathbb{R}^d$ into constant-size cells so that the interior of each cell does not intersect any object of S. A ray-shooting query can be answered by traversing the query ray through the subdivision until we find an object that intersects the ray. The worst-case query time is proportional to the maximum number of cells intersected by a segment that does not intersect any object in S. Hershberger and Suri [HS95] showed that a triangulation with $O(\log n)$ query time can be constructed when S is the boundary of a simple polygon in the plane. Agarwal et al. [AAS95] proved worst-case bounds for many cases on the number of cells in the subdivision that a line can intersect. Practical data structures for ray shooting, based on k-d trees, quad trees, and binary space partitions, have been developed. See the book by Glassner [Gla89] for a survey of such data structures.

We next explain how the nearest-neighbor searching problem can be formulated as an instance of a ray-shooting problem. The ***nearest-neighbor searching*** problem is defined as follows: *Preprocess a set S of points in $\mathbb{R}^d$ into a data structure so that a point in S closest to a query point ξ can be reported quickly.* We map each point $p = (p_1, \ldots, p_d)$ in S to a hyperplane in $\mathbb{R}^{d+1}$ that is the graph of the function

$$f_p(x_1, \ldots, x_n) = 2p_1x_1 + \cdots + 2p_dx_d - (p_1^2 + \cdots + p_d^2).$$

Then p is a closest neighbor of a point $\xi = (\xi_1, \ldots, \xi_d)$ if and only if

$$f_p(\xi_1, \ldots, \xi_d) = \max_{q \in S} f_q(\xi_1, \ldots, \xi_d).$$

That is, if and only if f_p is the first hyperplane intersected by the vertical ray $\rho(\xi)$ emanating from the point $(\xi_1, \ldots, \xi_d, 0)$ in the $(-x_{d+1})$-direction. If we define $P = \bigcap\{x_{d+1} \geq f_p(x_1, \ldots, x_d) \mid p \in S\}$, then p is the nearest neighbor of ξ if and only if the (first) intersection point of $\rho(\xi)$ and ∂P lies on the graph of f_p. Thus a nearest-neighbor query can be answered in time roughly $n/m^{1/\lfloor d/2 \rfloor}$ using $O(m)$ space. This approach can be extended to answer farthest-neighbor and k-nearest-neighbor queries also. For applications, see Chapter 43.

LINEAR-PROGRAMMING QUERIES

Let S be a set of n halfspaces in $\mathbb{R}^d$. *We wish to preprocess S into a data structure so that for a direction vector $\vec{v}$, we can determine the first point of $\bigcap_{h \in S} h$ in the direction $\vec{v}$.* For $d \leq 3$, such a query can be answered in $O(\log n)$ time using $O(n)$ storage, by constructing the normal diagram of the convex polytope $\bigcap_{h \in S} h$ and preprocessing it for point-location queries. For higher dimensions, Matoušek [Mat93a] showed that, using multidimensional parametric searching and a data structure for answering halfspace emptiness queries, a linear-programming query can be answered in $O((n/m^{1/\lfloor d/2 \rfloor}) \log^c n)$ with $O(m)$ storage. Recently Chan [Cha96] has described a randomized procedure whose expected query time is slightly faster.

SEGMENT-DRAGGING QUERIES

Preprocess a set S of objects in the plane so that for a query segment e and a ray ρ, the first position at which e intersects any object of S as it is translated (dragged) along ρ can be determined quickly. This query can be answered in $O((n/\sqrt{m}) \log^c n)$ time, with $O(m)$ storage, using segment intersection-searching structures and the parametric-search technique. Chazelle [Cha88a] gave a linear-size, $O(\log n)$ query-time data structure for the special case in which S is a set of points, e is a horizontal segment, and ρ is the vertical direction. Instead of dragging a segment along a ray, one can ask the same question for dragging along a more complex trajectory (along a curve, and allowing both translation and rotation). These problems arise naturally in motion planning and manufacturing.

OPEN PROBLEMS

1. Ray shooting amid a set of intersecting arcs in the plane is relatively unexplored.
2. Can a ray-shooting query in a nonconvex polytope in $\mathbb{R}^3$ be answered any faster than a ray-shooting query amid triangles?
3. Find nontrivial lower bounds for the ray-shooting problem.
4. Design efficient data structures for arc-shooting queries, where one wants to find the first object intersected by an oriented curve.

31.6 SOURCES AND RELATED MATERIAL

RELATED READING

Books and Monographs

[Meh84]: A textbook on computational geometry, whose first part covers multidimensional searching.

[Mul94]: A textbook on randomized techniques in computational geometry. Chapters 6 and 8 cover range-searching, intersection-searching, and ray-shooting data structures.

[PS85]: A textbook on basic topics in computational geometry. Chapter 2 includes earlier results on orthogonal range searching.

[FvD+90]: A textbook on graphics. Discusses practical data structures for ray tracing and intersection searching.

[dB93]: A monograph on ray shooting and related problems.

[Sch92]: This Ph.D. thesis includes many results on randomized dynamic data structures.

Survey Papers

[BF79]: A survey of earlier results on orthogonal range searching.

[Cha94a]: A general survey of recent developments in computational geometry. It contains most of the references on simplex and semialgebraic range searching.

[CT92]: A survey of dynamic data structures.

[GPW93]: A survey of stabbing problems and related topics.

[Mat94]: A comprehensive survey of simplex range searching and related topics.

Technical Articles

[AM95a]: Data structures for dynamic halfspace range searching and related problems.

[BCP93]: Lower bounds for halfspace range searching under the semigroup model.

[Cha94b]: Lower bounds for simplex range searching under the group model.

[Mat93b]: Data structures for simplex range searching and an abstract framework for multilevel data structures.

[MS93]: Data structures for halfspace-emptiness queries and for ray-shooting queries in a convex polytope.

RELATED CHAPTERS

Chapter 4: Helly-type theorems and geometric transversals
Chapter 10: Geometric discrepancy theory and uniform distribution

Chapter 21: Arrangements
Chapter 29: Computational real algebraic geometry
Chapter 30: Point location
Chapter 32: Ray shooting and lines in space
Chapter 34: Randomized algorithms
Chapter 37: Parametric search
Chapter 40: Algorithmic motion planning

REFERENCES

[AAS95] P.K. Agarwal, B. Aronov, and S. Suri. Line stabbing bounds in three dimensions. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 267–276, 1995.

[AM93] P.K. Agarwal and J. Matoušek. Ray shooting and parametric search. *SIAM J. Comput.*, 22:794–806, 1993.

[AM94] P.K. Agarwal and J. Matoušek. On range searching with semialgebraic sets. *Discrete Comput. Geom.*, 11:393–418, 1994.

[AM95a] P.K. Agarwal and J. Matoušek. Dynamic half-space range reporting and its applications. *Algorithmica*, 13:325–345, 1995.

[AM95b] S. Arya and D.M. Mount. Approximate range searching. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 172–181, 1995.

[AS93] P.K. Agarwal and M. Sharir. Applications of a new partition scheme. *Discrete Comput. Geom.*, 9:11–38, 1993.

[AvK93] P.K. Agarwal and M. van Kreveld. Connected component and simple polygon intersection searching. In *Proc. 3rd Workshop Algorithms Data Struct.*, volume 709 of *Lecture Notes in Comput. Sci.*, pages 36–47. Springer-Verlag, New York, 1993.

[BCP93] H. Brönnimann, B. Chazelle, and J. Pach. How hard is halfspace range searching? *Discrete Comput. Geom.*, 10:143–155, 1993.

[Ben80] J.L. Bentley. Multidimensional divide-and-conquer. *Commun. ACM*, 23:214–229, 1980.

[BF79] J.L. Bentley and J.H. Friedman. Data structures for range searching. *ACM Comput. Surv.*, 11:397–409, 1979.

[BS80] J.L. Bentley and J.B. Saxe. Decomposable searching problems I: Static-to-dynamic transformation. *J. Algorithms*, 1:301–358, 1980.

[CE87] B. Chazelle and H. Edelsbrunner. Linear space data structures for two types of range search. *Discrete Comput. Geom.*, 2:113–126, 1987.

[Cha86] B. Chazelle. Filtering search: A new approach to query-answering. *SIAM J. Comput.*, 15:703–724, 1986.

[Cha88a] B. Chazelle. An algorithm for segment-dragging and its implementation. *Algorithmica*, 3:205–221, 1988.

[Cha88b] B. Chazelle. A functional approach to data structures and its use in multidimensional searching. *SIAM J. Comput.*, 17:427–462, 1988.

[Cha89] B. Chazelle. Lower bounds on the complexity of polytope range searching. *J. Amer. Math. Soc.*, 2:637–666, 1989.

[Cha90a] B. Chazelle. Lower bounds for orthogonal range searching, I: The reporting case. *J. Assoc. Comput. Mach.*, 37:200–212, 1990.

[Cha90b] B. Chazelle. Lower bounds for orthogonal range searching, II: The arithmetic model. *J. Assoc. Comput. Mach.*, 37:439–463, 1990.

[Cha93] B. Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete Comput. Geom.*, 9:145–158, 1993.

[Cha94a] B. Chazelle. Computational geometry: A retrospective. In *Proc. 26th Annu. ACM Sympos. Theory Comput.*, pages 75–94, 1994.

[Cha94b] B. Chazelle. A spectral approach to lower bounds. In *Proc. 35th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 674–682, 1994.

[Cha95] B. Chazelle. Lower bounds for off-line range searching. In *Proc. 27th Annu. ACM Sympos. Theory Comput.*, pages 733–740, 1995.

[Cha96] T. Chan. Fixed-dimensional linear programming queries made easy. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, page 284–290, 1996.

[CR89] B. Chazelle and B. Rosenberg. Computing partial sums in multidimensional arrays. In *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, pages 131–139, 1989.

[CSW92] B. Chazelle, M. Sharir, and E. Welzl. Quasi-optimal upper bounds for simplex range searching and new zone theorems. *Algorithmica*, 8:407–429, 1992.

[CT92] Y.-J. Chiang and R. Tamassia. Dynamic algorithms in computational geometry. *Proc. IEEE*, 80:1412–1434, 1992.

[dB93] M. de Berg. *Ray Shooting, Depth Orders and Hidden Surface Removal*, volume 703 of *Lecture Notes in Comput. Sci.* Springer-Verlag, Berlin, 1993.

[Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monogr. Theoret. Comput. Sci.*. Springer-Verlag, Heidelberg, 1987.

[Fre80] M.L. Fredman. The inherent complexity of dynamic data structures which accommodate range queries. In *Proc. 21st Annu. IEEE Sympos. Found. Comput. Sci.*, pages 191–199, 1980.

[Fre81] M.L. Fredman. A lower bound on the complexity of orthogonal range queries. *J. Assoc. Comput. Mach.*, 28:696–705, 1981.

[FvD+90] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, Reading, 1990.

[GJS94] P. Gupta, R. Janardan, and M. Smid. Efficient algorithms for generalized intersection searching on non-iso-oriented objects. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 369–378, 1994.

[GJS95] P. Gupta, R. Janardan, and M. Smid. Further results on generalized intersection searching problems: Counting, reporting and dynamization. *J. Algorithms*, 19:282–317, 1995.

[Gla89] A.S. Glassner. *Ray Tracing.* Academic Press, San Diego, 1989.

[GPW93] J.E. Goodman, R. Pollack, and R. Wenger. Geometric transversal theory. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms Combin.*, pages 163–198. Springer-Verlag, Heidelberg, 1993.

[GT93] M.T. Goodrich and R. Tamassia. Dynamic ray shooting and shortest paths via balanced geodesic triangulations. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 318–327, 1993.

[HS95] J. Hershberger and S. Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. *J. Algorithms*, 18:403–431, 1995.

[HW87] D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. *Discrete Comput. Geom.*, 2:127–151, 1987.

[IA87] H. Imai and Ta. Asano. Dynamic orthogonal segment intersection search. *J. Algorithms*, 8:1–18, 1987.

[KPW92] J. Komlós, J. Pach, and G. Woeginger. Almost tight bounds for ϵ-nets. *Discrete Comput. Geom.*, 7:163–173, 1992.

[Mat92] J. Matoušek. Efficient partition trees. *Discrete Comput. Geom.*, 8:315–334, 1992.

[Mat93a] J. Matoušek. Linear optimization queries. *J. Algorithms*, 14:432–448, 1993.

[Mat93b] J. Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.*, 10:157–182, 1993.

[Mat94] J. Matoušek. Geometric range searching. *ACM Comput. Surv.*, 26:421–461, 1994.

[Meh84] K. Mehlhorn. *Multi-dimensional Searching and Computational Geometry*, volume 3 of *Data Structures and Algorithms*. Springer-Verlag, Heidelberg, 1984.

[MN90] K. Mehlhorn and S. Näher. Dynamic fractional cascading. *Algorithmica*, 5:215–241, 1990.

[MS93] J. Matoušek and O. Schwarzkopf. On ray shooting in convex polytopes. *Discrete Comput. Geom.*, 10:215–232, 1993.

[Mul91] K. Mulmuley. Randomized multidimensional search trees: Dynamic sampling. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 121–131, 1991.

[Mul94] K. Mulmuley. *Computational Geometry: An Introduction through Randomized Algorithms*. Prentice Hall, Englewood Cliffs, 1994.

[Ove83] M.H. Overmars. *The Design of Dynamic Data Structures*, volume 156 of *Lecture Notes in Comput. Sci.* Springer-Verlag, Berlin, 1983.

[Ove88] M.H. Overmars. Efficient data structures for range searching on a grid. *J. Algorithms*, 9:254–275, 1988.

[PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, 1985.

[Sch92] O. Schwarzkopf. *Dynamic Maintenance of Convex Polytopes and Related Structures*. Ph.D. thesis, Fachbereich Mathematik, Freie Universität Berlin, Berlin, 1992.

[SR95] S. Subramanian and S. Ramaswamy. The P-range tree: A new data structure for range searching in secondary memory. In *Proc. 6th ACM-SIAM Sympos. Discrete Algorithms*, pages 378–387, 1995.

[VV96] J.S. Vitter and D.E. Vengroff. Efficient 3-d range searching in external memory. In *Proc. 28th Annu. ACM Sympos. Theory Comput.*, pages 192–201, 1996.

[Wil82] D.E. Willard. Polygon retrieval. *SIAM J. Comput.*, 11:149–165, 1982.

[Wil92] D.E. Willard. Applications of the fusion tree method to computational geometry and searching. In *Proc. 3rd ACM-SIAM Sympos. Discrete Algorithms*, pages 286–296, 1992.

32 RAY SHOOTING AND LINES IN SPACE

Marco Pellegrini

INTRODUCTION

The geometry of lines in 3-space has been a part of the body of classical algebraic geometry since the pioneering work of Plücker. Interest in this branch of geometry has been revived in recent years by several converging trends in computer science. The discipline of computer graphics has pursued the task of rendering realistic images by simulating the flow of light within a scene according to the laws of elementary optical physics. In these models light moves along straight lines in 3-space and a computational challenge is to find efficiently the intersections of a very large number of rays with the objects comprising the scene. In robotics the chief problem is that of moving 3-dimensional objects without collisions. Effects due to the edges of objects have been studied as a special case of the more general problem of representing and manipulating lines in 3-space. Computational geometry (whose core is better termed "design and analysis of geometric algorithms") has moved recently from the realm of planar problems to tackling directly problems that are specifically 3-dimensional. The new and sometimes unexpected computational phenomena generated by lines (and segments) in 3-space have emerged as a main focus of research.

In this chapter we will survey the present state of the art on lines and ray shooting in 3-space from the point of view of computational geometry. The emphasis is on provable nontrivial bounds for the time and storage used by algorithms for solving natural problems on lines, rays, and polyhedra in 3-space. We start by mentioning different possible choices of coordinates for lines (Section 32.1). This is an essential initial step because different coordinates highlight different properties of the lines in their interaction with other geometric objects. Here a special role is played by *Plücker coordinates*

(Section 32.1), which represent the starting point for many of the most recent results. Then we consider how lines interact with each other (Section 32.2). We are given a finite set of lines L that act as obstacles and we will define other (infinite) sets of lines induced by L that capture some of the important properties of visibility and motion problems. We show bounds on the storage required for a complete description of such sets. Then we move a step forward by considering the same sets of lines when the obstacles are polyhedral sets, more commonly encountered in applications. We arrive in Section 32.3 at the ray-shooting problem and its variants (on-line, off-line, arbitrary direction, fixed direction, and shooting with objects other than rays). Again, the obstacles are usually polyhedral objects, but in one case we are able to report a ray-shooting result on curved objects (spheres).

Section 32.4 is devoted to the problem of collision-free movements (arbitrary or translation only) of lines among obstacles. This problem arises, for example, when lines are used to model radiation or light beams (e.g., lasers). In Section 32.5 we define a few notions of distance among lines, and as a consequence we have several

natural proximity problems for lines in 3-space. Finding the closest pair in a set of lines is the most basic of such problems.

In Section 32.6 we survey what is known about the "dominance" relation among lines. This relation is central for many visibility problems in graphics. It is used, for example, in the painter's algorithm for hidden surface removal (Chapter 42). Another direction of research has explored the relation between lines in 3-space and their orthogonal projections. One main topic is that of realizability, that is, given a set of planar lines together with a relation, does there exist a corresponding set of lines in 3-space whose dominance is consistent with the given relation?

32.1 COORDINATES OF LINES

GLOSSARY

Homogeneous coordinates: A point (x, y, z) in Cartesian coordinates has homogeneous coordinates (x_0, x_1, x_2, x_3), where $x = x_1/x_0$, $y = x_2/x_0$, and $z = x_3/x_0$.

Oriented lines: A line may have two distinct orientations. A line and an orientation form an oriented line.

Unoriented line: A line for which an orientation is not distinguished.

(I) Canonical coordinates by pairs of planes. The intersection of two planes with equations $y = az + b$ and $x = cz + d$ is a nonhorizontal line in 3-space, uniquely defined by the four parameters (a, b, c, d). Thus these parameters can be taken as coordinates of such lines. In fact, the space of nonhorizontal lines is homeomorphic to $\mathbb{R}^4$. Results on ray shooting among boxes and some lower bounds on stabbing are obtained using these coordinates.

(II) Canonical coordinates by pairs of points. Given two parallel horizontal planes, $z = 1$ and $z = 0$, the intersection points of a nonhorizontal line l with the two planes uniquely define that line. If $(x_0, y_0, 0)$ and $(x_1, y_1, 1)$ are two such points for l, then the quadruple (x_0, y_0, x_1, y_1) can be used as coordinates of l. Results on sets of horizontal polygons are obtained using these coordinates.

Although four is the minimum number of coordinates needed to represent an *unoriented* line, such parametrizations have proved useful only in special cases. Many interesting results have been derived using instead a five-dimensional parametrization for *oriented* lines, called ***Plücker coordinates.***

(III) Plücker coordinates of lines. An oriented line in 3-space can be given by the homogeneous coordinates of two of its points. Let l be a line in 3-space and let $a = (a_0, a_1, a_2, a_3)$ and $b = (b_0, b_1, b_2, b_3)$ be two distinct points in homogeneous coordinates on l. We can represent the line l, oriented from a to b, by the matrix

$$l = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ b_0 & b_1 & b_2 & b_3 \end{pmatrix}, \quad \text{with } a_0, b_0 > 0.$$

By taking the determinants of the six 2×2 submatrices of the above 2×4 matrix

we obtain the ***homogeneous Plücker coordinates*** of the line:

$$p(l) = (\xi_{01}, \xi_{02}, \xi_{03}, \xi_{12}, \xi_{31}, \xi_{23}), \quad \text{with} \quad \xi_{ij} = det \begin{pmatrix} a_i & a_j \\ b_i & b_j \end{pmatrix}.$$

The six numbers ξ_{ij} are interpreted as homogeneous coordinates of a point in 5-space. For a given line l the six numbers are unique modulo a positive multiplicative factor, and they do not depend on the particular distinct points a and b that we have chosen on l. We call $p(l)$ the ***Plücker point*** of l in projective 5-dimensional space $\mathbb{P}^5$.

We also define the ***Plücker hyperplane*** of the line l to be the hyperplane in $\mathbb{P}^5$ with vector of coefficients $v(l) = (\xi_{23}, \xi_{31}, \xi_{12}, \xi_{03}, \xi_{02}, \xi_{01})$. So the Plücker hyperplane is:

$$h(l) = \{p \in \mathbb{P}^5 \mid v(l) \cdot p = 0\}.$$

For each Plücker hyperplane we have a positive and a negative halfspace given by $h^+(l) = \{p \in \mathbb{P}^5 \mid v(l) \cdot p \geq 0\}$ and $h^-(l) = \{p \in \mathbb{P}^5 \mid v(l) \cdot p \leq 0\}$. Not every tuple of 6 real numbers corresponds to a line in 3-space since the Plücker coordinates must satisfy the condition

$$\xi_{01}\xi_{23} + \xi_{02}\xi_{31} + \xi_{03}\xi_{12} = 0. \tag{32.1.1}$$

The set of points in $\mathbb{P}^5$ satisfying Equation 32.1.1 forms the so-called ***Plücker hypersurface*** II; it is also called the ***Klein quadric*** or the ***Grassmannian*** (manifold). The converse is also true: every tuple of six real numbers satisfying Equation 32.1.1 is the Plücker point of some line in 3-space. Given two lines l and l', they intersect or are parallel (i.e., they intersect at infinity) when the four defining points are coplanar. In this case the determinant of the 4×4 matrix formed by the 16 homogeneous coordinates of the four points is zero. In terms of Plücker coordinates we have the following basic lemmas.

LEMMA 32.1.1

Lines l and l' intersect or are parallel (meet at infinity) if and only if $p(l) \in h(l')$.

Note that Equation 32.1.1 states in terms of Plücker coordinates the fact that any line always meets itself.

LEMMA 32.1.2

Let l be an oriented line and t a triangle in Cartesian 3-space with vertices (p_0, p_1, p_2). Let l_i be the oriented line through (p_i, p_{i+1}) (indices mod 3). Then l intersects t if and only if either $p(l) \in h^+(l_0) \cap h^+(l_1) \cap h^+(l_2)$ or $p(l) \in h^-(l_0) \cap h^-(l_1) \cap h^-(l_2)$.

These two lemmas allow us to map combinatorial and algorithmic problems involving lines (and polyhedral sets) in 3-space into problems involving sets of hyperplanes and points in projective 5-space (Plücker space). The main advantage is that we can use the rich collection of results on the combinatorics of high dimensional arrangements of hyperplanes (see Chapter 21). The main drawback is that we are using five (nonhomogeneous) parameters, instead of four which is the minimum number necessary. This choice has a potential for increasing the time bounds of line algorithms. We are rescued by the following theorem:

THEOREM 32.1.3 [APS93]

Given a set H of n hyperplanes in 5-dimensional space, the complexity of the cells of the arrangement $\mathcal{A}(H)$ intersected by the Plücker hypersurface Π (also called the ***zone*** *of Π in $\mathcal{A}(H)$) is $O(n^4 \log n)$.*

Although the entire arrangement $\mathcal{A}(H)$ can be of complexity $\Theta(n^5)$, if we are working only with Plücker points we can limit our constructions to the zone of Π, the complexity of which is one order of magnitude smaller. Theorem 32.1.3 is especially useful for deriving ray-shooting results.

The list of coordinatizations discussed in this section is by no means exhaustive. Other parametrizations are used, for example, in [Ame92], [AAS94], and [AS96].

A TYPICAL EXAMPLE

A typical example of the use of Plücker coordinates in three-dimensional problems is the result for fast ray shooting among polyhedra (see Table 32.3.1). We triangulate the faces of the polyhedra and extend each edge to a full line. Each such line is mapped to a Plücker hyperplane. Lemma 32.1.2 guarantees that each cell in the resulting arrangement of Plücker hyperplanes contains Plücker points that pass through the same set of triangles. Thus to answer a ray-shooting query, we first locate the query Plücker point in the arrangement, and then search the list of triangles associated with the retrieved cell. This final step is accomplished using a binary search strategy when the polyhedra are disjoint. Theorem 32.1.3 guarantees that we need to build a point location structure only for the zone of the Plücker hypersurface, thus saving an order of magnitude over general point location methods for arrangements (see Sections 21.3 and 30.7).

32.2 SETS OF LINES IN 3-SPACE

With Plücker coordinates (III) to represent oriented lines, we can use the topology induced by the standard topology of 5-dimensional projective space $\mathbb{P}^5$ on Π as a natural topology on sets of oriented lines. Using the four-dimensional coordinatizations (I) or (II), we can impose the standard topology of $\mathbb{R}^4$ on the set of nonhorizontal unoriented lines. Thus we can define the concepts of "neighbourhood," "continuous path," "open set," "closed set," "boundary," "path-connected component," and so on, for the set $\mathcal{L}$ of lines in 3-space. The distinction between oriented lines and unoriented lines is mainly technical and the complexity bounds hold in either case.

GROUPS OF LINES INDUCED BY A FINITE SET OF LINES

GLOSSARY

Semialgebraic set: The set of all points that satisfy a Boolean combination of a finite number of algebraic constraints (equalities and inequalities) in the Cartesian coordinates of $\mathbb{R}^d$. See Chapter 29.

Path-connected component: A maximal set of lines that can be connected by a path of lines, a continuous function from the interval $[0, 1]$ to the space of lines.

Positively-oriented lines: Oriented lines l'_1 and l'_2 on the xy-plane are positively-oriented if the triple scalar product of vectors parallel to l'_1, l'_2, and the positive z-axis is positive.

Consistently-oriented lines: An oriented line l in 3-space is oriented consistently with a 3-dimensional set L of oriented lines if the projection l' of l is positively-oriented with the projection of every line in L.

A finite set L of n lines in 3-space can be viewed as an obstacle to the free movement of other lines in 3-space. Many applications lead to defining groups of lines with some special properties with respect to the fixed lines L. The resources used by algorithms for these applications are often bounded by the "complexity" of such groups.

The boundary of a semialgebraic set in $\mathbb{R}^4$ is partitioned into a finite number of faces of dimension 0, 1, 2, and 3, each of which is also a semialgebraic set. The number of faces on the boundary of a semialgebraic set is the ***complexity*** of that set. The groups of lines that we consider are represented in $\mathbb{R}^4$ by semialgebraic sets, with the coefficients of the corresponding algebraic constraints a function of the given finite set of lines L.

The set Miss(L) consists of lines that do not meet any line in L. The sets Vert(L) and Free(L) consists of lines that may be translated to infinity without collision with lines in L. The basic complexities are displayed in Table 32.2.1.

TABLE 32.2.1 Complexity of groups of lines defined by lines.

SET OF LINES	DEFINITION	COMPLEXITY
Miss(L)	do not meet any line in L	$\Theta(n^4)$
1 component of Miss(L)	1 path-connected component	$\Theta(n^2)$
Vert(L)	can be translated vertically to ∞	$\Theta(n^3)$
Free(L)	can be translated to ∞ in some direction	$\Omega(n^3)$,$O(n^3 \log n)$
VertCO(L)	above L and oriented consistently with L	$\Theta(n^2)$

MEMBERSHIP TESTS

When we are given L we can build a data structure during a preprocessing phase so that when we are given a new (query) line l we can decide very efficiently whether l is in one of the sets defined in the previous section. Such an algorithm implements a membership test for a group of lines. Table 32.2.2 shows the main results.

GROUPS OF LINES INDUCED BY POLYHEDRA

GLOSSARY

ϵ: A positive real number, which we may choose arbitrarily close to zero for each algorithm or data structure. A caveat is that the multiplicative constant implicit

TABLE 32.2.2 Membership tests for groups of lines defined by lines.

SET OF LINES	QUERY TIME	PREPROC/STORAGE
Miss(L)	$O(\log n)$	$O(n^{4+\epsilon})$
1 component of Miss(L), Vert(L), VertCO(L)	$O(\log n)$	$O(n^{2+\epsilon})$
Free(L)	$O(\log n)$	$O(n^{3+\epsilon})$

in the big-O notation depends on ϵ and its value increases when ϵ tends to zero.

$\alpha(\cdot)$: The inverse of Ackerman's function. $\alpha(n)$ grows very slowly and is at most 4 for any practical value of n. See Section 40.4.

$\beta(\cdot)$: $\beta(n) = 2^{c\sqrt{\log n}}$ for a constant c. $\beta(\cdot)$ is a function that is smaller than any polynomial but larger than any polylogarithmic factor. Formally we have that for every $a, b > 0$, $\log^a n \leq \beta(n) \leq n^b$ for any $n \geq n_0(a, b)$.

Polyhedral set P: A region of 3-space bounded by a collection of interior-disjoint vertices, segments, and planar polygons. We denote with n the total number of vertices, edges, and faces.

Star-shaped polyhedron: A polyhedron P for which there exists a point $o \in P$ such that for every point $p \in P$, the open segment op is contained in P.

Terrain: When the star-shaped polyhedron is unbounded and o is at infinity we obtain a terrain, a monotone surface (cf. Section 23.1).

A collection of polyhedra in 3-space may act as obstacles limiting the collision-free movements of lines. Thus, following the blueprint of the previous section, we consider the complexity of some interesting groups of lines induced by polyhedra, displayed in Table 32.2.3.

TABLE 32.2.3 Complexity of groups of lines defined by polyhedra.

SET OF LINES	DEFINITION	COMPLEXITY
Miss(P)	do not meet polyhedron P	$\Theta(n^4)$
Vert(P)	can be translated vertically to ∞	$\Omega(n^3)$, $O(n^3\beta(n))$
Free(P)	can be translated to ∞ in some direction	$\Omega(n^3)$
Miss(Q), Free(Q)	Q star-shaped polyhedron or a terrain	$\Omega(n^2\alpha(n))$, $O(n^3 \log n)$

OPEN PROBLEMS

1. Find an almost cubic upper bound on the complexity of the group of lines Free(P) for a polyhedron P.

2. Close the gap between the quadratic lower and the cubic upper bound for the group Free(T) induced by a terrain T (Table 32.2.3).

SETS OF STABBING LINES

GLOSSARY

Stabber: A line l that intersects every member of a collection $\mathcal{P} = \{P_1, ..., P_k\}$ of polyhedral sets. The sum of the sizes of the polyhedral sets in $\mathcal{P}$ is n. The set of lines stabbing $\mathcal{P}$ is denoted $S(\mathcal{P})$.

Box: A parllelopiped each of whose faces is orthogonal to one of the three Cartesian axes.

c-oriented: Polyhedra whose face normals come from a set of c fixed directions.

Table 32.2.4 lists the worst-case complexity of the set $S(\mathcal{P})$ and the time to find a witness stabbing line.

TABLE 32.2.4 Complexity of the set of stabbing lines and detection time.

OBJECTS	COMPLEXITY OF $S(\mathcal{P})$	FIND TIME
Convex polyhedra	$\Omega(n^3)$, $O(n^3 \log n)$	$O(n^3\beta(n))$
Boxes	$O(n^2)$	$O(n)$
c-oriented polyhedra	$O(n^2)$	$O(n^2)$
Horiz polygons	$\Theta(n^2)$	$O(n)$

Note that in some cases (boxes, parallel polygons) a stabbing line can be found in linear time, even though the best known bound on the complexity of the stabbing set is quadratic. These results are obtained using linear programming techniques (Chapter 38).

We can determine whether a given line l is a stabber for a preprocessed set $\mathcal{P}$ of convex polyhedra in time $O(\log n)$, using data structures of size $O(n^{2+\epsilon})$ that can be constructed in time $O(n^{2+\epsilon})$.

OPEN PROBLEMS

1. Can linear programming techniques yield a linear-time algorithm for c-oriented polyhedra?
2. The lower bound for $S(\mathcal{P})$ for a set of pairwise *disjoint* convex polyhedra is only $\Omega(n^2)$ [PS92]. Close the gap between this and the cubic upper bound.

32.3 RAY SHOOTING

Ray shooting is an important operation in computer graphics and a primitive operation useful in several geometric computations (e.g., hidden surface removal, and

detecting and computing intersections of polyhedra). The problem is defined as follows. Given a large collection $\mathcal{P}$ of simple polyhedral objects, we want to know, for a given point p and direction $\vec{d}$, the first object in $\mathcal{P}$ intersected by the ray defined by the pair $p, \vec{d}$. A single polyhedron with many faces can be represented without loss of generality by the collection of its faces, each treated as a separate polygon.

ON-LINE RAY SHOOTING IN AN ARBITRARY DIRECTION

Here we consider the on-line model in which the set $\mathcal{P}$ is given in advance and a data structure is produced and stored. Afterwards we are given the query rays one-by-one and the answer to one query must be produced before the next query is asked.

Table 32.3.1 summarizes the known complexity bounds on this problem. For a given class of objects we report the query time, the storage, and the preprocessing time of the method with the best bound. In this table and in the following ones we have omitted the big-O symbols. Again, n denotes the sum of the sizes of all the polyhedra in $\mathcal{P}$.

GLOSSARY

Fat horizontal polygons: Convex polygons contained in planes parallel to the xy-plane, with a lower bound on the size of their minimum interior angle.

Curtains: Polygons in 3-space bounded by one segment and by two vertical rays from the endpoints of the segment.

Axis-oriented curtains: Curtains hanging from a segment parallel to the x- or y-axis.

TABLE 32.3.1 On-line ray shooting in an arbitrary direction.

OBJECTS	QUERY	STORAGE	PREPROCESSING
Boxes, terrains, curtains	$\log n$	$n^{2+\epsilon}$	$n^{2+\epsilon}$
Boxes	$n^{1+\epsilon}/m^{1/2}$	$n \le m \le n^2$	$m^{1+\epsilon}$
Polyhedra	$\log n$	$n^{4+\epsilon}$	$n^{4+\epsilon}$
Polyhedra	$n^{1+\epsilon}/m^{1/4}$	$n \le m \le n^4$	$m^{1+\epsilon}$
Fat horiz polygons	$\log n$	$n^{2+\epsilon}$	$n^{2+\epsilon}$
Horiz polygons	$\log^3 n$	$n^{3+\epsilon} + K$	$n^{3+\epsilon} + K \log n$
Spheres	$\log^4 n$	$n^{3+\epsilon}$	$n^{3+\epsilon}$
1 convex polyhedron	$\log n$	n	$n \log n$
s convex polyhedra	$\log^2 n$	$n^{2+\epsilon}s^2$	$n^{2+\epsilon}s^2$

When we drop the fatness assumption for horizontal polygons we obtain bounds that depend on K, the complexity of the set of lines missing the *edges* of the polygons. K is in the range $[n^2, ..., n^4]$ and reaches the upper end of the range only when the polygons are very long and thin.

Most of the data structures for ray shooting mentioned in Table 32.3.1 can be made dynamic (under insertion and deletion of objects in the scene) by using general dynamization techniques (see [Meh84]) and other more recent results [AEM92].

ON-LINE RAY SHOOTING IN A FIXED DIRECTION

We can usually improve on the general case if the direction of the rays is fixed a priori, while the source of the ray can lie anywhere in $\mathbb{R}^3$. See Table 32.3.2; here k is the number of vertices, edges, faces, and cells of the arrangement of the (possibly intersecting) polyhedra.

TABLE 32.3.2 On-line ray shooting in a fixed direction.

OBJECTS	QUERY TIME	STORAGE	PREPROCESSING
Boxes	$\log n$	$n^{1+\epsilon}$	$n^{1+\epsilon}$
Boxes	$\log n(\log\log n)^2$	$n\log n$	$n\log^2 n$
Axis-oriented curtains	$\log n$	$n\log n$	$n\log n$
Polyhedra	$\log^2 n$	$n^{2+\epsilon}+k$	$n^{2+\epsilon}+k\log n$
Polyhedra	$n^{1+\epsilon}/m^{1/3}$	$n\le m\le n^3$	$m^{1+\epsilon}$

OFF-LINE RAY SHOOTING IN AN ARBITRARY DIRECTION

In the previous section we considered the on-line situation when the answer to the query must be generated before the next question is asked. In many situations we do not need such strict requirements. For example, we might know all the queries from the start and are interested in minimizing the total time needed to answer all of the queries (the ***off-line*** situation). In this case there are simpler algorithms that improve on the storage bounds of on-line algorithms:

THEOREM 32.3.1

Given a polyhedral set $\mathcal{P}$ with n vertices, edges, and faces, and given m rays off-line, *we can answer the m ray-shooting queries in time $O(m^{0.8}n^{0.8+\epsilon}+m\log^2 n+\log n\log m)$ using $O(n+m)$ storage.*

One of the most interesting applications of this result is the current asymptotically fastest algorithm for detecting whether two nonconvex polyhedra in 3-space intersect, and to compute their intersection. See Table 32.3.3; here k is the size of the intersection.

TABLE 32.3.3 Detection and computation of intersection among polyhedra.

OBJECTS	DETECTION	COMPUTATION
Polyhedra	$n^{1.6+\epsilon}$	$n^{1.6+\epsilon}+k\log^2 n$
Terrains	$n^{4/3+\epsilon}$	$n^{4/3+\epsilon}+k^{1/3}n^{1+\epsilon}+k\log^2 n$

EXTENSIONS AND ALTERNATIVE METHODS

Some ray-shooting results of Agarwal and Matoušek are obtained from the observation that a ray is traced by a family of segments $\rho(t)$, where one endpoint is the ray source and the second endpoint lies on the ray at distance t from the source. Using *parametric search* techniques (Chapter 37), Agarwal and Matoušek compute the first value of t for which $\rho(t)$ intersects $\mathcal{P}$, and thus answer the ray-shooting query.

An interesting extension of the concept of shooting rays against obstacles is obtained by shooting triangles and more generally simplices. We consider a family of simplices $s(t)$, indexed by real parameter $t \in \mathbb{R}^+$, where t is the volume of the simplex $s(t)$, such that the simplices form a chain of inclusions: $t_1 \leq t_2 \Rightarrow s(t_1) \subset s(t_2)$, Intuitively we grow a simplex until it first meets one of the obstacles. Surprisingly, when the obstacles are general polyhedra, shooting simplices is not harder than shooting rays.

THEOREM 32.3.2

Given a set of polyhedra $\mathcal{P}$ with n edges we can preprocess it in time $O(m^{1+\epsilon})$ into a data structure of size m, such that the following queries can be answered in time $O(n^{1+\epsilon}/m^{1/4})$: Given a simplex s, does s avoid $\mathcal{P}$? Given a family of simplices $s(t)$ as above, which is the first value of t for which $s(t)$ intersects $\mathcal{P}$?

Other popular methods for solving ray-shooting problems are based on triangulations of three-dimensional space, binary space partitions, solid modeling schemes, etc. The performances of these methods are usually not fully analyzable using algorithmic analysis.

OPEN PROBLEMS

1. Find time and storage bounds for ray-shooting general polyhedra that are sensitive to the actual complexity of a group of lines (as opposed to the worst case bound on such a complexity).
2. For a collection of s convex polyhedra there is a wide gap in storage and preprocessing between the special case $s = 1$ and the case for general s. It would be interesting to obtain a bound that depends smoothly on s.
3. No lower bound on time or storage required for ray shooting is known.

32.4 MOVING LINES AMONG OBSTACLES

ARBITRARY MOTIONS

So far we have treated lines as static objects. In this section we consider moving lines. A laser beam in manufacturing or a radiation beam in radiation therapy can be modeled as lines in 3-space moving among obstacles. The main computational problem is to decide whether a source line l_1 can be moved continuously until it

coincides with a target line l_2 so that it avoids a set of obstacles $\mathcal{P}$. We consider the following situation where the set of obstacles $\mathcal{P}$ is given in advance and preprocessed to obtain a data structure. When the query lines l_1 and l_2 are given the answer is produced before a new query is accepted. We have the results shown in Table 32.4.1, where K is the complexity of the set of lines missing the edges of the polygons (cf. Section 32.2).

TABLE 32.4.1 On-line collision-free movement of lines among obstacles.

OBJECTS	QUERY TIME	STORAGE	PREPROC
Polyhedra	$\log n$	$n^{4+\epsilon}$	$n^{4+\epsilon}$
Horiz polygons	$\log^3 n$	$n^{3+\epsilon} + K$	$n^{3+\epsilon} + K \log n$

OPEN PROBLEMS

It is not known how to trade off storage and query time, or whether better bounds can be obtained in an off-line situation.

TRANSLATIONS

We now restrict the type of motion and consider only translations of lines. The first result is negative: there are sets of lines which cannot be split by any collision-free translation. There exists a set L of 9 lines such that, for all directions v and all subsets $L_1 \subset L$, L_1 cannot be translated continuously in direction v without collisions with $L \setminus L_1$. Positive results are displayed in Table 32.4.2.

GLOSSARY

Towering property: Two sets of lines L_1 and L_2 are said to satisfy the towering property if we can translate simultaneously all lines in L_1 in the vertical direction without any collision with any lines in L_2.

Separation property: Two sets of lines satisfy the separation property if they satisfy the towering property in some direction (not necessarily vertical).

TABLE 32.4.2 Separating lines by translations.

PROPERTY	TIME TO CHECK PROPERTY
Towering	$O(n^{4/3+\epsilon})$
Separation	$O(n^{3/2+\epsilon})$

32.5 CLOSEST PAIR OF LINES

GLOSSARY

Distance between lines: The Euclidean distance between two lines l_1 and l_2 in 3-space is the length of the shortest segment with one endpoint on l_1 and the other on l_2.

Vertical distance between lines: The length of the vertical segment with one endpoint on l_1 and one endpoint of l_2 (provided a unique such segment exists).

Vertical distance between segments: The length of the vertical segment with one endpoint in s_1 and one in s_2. If a unique such vertical segment does not exist the vertical distance is undefined.

TABLE 32.5.1 Closest and farthest pair of lines and segments.

PROBLEM	OBJECTS	TIME
Smallest distance	lines	$O(n^{8/5+\epsilon})$
Smallest vertical distance	lines, segments	$O(n^{8/5+\epsilon})$
Largest vertical distance	lines, segments	$O(n^{4/3+\epsilon})$

OPEN PROBLEM

Finding an algorithm with subquadratic time complexity for the smallest distance among segments (and more generally, among polyhedra) is a notable open question.

32.6 DOMINANCE RELATION AND WEAVINGS

GLOSSARY

Dominance relation: Given a finite set L of nonvertical disjoint lines in $\mathbb{R}^3$, define a dominance relation $\prec$ among lines in L as follows: $l_1 \prec l_2$ if l_2 lies above l_1, i.e., if, on the vertical line intersecting l_1 and l_2, the intersection with l_1 has smaller z-coordinate than the intersection with l_2.

Weaving: A weaving is a pair $(L', \prec')$ where L' is a set of lines on the plane and $\prec'$ is an anti-symmetric nonreflexive binary relation $\prec' \subset L' \times L'$ among the lines in L'.

Realizable: A weaving is realizable if there exists a set of lines L in 3-space such that L' is the projection of L and $\prec'$ is the image of the dominance relation $\prec$ for L.

Elementary cycle: A cycle in the dominance relation such that the projections of the lines in such a cycle bound a cell of the arrangement of projected lines.

Perfect: A weaving $(L', \prec')$ is perfect if each line l alternates below and above the other lines in the order they cross l (see Figure 32.6.1a).

Bipartite weaving: Two families of segments in 3-space such that, when projecting on the xy-plane, each segment does not meet segments from its own family and meets all the segments from the other family. (A bipartite weaving of size 4×4 is shown in Figure 32.6.1b.)

Perfect bipartite weaving: Every segment alternates above and below the segments of the other family (see Figure 32.6.1b).

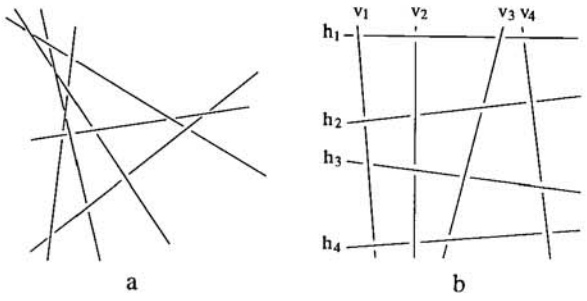


FIGURE 32.6.1
(a) *A perfect weaving;*
(b) *a perfect bipartite weaving.*

The dominance relation is possibly cyclic, that is, there may be three lines such that $l_1 \prec l_2 \prec l_3 \prec l_1$. Some results related to dominance are the following:

1. *How fast can we generate a consistent linear extension if the relation $\prec$ is acyclic?* $O(n^{4/3+\epsilon})$ time is sufficient for the case of lines. This result has been extended to the case of segments and polyhedra. If an ordering is given as input, it is possible to *verify* that it is a linear extension of $\prec$ in time $O(n^{4/3+\epsilon})$.

2. *How many elementary cycles in the dominance relation can n lines define?* In the case of bipartite weavings, the dominance relation can have $O(n^{3/2})$ elementary cycles and there is a family of bipartite weavings attaining the lower bound $\Omega(n^{4/3})$.

3. *If we cut the segments to eliminate cycles, how many "cuts" are necessary to eliminate all cycles?* From the previous result we have that sometimes $\Omega(n^{4/3})$ cuts are necessary since a single cut can eliminate only one elementary cycle. In order to eliminates all cycles (including the nonelementary ones) in a bipartite weaving, $O(n^{9/5})$ cuts are always sufficient.

4. The fraction of realizable weavings over all possible weavings of n lines tends to 0 exponentially as n tends to ∞.

5. A perfect weaving of $n \geq 4$ lines is not realizable.

6. Perfect bipartite weavings are realizable only if one of the families has fewer than four segments.

32.7 SOURCES AND RELATED MATERIAL

FURTHER READING

Books and Surveys.

[Som51, HP52, Jes03]: Extensive book-length treatments of the geometry of lines in space.

[Sto89, Sto91]: Algorithmic aspects of computing in projective spaces.

[BR79, Shi78]: Uses of the geometry of lines in robotics. For uses in graphics see [FVFH90].

[dB93]: A detailed description of many ray-shooting results.

[Spe92]: A guide to the vast related literature on pragmatic aspects of ray shooting.

Key papers.

[Plu65]: The original description of Plücker coordinates. See [APS93] for Theorem 32.1.3.

Groups of lines induced by lines (Table 32.2.1) are discussed in [CEGS89, Pel94b]. Membership tests (Table 32.2.2) are in [CEGS89, Pel93b, Pel94b]. The bounds on sets of lines induced by polyhedra in Table 32.2.3 are in [HS94, Pel94b, Aga94]. A key paper in stabbing is [PS92]. Other results on stabbing are in [Pel93a, Aga94, Ame92, Meg91, Pel91].

The main references on ray shooting (Table 32.3.1) are in [Pel93b, dBH$^+$94] (boxes), [AM93, AM94, Pel93b, dBH$^+$94, AS93b] (polyhedra), [Pel94c] (horizontal polygons), [AAS94, MS97] (spheres), and [DK85, AS93a] (convex polyhedra). References for ray shooting in a fixed direction (Table 32.3.2) are [dB93, dBGH94].

Results on off-line ray shooting and computing intersection of polyhedra (Table 32.3.3) are in [CEGS94, Pel94b, Pel93b]. Theorem 32.3.2 is in [Pel94a]. Results on moving and translating lines in 3-space (Tables 32.4.1 and 32.4.2) are in [SS93], [Pel93b, Pel94c], and [CEGS89, Pel94b]. Neighbor problems for lines (Table 32.5.1) are discussed in [CEGS93, Pel94a].

Weavings are discussed in [CEG$^+$92] and [PPW93]. Results on linear extensions of the dominance relation are in [dBOS94].

RELATED CHAPTERS

Chapter 21: Arrangements
Chapter 30: Point location
Chapter 31: Range searching
Chapter 33: Geometric intersection
Chapter 37: Parametric search
Chapter 40: Algorithmic motion planning
Chapter 42: Computer graphics

REFERENCES

[AAS94] P.K. Agarwal, B. Aronov, and M. Sharir. Computing envelopes in four dimensions with applications. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 348–358, 1994.

[AEM92] P.K. Agarwal, D. Eppstein, and J. Matoušek. Dynamic half-space range searching, geometric optimization and minimum spanning trees. In *Proc. 33rd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 80–89, 1992.

[Aga94] P.K. Agarwal. On stabbing lines for convex polyhedra in 3D. *Comp. Geom. Theory Appl.*, 4:177–189, 1994.

[AM93] P.K. Agarwal and J. Matoušek. Ray shooting and parametric search. *SIAM J. Comput.*, 22:794–806, 1993.

[AM94] P.K. Agarwal and J. Matoušek. Range searching with semialgebraic sets. *Discrete Comput. Geom.*, 11:393–418, 1994.

[Ame92] N. Amenta. Finding a line transversal of axial objects in three dimensions. In *Proc. 3rd ACM-SIAM Sympos. Discrete Algorithms*, pages 66–71, 1992.

[APS93] B. Aronov, M. Pellegrini, and M. Sharir. On the zone of an algebraic surface in a hyperplane arrangement. *Discrete Comput. Geom.*, 9:177–188, 1993.

[AS93a] P.K. Agarwal and M. Sharir. Ray shooting amidst convex polytopes in three dimensions. In *Proc. 4th ACM-SIAM Sympos. Discrete Algorithms*, pages 260–270, 1993.

[AS93b] P.K. Agarwal and M. Sharir. Applications of a new space partitioning technique. *Discrete Comput. Geom.*, 9:11–38, 1993.

[AS96] P.K. Agarwal and M. Sharir. Efficient randomized algorithms for some geometric optimization problems. *Discrete Comput. Geom.*, 16:317–337, 1996.

[BR79] O. Bottima and B. Roth. *Theoretical Kinematics.* North-Holland, Amsterdam, 1979.

[CEG+92] B. Chazelle, H. Edelsbrunner, L.J. Guibas, R. Pollack, R. Seidel, M. Sharir, and J. Snoeyink. Counting and cutting cycles of lines and rods in space. *Comput. Geom. Theory Appl.*, 1:305–323, 1992.

[CEGS89] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. Lines in space: Combinatorics, algorithms and applications. In *Proc. 21st Annu. ACM Sympos. Theory Comput.*, pages 382–393, 1989.

[CEGS93] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. Diameter, width, closest line pair and parametric search. *Discrete Comput. Geom.*, 10:183–196, 1993.

[CEGS94] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. Algorithms for bichromatic line segment problems and polyhedral terrains. *Algorithmica*, 11:116–132, 1994.

[dB93] M. de Berg. *Ray Shooting, Depth Orders and Hidden Surface Removal*, volume 703 of *Lecture Notes in Comput. Sci.* Springer-Verlag, New York, 1993.

[dBGH94] M. de Berg, L. Guibas, and D. Halperin. Vertical decompositions for triangles in 3-space. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 1–10, 1994.

[dBH+94] M. de Berg, D. Halperin, M. Overmars, J. Snoeyink, and M. van Kreveld. Efficient ray-shooting and hidden surface removal. *Algorithmica*, 12:31–53, 1994.

[dBOS94] M. de Berg, M. Overmars, and O. Schwarzkopf. Computing and verifying depth orders. *SIAM J. Comput.*, 23:437–446, 1994.

[DK85] D.P. Dobkin and D.G. Kirkpatrick. A linear algorithm for determining the separation of convex polyhedra. *J. Algorithms*, 6:381–392, 1985.

[FVFH90] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, Reading, 1990.

[HP52] W.V.D. Hodge and D. Pedoe. *Methods of Algebraic Geometry.* Cambridge University Press, 1952.

[HS94] D. Halperin and M. Sharir. New bounds for lower envelopes in three dimensions, with applications to visibility in terrains. *Discrete Comput. Geom.*, 12:313–326, 1994.

[Jes03] C.M. Jessop. *A Treatise on the Line Complex.* Cambridge University Press, 1903.

[Meg91] N. Megiddo. Personal communication. 1991.

[Meh84] K. Mehlhorn. *Multidimensional Searching and Computational Geometry.* Springer-Verlag, Berlin, 1984.

[MS97] S. Mohaban and M. Sharir. Ray-shooting amidst spheres in three-dimensions and related problems. *SIAM J. Comput.* 26, 1997, to appear.

[Pel91] M. Pellegrini. *Combinatorial and Algorithmic Analysis of Stabbing and Visibility Problems in 3-Dimensional Space.* PhD thesis, Courant Institute, New York Univ., 1991. Robotics Lab Tech. Rep. 241.

[Pel93a] M. Pellegrini. Lower bounds on stabbing lines in 3-space. *Comput. Geom. Theory Appl.*, 3:53–58, 1993.

[Pel93b] M. Pellegrini. Ray shooting on triangles in 3-space. *Algorithmica*, 9:471–494, 1993.

[Pel94a] M. Pellegrini. On collision-free placements of simplices and the closest pair of lines in 3-space. *SIAM J. Comput.*, 23:133–153, 1994.

[Pel94b] M. Pellegrini. On lines missing polyhedral sets in 3-space. *Discrete Comput. Geom.*, 12:203–221, 1994.

[Pel94c] M. Pellegrini. On point location and motion planning in arrangements of simplices. In *Proc. 26th Annu. ACM Sympos. Theory Comput.*, pages 95–104, 1994.

[Plu65] J. Plücker. On a new geometry of space. *Philos. Trans. Royal Soc. London*, 155:725–791, 1865.

[PPW93] J. Pach, R. Pollack, and E. Welzl. Weaving patterns of lines and line segments in space. *Algorithmica*, 9:561–571, 1993.

[PS92] M. Pellegrini and P. Shor. Finding stabbing lines in 3-space. *Discrete Comput. Geom.*, 8:191–208, 1992.

[Shi78] B.E. Shimano. *The Kinematic Design and Force Control of Computer Controlled Manipulators.* PhD thesis, Dept. of Mechanical Engineering, Stanford Univ., 1978.

[Som51] D.M.H. Sommerville. *Analytical Geometry of Three Dimensions.* Cambridge University Press, 1951.

[Spe92] R. Speer. An updated cross-indexed guide to the ray-tracing literature. *Comput. Graphics*, 26:41–72, 1992.

[SS93] J. Snoeyink and J. Stolfi. Objects that cannot be taken apart with two hands. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 246–256, 1993.

[Sto89] J. Stolfi. Primitives for computational geometry. Tech. Rep. 36, Digital Systems Research Center, Palo Alto, 1989.

[Sto91] J. Stolfi. *Oriented Projective Geometry: a Framework for Geometric Computations.* Academic Press, San Diego, 1991.

33 GEOMETRIC INTERSECTION

David M. Mount

INTRODUCTION

Detecting whether two geometric objects intersect and computing the region of intersection are fundamental problems in computational geometry. Geometric intersection problems arise naturally in a number of applications. Examples include geometric packing and covering, wire and component layout in VLSI, motion planning, and collision detection. In solid modeling, computing the volume of intersection of two shapes is an important step in defining complex solids. In computer graphics, detecting the objects that overlap a viewing window is an example of an intersection problem, as is computing the first intersection of a ray and a collection of geometric solids.

Intersection problems are fundamental to many aspects of geometric computing. It is beyond the scope of this chapter to completely survey this area. Instead we illustrate a number of the main techniques used in efficient intersection algorithms. This chapter is organized as follows. Section 33.1 discusses intersection primitives, the low-level issues of computing intersections that are common to high-level algorithms. Section 33.2 discusses detecting the existence of intersections. Section 33.3 focuses on issues related to counting the number of intersections and reporting intersections. Section 33.4 deals with problems related to constructing the actual region of intersection. Section 33.5 considers methods for geometric intersections based on spatial subdivisions.

33.1 INTERSECTION PRIMITIVES

GLOSSARY

Primitive objects: Simple building blocks for complex objects, each of constant description complexity.

Boundary elements: The vertices, edges, and faces of various dimensions that make up the boundary of an object.

Implicit representation: Points (of a surface) defined as those satisfying some equation.

Parametric representation: Points whose coordinates are defined by functions of one or more real-valued parameters.

Complex geometric objects are typically constructed from a number of primitive objects. Intersection algorithms that operate on complex objects often work by

breaking the problem into a number of primitive intersection operations applied to pairs of primitive objects.

Computing primitive intersection operations in a manner that is efficient, accurate, and robust can be quite difficult, and even experts disagree on the best approaches to this problem. Much of the research in computational geometry has focused on flat polygonal and polyhedral objects. Such objects are typically represented by their boundary elements and *topological information* that describes how these boundary elements are connected. Intersection primitives are applied to pairs of boundary elements, and reduce these objects to simple problems in linear algebra.

In computer graphics and solid modeling, curved surfaces are often used as well. Curved surfaces may be represented implicitly, or more often parametrically. Computing intersection primitives typically involves combining these equations into a system, which can be solved either algebraically or numerically.

Curved surfaces provide their own unique set of challenges in maintaining accuracy and robustness. Dobkin et al. have shown that many of the techniques that have been developed for flat polygonal and polyhedral intersections can be generalized to handle objects with curved boundaries as well [DSV88]. We will focus primarily on intersection problems involving polygonal and polyhedral objects.

33.2 INTERSECTION DETECTION

GLOSSARY

Polygonal chain: A closed chain of segments.

Self-intersecting: Said of a polygonal chain if any pair of nonadjacent edges intersects one another.

Bounding box: A rectangular box surrounding an object, usually axis-aligned (***isothetic***).

Ray-shooting: Determining which obstacle is hit first when a ray is "shot" from a given point in a particular direction. (See Chapter 32.)

Independent set: A subset of nodes of a graph such that no two nodes of the subset are adjacent to one another.

Intersection detection, the easiest of all intersection tasks, requires merely determining the existence of an intersection. The types of objects considered in this section are polygons, polyhedra, and line segments. Let P and Q denote the two objects to be tested for intersection. Throughout, n_p and n_q denote the combinatorial complexity of P and Q, respectively, that is, the number of vertices, edges, and faces (for polyhedra). Let $n = n_p + n_q$ denote the total complexity.

Table 33.2.1 summarizes a number of results on intersection detection. In the table, the terms *convex* and *simple* refer to convex and simple polygons, respectively. The notation $(s(n), q(n))$ in the "Time" column means that the solution involves preprocessing, where a data structure of size $O(s(n))$ is constructed so that intersection detection queries can be answered in $O(q(n))$ time.

TABLE 33.2.1 Intersection detection.

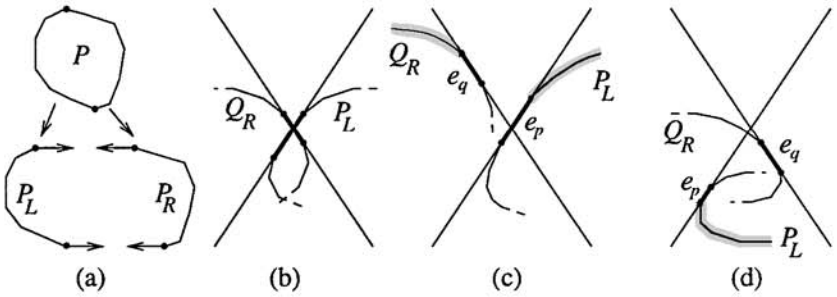
DIM	OBJECTS	TIME	SOURCE
2	convex-convex	$\log n$	[DK83]
2	simple-simple	n	[Cha91]
2	simple-simple	$(n, s \log^2 n)$	[Mou92]
2	line segments	$n \log n$	[SH76]
3	convex-convex	n	[DK85]
3	convex-convex	$(n, \log n_p \log n_q)$	[DK90]

INTERSECTION DETECTION OF CONVEX POLYGONS

Perhaps the most easily understood example of how the structure of geometric objects can be exploited to yield an efficient intersection test is that of detecting the intersection of two convex polygons. There are a number of solutions to this problem that run in $O(\log n)$ time. We present one due to Dobkin and Kirkpatrick [DK83].

Assume that each polygon is given by an array of vertex coordinates. The first step of the algorithm is to find the vertices of each of P and Q with the highest and lowest y-coordinates. This can be done in $O(\log n)$ time by an appropriate modification of binary search and consideration of the direction of the edges incident to each vertex [O'R94, Section 7.3]. After these vertices are located, the boundary of each polygon is split into two semi-infinite convex chains, denoted P_L, P_R and Q_L, Q_R (see Figure 33.2.1(a)). P and Q intersect if and only if P_L and Q_R intersect, and P_R and Q_L intersect.

FIGURE 33.2.1
Intersection detection for two convex polygons.



Consider the case of P_L and Q_R. The algorithm applies a variant of binary search. Consider the median edge e_p of P_L and the median edge e_q of Q_R (shown as heavy lines in the figure). By a simple analysis of the relative positions of these edges and the intersection point of the two lines on which they lie, it is possible to determine in constant time either that the polygons intersect, or that half of at least one of the two boundary chains can be eliminated from further consideration. The cases that arise are illustrated in Figure 33.2.1(b)-(d). The shaded regions indicate the portion of the boundary that can be eliminated from consideration.

SIMPLE POLYGONS

Without convexity, it is generally not possible to detect intersections in sublinear time without preprocessing; but efficient tests do exist.

One of the important intersection questions is whether a polygonal chain defines the edges of a simple polygon. The problem reduces to detecting whether the chain is self-intersecting. This problem can be solved efficiently by supposing that the polygonal chain is a simple polygon, attempting to triangulate the polygon, and seeing whether anything goes wrong in the process. Some triangulation algorithms can be modified to detect self intersections. In particular, the problem can be solved in $O(n)$ time by modifying Chazelle's linear-time triangulation algorithm [Cha91]. See Section 23.2.

Another variation is that of determining the intersection of two simple polygons. Chazelle observed that this can also be reduced to testing self intersections in $O(n)$ time by joining the polygons into a single closed chain by a narrow channel as shown in Figure 33.2.2.

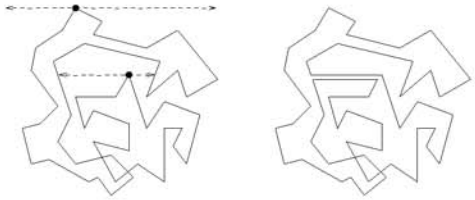


FIGURE 33.2.2
Intersection detection for two simple polygons.

DETECTING INTERSECTIONS OF MULTIPLE OBJECTS

In many applications it is important to know whether any pair of a set of objects intersects one another. Shamos and Hoey showed that the problem of detecting whether a set of n line segments in the plane have an intersecting pair can be solved in $O(n \log n)$ time [SH76]. This is done by plane sweep, which will be discussed later in Section 33.3. They also showed that the same can be done for a set of circles. Reichling showed that this can be generalized to detecting whether any pair of m convex n-gons intersects in $O(m \log m \log n)$ time, and whether they all share a common intersection point in $O(m \log^2 n)$ time [Rei88]. Hopcroft, Schwartz, and Sharir [HSS83] showed how to detect the intersection of any pair of n spheres in 3-space in $O(n \log^2 n)$ time and $O(n \log n)$ space by applying a 3-dimensional plane sweep.

OPEN PROBLEM

What is the complexity of dynamically maintaining a set of objects (e.g., line segments, circles, convex polygons) under the operations of insertion and deletion, so that any pairwise intersections can be detected efficiently?

INTERSECTION DETECTION WITH PREPROCESSING

If preprocessing is allowed, then significant improvements in intersection detection

time may be possible. One of the best-known techniques is to compute an axis-aligned bounding box for each of the objects. Two objects need to be tested for intersection only if their bounding boxes intersect. It is very easy to test whether two such boxes intersect by comparing their projections on each coordinate axis.

Another example is that of ray shooting in a simple polygon. This is a planar version of a well-known 3-dimensional problem in computer graphics. The problem is to preprocess a simple polygon so that given a query ray, the first intersection of the ray with the boundary of the polygon can be determined. After $O(n)$ preprocessing it is possible to answer ray-shooting queries in $O(\log n)$ time. A particularly elegant solution was given by Hershberger and Suri [HS95]. The polygon is triangulated in a special way, called a *geodesic triangulation*, so that any line segment that does not intersect the boundary of the polygon crosses at most $O(\log n)$ triangles. Ray-shooting queries are answered by simply locating the triangle that contains the origin of the ray, and then "walking" the ray through the triangulation. See also Section 23.4.

Mount showed how the geodesic triangulation can be used to generalize the bounding box test for the intersection of simple polygons. Each polygon is preprocessed by computing a geodesic triangulation of its exterior. From this it is possible to determine whether they intersect in $O(s \log^2 n)$ time, where s is the minimum number of edges in a polygonal chain that separates the two polygons [Mou92].

CONVEX POLYHEDRA IN 3-SPACE

The extension of many problems from the plane to 3-space results in a significant increase in difficulty. The problem of detecting the intersection of two convex polyhedra follows this pattern: it is quite a bit harder than is the problem of detecting the intersection of convex polygons. Nonetheless, Dobkin and Kirkpatrick showed that this detection can be performed efficiently by adapting Kirkpatrick's hierarchical decomposition of planar triangulations. Given two polyhedra P and Q having boundary complexity n_p and n_q, respectively, their algorithm runs in $O(\log n_p \log n_q)$ time, assuming that each polyhedron has been individually preprocessed in linear time and space [DK90].

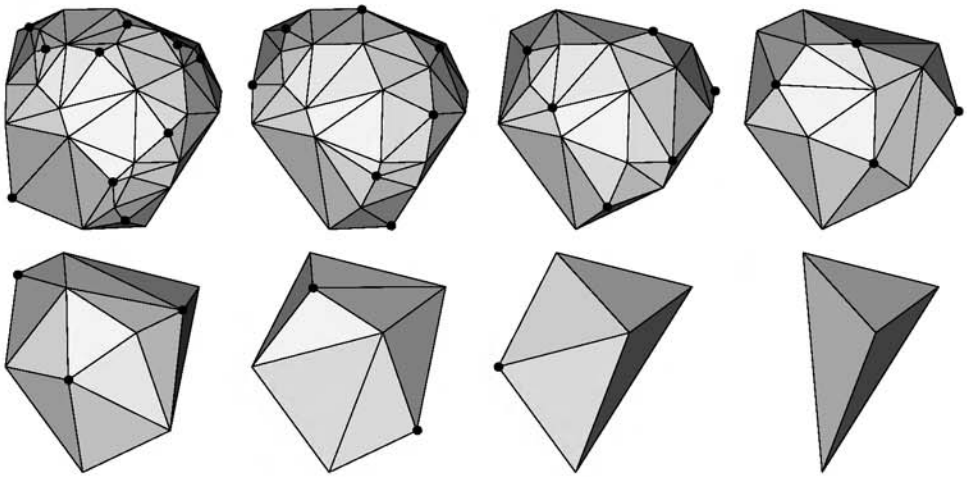
DOBKIN-KIRKPATRICK DECOMPOSITION

Before describing the intersection algorithm, it is important to understand how the hierarchical representation works. Let $P = P_0$ be the initial polyhedron. Assume that P's faces have been triangulated. The vertices, edges, and faces of P's boundary define a planar graph with triangular faces. Let n denote the number of vertices in this graph. An important fact is that every planar graph has an independent set that contains a constant fraction of the vertices formed entirely from vertices of bounded degree. Such an independent set is computed and is removed along with any incident edges and faces from P. Then any resulting "holes" in the boundary of P are filled in with triangles, resulting in a convex polyhedron with fewer vertices (cf. Section 30.6).

These holes can be triangulated independently of one another, each in constant time. The resulting convex polyhedron is denoted P_1. The process is repeated until reaching a polyhedron having at most four vertices. The result is a sequence of polyhedra, $\langle P_0, P_1, \ldots, P_k \rangle$, called the ***Dobkin-Kirkpatrick hierarchy***. Because

a constant fraction of vertices are eliminated at each stage, the depth k of the hierarchy is $O(\log n)$. The hierarchical decomposition is illustrated in Figure 33.2.3. The vertices that are eliminated at each stage, which form an independent set, are highlighted in the figure.

FIGURE 33.2.3
Dobkin-Kirkpatrick decomposition of a convex polyhedron.



INTERSECTION DETECTION ALGORITHM

Suppose that the hierarchical representations of P and Q have already been computed. The intersection detection algorithm actually computes the *separation*, that is, the minimum distance between the two polyhedra. First consider the task of determining the separation between P and a triangle T in 3-space. We start with the top of the hierarchy, P_k. Because P_k and T are both of constant complexity, the separation between P_k and T can be computed in constant time. Given the separation between P_i and T, it is possible to determine the separation between P_{i-1} and T in constant time. This is done by a consideration of the newly added boundary elements of P_{i-1} that lie in the neighborhood of the two closest points.

Given the hierarchical decompositions of two polyhedra P and Q, the Dobkin-Kirkpatrick intersection algorithm begins by computing the separation at the highest common level of the two hierarchies (so that at least one of the decomposed polyhedra is of bounded complexity). They show that in $O(\log n_p + \log n_q)$ time it is possible to determine the separation of the polyhedra at the next lower level of the hierarchies. This leads to a total running time of $O(\log n_p \log n_q)$.

OPEN PROBLEM

Is it possible to detect the intersection of two preprocessed convex polyhedra in $O(\log(n_p + n_q))$ time?

33.3 INTERSECTION COUNTING AND REPORTING

GLOSSARY

Plane sweep: An algorithm paradigm based on simulating the left-to-right sweep of the plane with a vertical ***sweepline***. See Figure 33.3.1.

Event: In a plane-sweep algorithm, a condition at which the algorithm needs to update information in the ***active list*** of segments pierced by the sweepline, or produce output.

Red-blue intersection: Segment intersection between segments of two colors, where only intersections between segments of different colors are to be reported.

In many applications geometric intersections can be viewed as a discrete set of entities to be counted or reported. The problems of intersection counting and reporting have been heavily studied in computational geometry from the perspective of *intersection searching*, employing preprocessing and subsequent queries (Chapter 31). We limit our discussion here to batch problems, where the geometric objects are all given at once. In many instances, the best algorithms known for batch counting and reporting reduce the problem to intersection searching.

Table 33.3.1 summarizes a number of results on intersection counting and reporting. The quantity n denotes the combinatorial complexity of the objects, d denotes the dimension of the space, and k denotes the number of intersections. Because every pair of elements might intersect, the number of intersections k may generally be as large as $O(n^2)$, but it is frequently much smaller. The notation O^* means that polylogarithmic factors, those of the form $\log^c n$, have been ignored.

TABLE 33.3.1 Intersection counting and reporting.

PROBLEM	DIM	OBJECTS	TIME	SOURCE
Reporting	2	line segments	$n \log n + k$	[CE92]
Reporting	2	convex subdivisions	$n + k$	[GS87]
Reporting	d	orthogonal segments	$n \log^{d-1} n + k$	[EM81]
Counting	2	line segments	$O^*(n^{4/3})$	[Aga90]
Counting	d	orthogonal segments	$n \log^{d-1} n$	[EM81, Cha88]

REPORTING LINE SEGMENT INTERSECTIONS

Consider the problem of reporting the intersections of n line segments in the plane. This problem is an excellent vehicle for introducing the powerful technique of plane sweep (Figure 33.3.1). The plane-sweep algorithm maintains an active list of segments that intersect the current sweepline, sorted from bottom to top by intersection point. If two line segments intersect, then at some point prior to this intersection they must be consecutive in the sweep list. Thus, we need only test

consecutive pairs in this list for intersection, rather than testing all $O(n^2)$ pairs.

At each step the algorithm advances the sweepline to the next event: a line segment endpoint or an intersection point between two segments. Events are stored in a *priority queue* by their x-coordinates. After advancing the sweepline to the next event point, the algorithm updates the contents of the active list, tests new consecutive pairs for intersection, and inserts any newly-discovered events in the priority queue. For example, in Figure 33.3.1 the locations of the sweepline are shown with dashed lines.

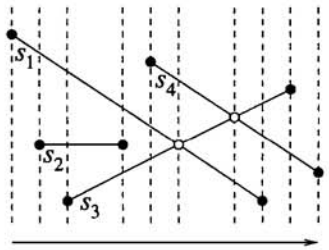


FIGURE 33.3.1
Plane sweep for line segment intersection.

Bentley and Ottmann showed that by using plane sweep it is possible to report all k intersecting pairs of n line segments in $O((n+k)\log n)$ time [BO79]. If the number of intersections k is much less than the $O(n^2)$ worst-case bound, then this is great savings over a brute-force test of all pairs.

For many years the question of whether this could be improved to $O(n \log n + k)$ was open, until Edelsbrunner and Chazelle presented such an algorithm [CE92]. This algorithm is optimal with respect to running time because at least $\Omega(k)$ time is needed to report the result, and it can be shown that $\Omega(n \log n)$ time is needed to detect whether there is any intersection at all. However, their algorithm uses $O(n+k)$ space. Clarkson and Shor presented a randomized algorithm with the same expected running time but using only $O(n)$ space [CS89].

If the line segments are taken from two sets, called say *red* and *blue*, and only intersections between segments of different colors are to be reported, then the problem is called a *red-blue intersection* problem. If the red segments form a convex subdivision of the plane, that is, a planar straight-line graph in which all the interior faces are convex polygons, then it is possible to report all intersections in $O(n+k)$ time [GS87]. (See Figure 33.3.2, where solid lines are red and dashed lines are blue.)

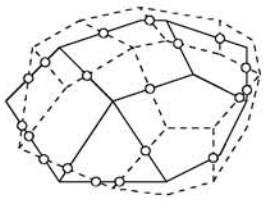


FIGURE 33.3.2
Intersection of convex subdivisions.

OPEN PROBLEMS

1. Is there an $O(n+k)$ algorithm for reporting all pairs of intersections between two simple polygons?
2. Is there a deterministic $O(n \log n + k)$ time and $O(n)$ space algorithm for reporting line segment intersections?

COUNTING LINE SEGMENT INTERSECTIONS

The most efficient techniques for counting line segment intersections are quite different from those used for reporting. Most methods use reductions to range searching and intersection searching. We will discuss only a few examples here. See Chapter 31 for more information.

Agarwal has shown that given a set of n line segments in the plane, it is possible to count the number of intersecting pairs in $O^*(n^{4/3})$ time [Aga90]. The same bound applies to red-blue intersection counting as well.

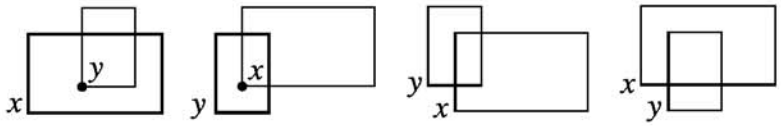
Among the most intriguing problems in this area is one posed by Hopcroft. Given a set of n points and n lines in the plane, does any point lie on any line? This is actually an intersection detection problem, but the best solutions are more closely related to solutions for intersection searching. Matoušek has shown that Hopcroft's problem can be solved in $O^*(n^{4/3})$ time [Mat93].

INTERSECTION COUNTING THROUGH RANGE SEARCHING

Range and intersection searching are powerful tools that can be applied to more complex intersection counting and reporting problems. This fact was first observed by Dobkin and Edelsbrunner [DE87], and has been applied to many other intersection searching problems since.

As an illustration, consider the problem of counting all intersecting pairs from a set of n rectangles. Edelsbrunner and Maurer [EM81] observed that intersections among orthogonal objects can be broken down to a set of orthogonal search queries (see Figure 33.3.3).

FIGURE 33.3.3
Four types of intersection between rectangles x and y.



For each rectangle x we can count all the intersecting rectangles of the set satisfying each of these conditions and sum them. Each of these counting queries can be answered in $O(\log n)$ time after $O(n \log n)$ preprocessing time [Cha88], leading

to an overall $O(n \log n)$ time algorithm. This counts every intersection twice and counts self intersections, but these are easy to factor out from the final result.

Generalizations to hyperrectangle intersection counting in higher dimensions are straightforward, with an additional factor of $\log n$ in time and space for each increase in dimension.

33.4 INTERSECTION CONSTRUCTION

GLOSSARY

Regularization: Discarding measure-zero parts of the result of an operation by taking the closure of the interior.

Clipping: Computing the intersection of each of many polygons with an axis-aligned rectangular viewing ***window***.

Kernel of a polygon: The set of points that can see every point of the polygon. (See Section 23.1.)

Intersection construction involves determining the region of intersection between geometric objects. It turns out that many of the same techniques that are used for computing geometric intersections are used for computing Boolean operations in general (e.g., union and difference). Many of the results presented here can be applied to these other problems as well. Typically intersection construction reduces to the following tasks: (1) compute the intersection between the boundaries of the objects; (2) if the boundaries do not intersect then determine whether one object is nested within the other; and (3) if the boundaries do intersect then classify the resulting boundary fragments and piece together the final intersection region.

When Boolean operations are computed on solid geometric objects, it is possible that lower-dimensional "dangling" components may result. It is common to eliminate these lower-dimensional components by a process called *regularization* (see Section 47.1). The ***regularized*** intersection of P and Q, denoted $P \cap^* Q$, is defined formally to be the closure of the interior of the standard intersection $P \cap Q$ (see Figure 33.4.1).

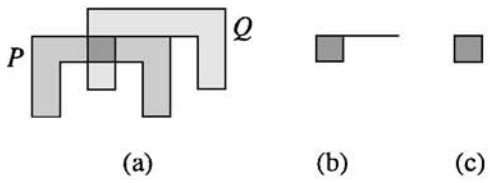


FIGURE 33.4.1
Regularized intersection: (a) *Polygons* P *and* Q*;* (b) $P \cap Q$; (c) $P \cap^* Q$.

Some results on intersection construction are summarized in Table 33.4.1, where n is the total complexity of the objects being intersected, and k is the number of pairs of intersecting edges.

TABLE 33.4.1 Intersection construction.

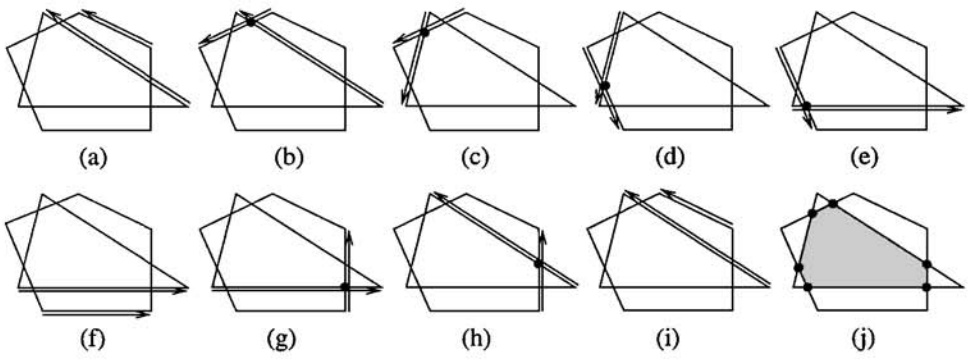
DIM	OBJECTS	TIME	SOURCE
2	convex-convex	n	[SH76, OCON82]
2	simple-simple	$n \log n + k$	[CE92]
2	kernel	n	[LP79]
3	convex-convex	n	[Cha92]

CONVEX POLYGONS

Determining the intersection of two convex polygons is illustrative of many intersection construction algorithms. Observe that the intersection of two convex polygons having a total of n edges is either empty or a convex polygon with at most n edges. O'Rourke et al. present an $O(n)$ time algorithm, which given two convex polygons P and Q determines their intersection [OCON82].

The algorithm can be viewed as a geometric generalization of merging two sorted lists. It performs a counterclockwise traversal of the boundaries of the two polygons. The algorithm maintains a pair of edges, one from each polygon. From a consideration of the relative positions of these edges the algorithm advances one of them to the next edge in counterclockwise order around its polygon. Intuitively, this is done in such a way that these two edges effectively "chase" each other around the boundary of the intersection polygon (see Figure 33.4.2(a)-(i)).

FIGURE 33.4.2
Convex polygon intersection construction.



OPEN PROBLEM

Reichling has shown that it is possible to detect whether m convex n-gons share a common point in $O(m \log^2 n)$ time [Rei88]. Is there an output-sensitive algorithm of similar complexity for constructing the intersection region?

SIMPLE POLYGONS AND CLIPPING

As with convex polygons, computing the intersection of two simple polygons reduces to first computing the points at which the two boundaries intersect and then classifying the resulting edge fragments. Computing the edge intersections and edge fragments can be performed by any algorithm for reporting line segment intersections. Classifying the edge fragments is a simple task. Margalit and Knott describe a method for edge classification that works not only for intersection, but for any Boolean operation on the polygons [MK89].

There is a special case of simple polygon intersection that is particularly important in computer graphics: clipping (Section 42.3). One popular algorithm for this problem is the Sutherland-Hodgman algorithm [FvD+90]. It works by intersecting each polygon with each of the four halfplanes that bound the clipping window. The algorithm traverses the boundary of the polygon, and classifies each edge as lying either entirely inside, entirely outside, or crossing each such halfplane.

An elegant feature of the algorithm is that it effectively "pipelines" the clipping process by clipping each edge against one of the window's four sides and then passing the clipped edge, if it is nonempty, to the next side to be clipped. This makes the algorithm easy to implement in hardware. An unusual consequence, however, is that if a polygon's intersection with the window has multiple connected components (as can happen with a nonconvex polygon), then the resulting clipped polygon consists of a single component connected by one or more "invisible" channels that run along the boundary of the window (see Figure 33.4.3).

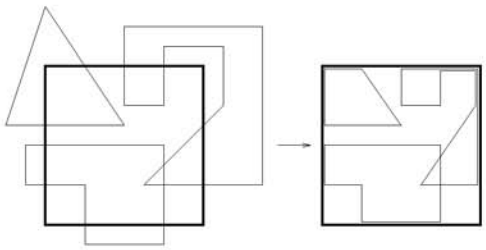


FIGURE 33.4.3
Clipping using the Sutherland-Hodgman algorithm.

INTERSECTION CONSTRUCTION IN HIGHER DIMENSIONS

Intersection construction in higher dimensions, and particularly in dimension 3, is important to many applications such as solid modeling. The basic paradigm of computing boundary intersections and classifying boundary fragments applies here as well, but good asymptotic results on the complexity of computing all the boundary intersections are harder to achieve.

There are some special cases where good solutions exist. One involves computing the intersection of two convex polyhedra in 3-space. Muller and Preparata gave an $O(n \log n)$ solution (see [PS85]). The existence of a linear-time algorithm remained open for years until Chazelle discovered such an algorithm [Cha92]. He showed that the Dobkin-Kirkpatrick hierarchical representation of polyhedra can be applied to the problem. A particularly interesting element of his algorithm is the use of the hierarchy for representing the interior of each polyhedron, and a dual hierarchy for representing the exterior of each polyhedron.

KERNELS AND THE INTERSECTION OF HALFSPACES

Because of the highly structured nature of convex polygons, algorithms for convex polygons can often avoid additional $O(\log n)$ factors that seem to be necessary when dealing with less structured objects. An example of this structure arises in computing the kernel of a simple polygon: the (possibly empty) locus of points that can see every point in the polygon (the shaded region of Figure 33.4.4). Put another way, the kernel is the intersection of inner halfplanes defined by all the sides of P. The kernel of P is a convex polygon having at most n sides. Lee and Preparata gave an $O(n)$ time algorithm for constructing it [LP79] (see also Table 25.3.1). Their algorithm operates by traversing the boundary of the polygon, and incrementally updating the boundary of the kernel as each new edge is encountered.

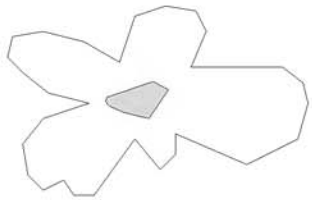


FIGURE 33.4.4
The kernel of a simple polygon.

The general problem of computing the intersection of halfplanes, when the halfplanes do not necessarily arise from the sides of a simple polygon, requires $\Omega(n \log n)$ time. See Chapter 19 for more information on this problem.

33.5 METHODS BASED ON SPATIAL SUBDIVISIONS

So far we have considered methods with proven worst-case asymptotic efficiency. However, there are numerous approaches to intersection problems for which worst-case efficiency is hard to establish, but that practical experience has shown to be quite efficient on the types of inputs that often arise in practice. Most of these methods are based on subdividing space into disjoint regions, or *cells*. Intersections can be computed by determining which objects overlap each cell, and then performing primitive intersection tests between objects that overlap the same cell.

GRIDS

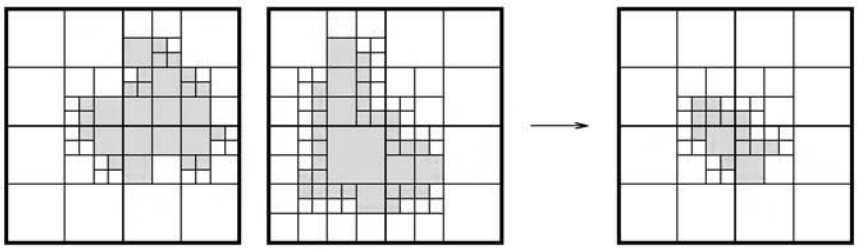
Perhaps the simplest spatial subdivision is based on "bucketing" with square grids. Space is subdivided into a regular grid of squares (or generally hypercubes) of equal side length. The side length is typically chosen so that either the total number of cells is bounded, or the expected number of objects overlapping each cell is bounded. Edahiro et al. showed that this method is competitive with and often performs much better than more sophisticated data structures for reporting intersections between randomly generated line segments in the plane [ETHA89]. Conventional wisdom is that grids perform well as long as the objects are small on average and their distribution is roughly uniform.

HIERARCHICAL SUBDIVISIONS

The principle shortcoming of grids is their inability to deal with nonuniformly distributed objects. Hierarchical subdivisions of space are designed to overcome this weakness. There is quite a variety of different data structures based on hierarchical subdivisions, but almost all are based on the principal of recursively subdividing space into successively smaller regions, until each region is sufficiently simple in the sense that it overlaps only a small number of objects. When a region is subdivided, the resulting subregions are its *children* in the hierarchy. Well-known examples of hierarchical subdivisions for storing geometric objects include quadtrees and k-d trees, R-trees, and binary space partition (BSP) trees. See [Sam90b] for a discussion of all of these.

Intersection construction with hierarchical subdivisions can be performed by a process of *merging* the two hierarchical spatial subdivisions. This method is described by Samet for quadtrees [Sam90a] and Naylor et al. [NAT90] for BSP trees. To illustrate the idea on a simple example, consider a quadtree representation of two black-and-white images. The problem is to compute the intersection of the two black regions. For example, in Figure 33.5.1 the two images on the left are intersected, resulting in the image on the right.

FIGURE 33.5.1
Intersection of images using quadtrees.



The algorithm recursively considers two overlapping square regions from each quadtree. A region of the quadtree is *black* if the entire region is black, *white* if the entire region is white, and *gray* otherwise. If either region is white, then the result is white. If either region is black, then the result is the other region. Otherwise both regions are gray, and we apply the procedure recursively to each of the four pairs of overlapping children.

33.6 SOURCES

Geometric intersections and related topics are covered in general sources on computational geometry. These include books by Preparata and Shamos [PS85], O'Rourke [O'R94], Edelsbrunner [Ede87], and Mehlhorn [Meh84]. Intersections of convex objects are discussed in the paper by Chazelle and Dobkin [CD87]. For information

on data structures useful for geometric intersections see Samet's books [Sam90a, Sam90b]. Sources on computing intersection primitives include O'Rourke's book on computational geometry [O'R94] and the book on computer graphics by Foley et al. [FvD+90]). For 3-dimensional surface intersections consult books on solid modeling, including those by Hoffmann [Hof89] and Mäntylä [Män88]. The *Graphics Gems* series (e.g., [Pae95]) contains a number of excellent tips and techniques for computing geometric operations including intersection primitives. See also Chapter 52 of this Handbook.

RELATED CHAPTERS

Chapter 19: Convex hull computations
Chapter 22: Triangulations
Chapter 31: Range searching
Chapter 32: Ray shooting and lines in space
Chapter 42: Computer graphics
Chapter 45: Splines and geometric modeling

REFERENCES

[Aga90] P.K. Agarwal. Partitioning arrangements of lines: II. Applications. *Discrete Comput. Geom.*, 5:533–573, 1990.

[BO79] J.L. Bentley and T.A. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Trans. Comput.*, C-28:643–647, 1979.

[CD87] B. Chazelle and D.P. Dobkin. Intersection of convex objects in two and three dimensions. *J. Assoc. Comput. Mach.*, 34:1–27, 1987.

[CE92] B. Chazelle and H. Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. *J. Assoc. Comput. Mach.*, 39:1–54, 1992.

[Cha88] B. Chazelle. A functional approach to data structures and its use in multidimensional searching. *SIAM J. Comput.*, 17:427–462, 1988.

[Cha91] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6:485–524, 1991.

[Cha92] B. Chazelle. An optimal algorithm for intersecting three-dimensional convex polyhedra. *SIAM J. Comput.*, 21:671–696, 1992.

[CS89] K.L. Clarkson and P.W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.

[DE87] D.P. Dobkin and H. Edelsbrunner. Space searching for intersecting objects. *J. Algorithms*, 8:348–361, 1987.

[DK83] D.P. Dobkin and D.G. Kirkpatrick. Fast detection of polyhedral intersection. *Theoret. Comput. Sci.*, 27:241–253, 1983.

[DK85] D.P. Dobkin and D.G. Kirkpatrick. A linear algorithm for determining the separation of convex polyhedra. *J. Algorithms*, 6:381–392, 1985.

[DK90] D.P. Dobkin and D.G. Kirkpatrick. Determining the separation of preprocessed polyhedra—a unified approach. In *Proc. 17th Internat. Colloq. Automata Lang. Program.*, volume 443 of *Lecture Notes in Comput. Sci.*, pages 400–413. Springer-Verlag, Berlin, 1990.

[DSV88] D.P. Dobkin, D.L. Souvaine, and C.J. Van Wyk. Decomposition and intersection of simple splinegons. *Algorithmica*, 3:473–486, 1988.

[Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monogr. Theoret. Comput. Sci.* Springer-Verlag, Heidelberg, 1987.

[EM81] H. Edelsbrunner and H.A. Maurer. On the intersection of orthogonal objects. *Inform. Process. Lett.*, 13:177–181, 1981.

[ETHA89] M. Edahiro, K. Tanaka, R. Hoshino, and Ta. Asano. A bucketing algorithm for the orthogonal segment intersection search problem and its practical efficiency. *Algorithmica*, 4:61–76, 1989.

[FvD+90] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, Reading, 1990.

[GS87] L.J. Guibas and R. Seidel. Computing convolutions by reciprocal search. *Discrete Comput. Geom.*, 2:175–193, 1987.

[Hof89] C. Hoffmann. *Geometric and Solid Modeling.* Morgan Kaufmann, San Mateo, 1989.

[HS95] J. Hershberger and S. Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. *J. Algorithms*, 18:403–431, 1995.

[HSS83] J.E. Hopcroft, J.T. Schwartz, and M. Sharir. Efficient detection of intersections among spheres. *Internat. J. Robot. Res.*, 2:77–80, 1983.

[LP79] D.T. Lee and F.P. Preparata. An optimal algorithm for finding the kernel of a polygon. *J. Assoc. Comput. Mach.*, 26:415–421, 1979.

[Män88] M. Mäntylä. *An Introduction to Solid Modeling.* Computer Science Press, Rockville, 1988.

[Mat93] J. Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.*, 10:157–182, 1993.

[Meh84] K. Mehlhorn. *Multi-dimensional Searching and Computational Geometry*, volume 3 of *Data Structures and Algorithms.* Springer-Verlag, Heidelberg, 1984.

[MK89] A. Margalit and G.D. Knott. An algorithm for computing the union, intersection or difference of two polygons. *Comput. & Graph.*, 13:167–183, 1989.

[Mou92] D.M. Mount. Intersection detection and separators for simple polygons. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 303–311, 1992.

[NAT90] B. Naylor, J.A. Amatodes, and W. Thibault. Merging BSP trees yields polyhedral set operations. *Comput. Graph.* (Proc. SIGGRAPH '90), 24:115–124, 1990.

[OCON82] J. O'Rourke, C.-B. Chien, T. Olson, and D. Naddor. A new linear algorithm for intersecting convex polygons. *Comput. Graph. Image Process.*, 19:384–391, 1982.

[O'R94] J. O'Rourke. *Computational Geometry in C.* Cambridge University Press, 1994.

[Pae95] A.W. Paeth, editor. *Graphics Gems V.* Academic Press, Boston, 1995.

[PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, 1985.

[Rei88] M. Reichling. On the detection of a common intersection of k convex objects in the plane. *Inform. Process. Lett.*, 29:25–29, 1988.

[Sam90a] H. Samet. *Applications of Spatial Data Structures.* Addison-Wesley, Reading, 1990.

[Sam90b] H. Samet. *The Design and Analysis of Spatial Data Structures.* Addison-Wesley, Reading, 1990.

[SH76] M.I. Shamos and D. Hoey. Geometric intersection problems. In *Proc. 17th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 208–215, 1976.

COMPUTATIONAL TECHNIQUES

34 RANDOMIZED ALGORITHMS

Ketan Mulmuley and Otfried Schwarzkopf

INTRODUCTION

Randomized (or probabilistic) algorithms and constructions were applied successfully in many areas of theoretical computer science before they were used widely in computational geometry. Following influential work in the mid-1980's, randomized algorithms became popular in geometry, and now a significant proportion of published research in computational geometry employs randomized algorithms or proof techniques. For many problems the best algorithms known are randomized, and even if both randomized and deterministic algorithms of comparable asymptotic complexity are available, the randomized algorithms are often much simpler and more efficient in an actual implementation. In some cases, the best deterministic algorithm known for a problem has been obtained by "derandomizing" a randomized algorithm.

This chapter focuses on the randomized algorithmic *techniques* being used in computational geometry, and not so much on particular results obtained using these techniques. Efficient randomized algorithms for specific problems are discussed in the relevant chapters throughout this Handbook.

GLOSSARY

Probabilistic or "Monte Carlo" algorithm: Traditionally, any algorithm that uses random bits. Now often used in contrast to *randomized algorithm* to denote an algorithm that is allowed to return an incorrect or inaccurate result, or fail completely, but with small probability. Monte Carlo methods for numerical integration provide an example. Algorithms of this kind are not used frequently in computational geometry.

Randomized or "Las Vegas" algorithm: An algorithm that uses random bits and is guaranteed to produce a correct answer; its running time and space requirements may depend on random choices. Typically, one tries to bound the expected running time (or other resource requirements) of the algorithm. In this chapter, we will only consider randomized algorithms in this sense.

Expected running time: The expected value of the running time of the algorithm, that is, the average running time over all possible choices of the random bits used by the algorithm. No assumptions are made about the distribution of input objects in space. When expressing bounds as a function of the input size, the worst case over all inputs of that size is given. Normally the random choices made by the algorithm are hidden from the outside, in contrast with average running time.

Average running time: The average of the running time, over all possible inputs. Some suitable distribution of inputs is assumed.

To illustrate the difference between expected running time and average running

time, consider the *Quicksort* algorithm. If it is implemented so that the pivot element is the first element of the list (and the assumed input distribution is the set of all possible permutations of the input set), then it has $O(n \log n)$ *average* running time. By providing a suitable input (here, a sorted list), an adversary can force the algorithm to perform worse than the average. If, however, Quicksort is implemented so that the pivot element is chosen at random, then it has $O(n \log n)$ *expected* running time, for any possible input. Since the random choices are hidden, an adversary cannot force the algorithm to behave badly, although it may perform poorly with some positive probability.

Randomized divide-and-conquer: A divide-and-conquer algorithm that uses a random sample to partition the original problem into subproblems (Section 34.1).

Randomized incremental algorithm: An incremental algorithm where the order in which the objects are examined is a random permutation (Section 34.2).

Tail estimate: A bound on the probability that a random variable deviates from its expected value. Tail estimates for the running time of randomized algorithms are useful but seldom available (Section 34.6).

High-probability bound: A strong tail estimate, where the probability of deviating from the expected value decreases as a fast-growing function of the input size n. The exact definition varies between authors, but a typical example would be to ask that for any $\alpha > 0$, there exists a $\beta > 0$ such that the probability that the random variable $X(n)$ exceeds $\alpha E[X(n)]$ be at most $n^{-\beta}$.

Derandomization: Obtaining a deterministic algorithm by "simulating" a randomized one (Section 34.8).

Trapezoidal map: A planar subdivision $\mathcal{T}(S)$ induced by a set S of line segments with disjoint interiors in the plane (cf. Section 30.4). $\mathcal{T}(S)$ can be obtained by passing vertical attachments through every endpoint of the given segments, extending upwards and downwards until each hits another segment, or extending to infinity; see Figure 34.0.1. Every face of the subdivision is a trapezoid (possibly degenerated to a triangle, or with a missing top or bottom side), hence the name.

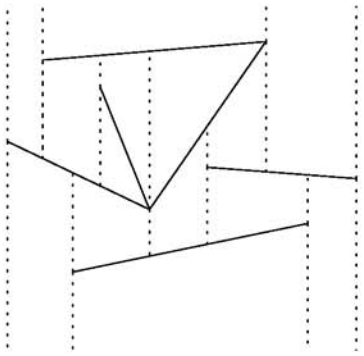


FIGURE 34.0.1
The trapezoidal map of a set of 6 *line segments.*

We will use the problem of computing the trapezoidal map of a set of line segments with disjoint interiors as a running example throughout this chapter. We assume for presentation simplicity that no two distinct endpoints have the same x-coordinate, so that every trapezoid is adjacent to at most four segments. (This can be achieved by slight rotation of the vertical direction.)

The trapezoidal map can also be defined for intersecting line segments. In that situation, vertical attachments must be added to intersection points as well, and the map may consist of a quadratic number of trapezoids. The trapezoidal map is also called the ***vertical decomposition*** of the set of line segments. Decompositions similar to this play an important role in randomized algorithms, because most algorithms assume that the structure to be computed has been subdivided into elementary objects.

34.1 RANDOMIZED DIVIDE-AND-CONQUER

GLOSSARY

Top-down sampling: Sampling with small, usually constant-size random samples, and recursing on the subproblems.

Cutting: A subdivision Ξ of space into simple cells Δ (of constant description complexity, most often simplices). The ***size*** of a cutting is the number of cells.

$1/r$-*cutting Ξ:* For a set X of n geometric objects, a cutting such that every cell $\Delta \in \Xi$ intersects at most n/r of the objects in X. See also Section 31.2.

Bottom-up sampling: Sampling with random samples large enough that the subproblems may be solved directly (without recursion).

Geometric problems lend themselves to solution by divide-and-conquer algorithms. It is natural to solve a geometric problem by dividing space into regions (perhaps with a grid), and solving the problem in every region separately. When the geometric objects under consideration are distributed uniformly over the space, then gridding or "slice-and-dice" techniques seem to work well. However, when object density varies widely throughout the environment, then the decomposition has to be fine in the areas where objects are abundant, while it may be coarse in places with low object density. Random sampling can help achieve this: the density of a random sample R of the set of objects will approach that of the original set. Therefore dividing space according to the sample R will create small regions where the geometric objects are dense, and larger regions that are sparsely populated.

We can distinguish two main types of randomized divide-and-conquer algorithm, depending on whether the size of the sample is rather small or quite large.

TOP-DOWN SAMPLING

Top-down sampling is the most common form of random sampling in computational geometry. It uses a random sample of small, usually constant, size to partition the problem into subproblems. We sketch the technique by giving an algorithm for the computation of the trapezoidal map of a set of segments in the plane.

Given a set S of n line segments with disjoint interiors, we take a sample $R \subset S$ consisting of r segments, where r is a constant. We compute the trapezoidal map $\mathcal{T}(R)$ of R. It consists of $O(r)$ trapezoids. For every trapezoid $\Delta \in \mathcal{T}(R)$, we determine the list S_Δ of segments in S intersecting Δ. We construct the trapezoidal

map of every set S_Δ recursively, clip it to the trapezoid Δ, and finally glue all these maps together to obtain $\mathcal{T}(S)$.

The running time of this algorithm can be analyzed as follows. Because r is a constant, we can afford to compute $\mathcal{T}(R)$ and the lists S_Δ naively, in time $O(r^2)$ and $O(nr)$ respectively. Gluing together the small maps can be done in time $O(n)$. But what about the recursive calls? If we denote the size of S_Δ by n_Δ, then bounding the n_Δ becomes the key issue here. It turns out that the right intuition is to assume that the n_Δ are about n/r. Assuming this, we get the recursion

$$T(n) \leq O(r^2 + nr) + O(r)T(n/r),$$

which solves to $T(n) = O(n^{1+\epsilon})$, where $\epsilon > 0$ is a constant depending on r. By increasing the value of r, ϵ can be made arbitrarily small, but at the same time the constant of proportionality hidden in the O-notation increases.

The truth is that one cannot really assume that $n_\Delta = O(n/r)$ holds for every trapezoid Δ at the same time. Valid bounds are as follows. For randomly chosen R of size r, we have:

- The ***pointwise bound:*** With probability increasing with r,

$$n_\Delta \leq C\frac{n}{r}\log r, \tag{34.1.1}$$

  for all $\Delta \in \mathcal{T}(R)$, where the constant C does not depend on r and n.

- The ***higher-moments bound:*** For any constant $c \geq 1$, there is a constant $C(c)$ (independent of r and n) such that

$$\sum_{\Delta \in \mathcal{T}(R)} (n_\Delta)^c = C(c)(\frac{n}{r})^c |\mathcal{T}(R)|. \tag{34.1.2}$$

In other words, while the maximum n_Δ can be as much as $O((n/r)\log r)$, on the average the n_Δ behave as if they indeed were $O(n/r)$.

Both bounds can be used to prove that $T(n) = O(n^{1+\epsilon})$, with the dependence on ϵ being somewhat better using the latter bound. The difference between the two bounds becomes more marked for larger values of r, as will be detailed below.

The same scheme used to compute $\mathcal{T}(S)$ will also give a data structure for point location in the trapezoidal map. This data structure is a tree, constructed as follows. If the set S is small enough, simply store $\mathcal{T}(S)$ explicitly. Otherwise, take a random sample R, and store $\mathcal{T}(R)$ in the root node. Subtrees are created for every $\Delta \in \mathcal{T}(R)$. These subtrees are constructed recursively, using the sets S_Δ.

By the pointwise bound, the depth of the tree is $O(\log n)$ with high probability, and therefore the query time is also $O(\log n)$. The storage requirement is easily seen to be $O(n^{1+\epsilon})$ as above.

The algorithmic technique described in this section is surprisingly robust. It works for a large number of problems in computational geometry, and for many problems it is the only known approach to solve the problem. It does have two major drawbacks, however. First, it seems to be difficult to remove the ϵ-term in the exponent, and truly optimal random-sampling algorithms are scarce. Second, the practicality of this method remains to be established. If the size of the random sample is chosen too small, then the problem size may not decrease fast enough to guarantee a fast-running algorithm, or even termination. Few papers in the literature calculate this size constant, and so for most applications it remains unclear

whether the size of the random sample can be chosen considerably smaller than the problem size in practice.

CUTTINGS

The only use of randomization in the above algorithm was to subdivide the plane into a number of simply-shaped regions Δ, such that every region is intersected by only a few line segments. Such a subdivision is called a *cutting* Ξ for the set X of n segments; if every $\Delta \in \Xi$ intersects at most n/r of the objects in X, it is a $1/r$-*cutting*. Cuttings are interesting in their own right, and have been studied intensively. This research has led to a number of ready-to-use results on the deterministic construction of efficient cuttings, with useful properties that go beyond those of the simple cutting based on a random sample discussed above. Cuttings form the basis for many algorithms and search structures in computational geometry; see Section 31.2. As a result, most recent geometric divide-and-conquer algorithms no longer explicitly use randomization, and randomized divide-and-conquer is currently in the process of being replaced by divide-and-conquer based on cuttings.

In practice, however, cuttings may still be constructed most efficiently using random sampling. There are two basic techniques, which we illustrate again using a set X of n line segments with disjoint interiors in the plane.

- ***ϵ-net based cuttings:*** The easiest way to obtain a $1/r$-cutting is to take a random sample $N \subset X$ of size $O(r \log r)$. If N is a $1/r$-net for the range space (X, Γ) (defined in Section 34.4 and Section 31.2), then the trapezoidal map of N is a $1/r$-cutting of size $O(r \log r)$. If not, we try a different sample.
- ***Splitting the excess:*** The construction based on ϵ-nets can be improved as follows. First take a random sample N of X of size $O(r)$, and compute its trapezoidal map. Every trapezoid Δ may be intersected by $O((n/r) \log r)$ segments. If we take a random sample of these segments, and form their trapezoidal map again (restricted to Δ), the pieces obtained are intersected by at most n/r segments. The size of this cutting is only $O(r)$, which is optimal.

BOTTOM-UP SAMPLING

In bottom-up sampling, the random sample is so large that the resulting subproblems are small enough to be solved directly. However, it is no longer trivial to compute the auxiliary structures needed to subdivide the problem. We again illustrate with the trapezoidal map.

Given a set S of n line segments, we take a sample R of size $n/2$, and compute the trapezoidal map of R recursively. For every $\Delta \in \mathcal{T}(R)$, we compute the list S_Δ of segments in $S \setminus R$ intersecting Δ. This can be done by locating an endpoint of every segment in $S \setminus R$ in $\mathcal{T}(R)$ and traversing $\mathcal{T}(R)$ from there. If we use a planar point location structure (Section 30.3), this takes time $O(n \log n + \sum_{\Delta \in \mathcal{T}(R)} n_\Delta)$. For every Δ, we then compute the trapezoidal map $\mathcal{T}(S_\Delta)$, and clip it to Δ. This can be done naively in time $O(n_\Delta^2)$. Finally, we glue together all the little maps.

The running time of the algorithm is bounded by the recursion

$$T(n) \leq T(n/2) + O(n \log n) + \sum_{\Delta \in \mathcal{T}(R)} O(n_\Delta^2).$$

The pointwise bound shows that with high probability, $n_\Delta = O(\log n)$ for all Δ. That would imply that the last term in the recursion is $O(n \log^2 n)$. Here, the higher-moments bound turns out to give a strictly better result, as it shows that the expected value of that term is only $O(n)$. The recursion therefore solves to $O(n \log^2 n)$.

Bottom-up sampling has the potential to lead to more efficient algorithms than top-down sampling, because it avoids the blow-up in problem size that manifests itself in the n^ϵ-term in top-down sampling. However, it needs more refined ingredients—as the constructions of $\mathcal{T}(R)$ and the lists S_Δ demonstrate—and therefore seems to apply to fewer problems.

As with top-down sampling, bottom-up sampling can be used for point location. These search structures have the advantage that they can often easily be made dynamic; see Section 34.3.

OPEN PROBLEM

The main open question in this area is "Is random sampling useful in practice?" This is a theoretically interesting question, as it involves calculating the constants in random sampling and cutting results.

34.2 RANDOMIZED INCREMENTAL ALGORITHMS

GLOSSARY

Backwards analysis: Analyzing the time complexity of an algorithm by viewing it running backwards in time.

Conflict graph: A bipartite graph whose arcs represent conflicts (usually intersections) between objects to be added and objects already constructed.

History graph: A directed, acyclic graph that records the history of changes in the geometric structure being maintained. Also known as an ***influence graph*** or ***I-DAG*** (influence-directed acyclic graph).

Many problems in computational geometry permit a natural computation by an incremental algorithm. Incremental algorithms need only process one new object at a time, which often implies that changes in the geometric data structure remain localized in the neighborhood of the new object.

As an example, consider the computation of the trapezoidal map of a set of line segments (for another example, see Section 19.3.1). To add a new line segment s to the map, one would first identify the trapezoids of the map intersected by s. Those trapezoids must be split, creating new trapezoids, some of which then must be merged along the segment s. All these update operations can be accomplished in time linear in the sum of the number of old trapezoids that are destroyed and the number of new trapezoids that are created during the insertion of s. This quantity is called the ***structural change***.

This results in a rather simple algorithm to compute the trapezoidal map of a set of line segments. Starting with the empty set, we treat the line segments

one-by-one, maintaining the trapezoidal map of the set of line segments inserted so far.

However, a general disadvantage of incremental algorithms is that the total structural change during the insertions of n objects, and hence the running time of the algorithm, depends strongly on the order in which the objects are processed. In our case, it is not difficult to devise a sequence of n line segments leading to a total structural change of $\Theta(n^2)$. Even if we know that a good order of insertion exists (one that implies a small structural change), it seems difficult to determine this order beforehand. And this is exactly where randomization can help: we simply treat the n objects in random order. In the case of the trapezoidal map, we will show below that if the n segments are processed in random order, the *expected* structural change in every step of the algorithm is only constant.

BACKWARDS ANALYSIS

An easy way to see this is via *backwards analysis*. We first observe that it suffices to bound the number of trapezoids created in each stage of the algorithm. All these trapezoids are incident to the segment inserted in that stage. We imagine the algorithm removing the line segments from the final map one-by-one. In each step, we must bound the number of trapezoids incident to the segment s removed. Now we make two observations:

- The trapezoidal map is a planar graph, with every trapezoid incident to at most 4 segments. Hence, if there are m segments in the current set, the total number of trapezoid-segment incidences is $O(m)$.
- Since the order of the segments is a random permutation of the set of segments, each of the m segments is equally likely to be removed.

These two facts suffice to show that the expected number of trapezoids incident to s is constant. In fact, this number is bounded by the average degree of a segment in a trapezoidal map.

It follows that the expected total structural change during the course of the algorithm is $O(n)$. To obtain an efficient algorithm, however, we need a second ingredient: whenever a new segment s is inserted, we need to identify the trapezoids of the old map intersected by s. Two basic approaches are known to solve this problem: the conflict graph and the history graph.

CONFLICT GRAPH

A conflict graph is a bipartite graph whose nodes are the not-yet-added segments on one side and the trapezoids of the current map on the other side. There is an arc between a segment s and a trapezoid Δ if and only if s intersects Δ, in which case we say that s *is in conflict with* Δ (because the two cannot co-exist in the same trapezoidal map).

It is possible to maintain the conflict graph during the course of the incremental algorithm. Whenever a new segment is inserted, all the conflicts of the newly-created trapezoids are found. This is not difficult, because a segment can only conflict with a newly-created trapezoid if it was previously in conflict with the old

trapezoids at the same place. Thus the trapezoids intersected by the new segment s are just the neighbors of s in the conflict graph.

The time necessary to maintain the conflict graph can be bounded by summing the number of conflicts of all trapezoids created during the course of the algorithm. It follows from the higher moments-bound (Equation 34.1.2) that the average number of conflicts of the trapezoids present after inserting the first r segments—note that these segments form a random sample of size r of S—is $O(n/r)$. Intuitively, we can assume that this is also correct if we look only at the trapezoids that are *created* by the insertion of the rth segment. Since the expected number of trapezoids created in every step of the algorithm is constant, the expected total time is $\sum_{i=1}^{n} O(n/r) = O(n \log n)$.

Note that an algorithm using a conflict graph needs to know the entire set of objects (segments in our example) in advance.

HISTORY GRAPH

A different approach uses a history graph, which records the history of changes in the maintained structure.

In our example, we can maintain a directed acyclic graph whose nodes correspond to trapezoids constructed during the course of the algorithm. The leaves are the trapezoids of the current map; all inner nodes correspond to trapezoids that have already been destroyed (with the root corresponding to the entire plane). When we insert a segment s, we create new nodes for the newly-created trapezoids, and create a pointer from an old trapezoid to every new one that overlaps it. Hence, there are at most four outgoing pointers for every inner node of the history graph.

We can now find the trapezoids intersected by a new segment s by performing a graph search from the root, using say, depth-first search on the connected subgraph consisting of all trapezoids intersecting s. Note that this search performs precisely the same computations that would have been necessary to maintain the conflict graph during the sequence of updates, but at a different time. We can therefore consider a history graph as a lazy implementation of a conflict graph: it postpones each computation to the moment it is actually needed. Consequently, the analysis is exactly the same as for conflict graphs.

Algorithms using a history graph are ***on-line*** or ***semidynamic*** in the sense that they do not need to know about a point until the moment it is inserted.

ABSTRACT FRAMEWORK AND ANALYSIS

Most randomized incremental algorithms in the literature follow the framework sketched here for the computation of the trapezoidal map: the structure to be computed is maintained while the objects defining it are inserted in random order. To insert a new object, one first has to find a "conflict" of that object (the *location step*), then local updates in the structure are sufficient to bring it up to date (the *update step*). The cost of the update is usually linear in the size of the structural change (the change in the combinatorial structure being maintained), and can often be bounded using backwards analysis. The location step can be implemented using either a conflict graph or a history graph. In both cases, the analysis is the same (since the actual computations performed are also often identical). To avoid

having to prove the same bounds repeatedly for different problems, researchers have defined an axiomatic framework that captures the combinatorial essence of most randomized incremental algorithms. This framework, which uses *configuration spaces*, provides ready-to-use bounds for the expected running time of most randomized incremental algorithms. See Section 34.5.

POINT LOCATION THROUGH HISTORY GRAPH

In our trapezoidal map example, the history graph may be used as a point location structure for the trapezoidal map: given a query point q, find the trapezoid containing q by following a path from the root to a leaf node of the history graph. At each step, we continue to the child node corresponding to the trapezoid containing q.

The search time is clearly proportional to the length of the path. Backwards analysis shows that the expected length of this path is $O(\log n)$ for any fixed query point. Even stronger, one can show that the maximum depth of the history graph is $O(\log n)$ with high probability.

If point location is the goal, the history graph can be simplified: instead of storing trapezoids, the inner nodes of the graph can denote two different kinds of elementary tests ("Does a point lie to the left or right of another point?" and "Does a point lie above or below a line?"). The final result is then an efficient and practical planar point location structure.

This observation can also lead to a somewhat different location step inside the randomized incremental algorithm. Instead of performing a graph search with the whole segment s, point location can be used to find the trapezoid containing one endpoint of s. From there, a traversal of the trapezoidal map allows locating all trapezoids intersected by s.

APPLICATIONS

The randomized incremental framework has been successfully applied to a large variety of problems. We list a number of important such applications. Details on the results can be found in the chapters dealing with the respective area, or in one of the surveys cited in Section 34.9.

- Trapezoidal decomposition formed by segments in the plane, and point location structures for this decomposition (Section 30.3).
- Triangulation of simple polygons: a simple randomized algorithm with running time $O(n \log^* n)$ (Section 23.2).
- Convex hulls of points in d-dimensional space, output-sensitive convex hulls in $\mathbb{R}^3$ (Section 19.3).
- Voronoi diagrams in different metrics, including higher order and abstract Voronoi diagrams (Section 20.3).
- Linear programming in finite-dimensional space (Chapter 38).
- Generalized linear programming: optimization problems that are combinatorially similar to linear programming (Section 38.4).
- Hidden surface removal (Section 25.6 and Chapter 42).
- Constructing a single face in an arrangement of (curved) segments in the

plane, or in an arrangement of triangles or surface patches in $\mathbb{R}^3$ (Sections 21.5 and 40.2); computing zones in an arrangement of hyperplanes in d-space (Section 21.4).

34.3 DYNAMIC ALGORITHMS

DYNAMIC RANDOMIZED INCREMENTAL

Any on-line randomized incremental algorithm can be used as a semidynamic algorithm, a dynamic algorithm that can only perform insertions of objects. The bound on the expected running time of the randomized incremental algorithm then turns into a bound on the *average* running time, under the assumption that every permutation of the input is equally likely. (The relation between the two uses of the algorithms is similar to that between randomized and ordinary Quicksort as mentioned in the Introduction.)

This observation has motivated researchers to extend randomized incremental algorithms so that they can also manage deletions of objects. Then bounds on the average running time of the algorithm are given, under the assumption that the input sequence is a *random update sequence*. In essence, one assumes that for an addition, every object currently not in the structure is equally likely to be inserted, while for a deletion every object currently present is equally likely to be removed (the precise definition varies between authors).

Two approaches have been suggested to handle deletions in history-graph based incremental algorithms. The first adds new nodes at the leaf level of the history graph for every deletion. This works for a wide variety of problems and is relatively easy to implement, but after a number of updates the history graph will become "dirty": it will contain elements that are no longer part of the current structure but which still must be traversed by the point location steps. Therefore, the history graph needs periodic "cleaning." This can be accomplished by discarding the current graph, and reconstructing it from scratch using the elements currently present.

In the second approach, for every deletion the history graph is transformed to the state it would have been had the object never been inserted. The history graph is therefore always "clean." However, in this model deletions are more complicated, and it therefore seems to apply to fewer problems.

DYNAMIC SAMPLING

A rather different approach permits a number of search structures based on bottom-up sampling to be dynamized surprisingly easily. Such a search structure consists of a hierarchy of $O(\log n)$ levels. Every object is included in the first level, and is chosen independently to be in the second level with probability 1/2. Every object in the second level is propagated to level 3 with probability 1/2, and so forth. Whenever an object is added to or removed from the current set, the search structure is updated to the proper state. When adding an object, it suffices to

flip a coin at most $\log n$ times to determine where to place the object. Using this technique, it is possible to give high-probability bounds on the search time and sometimes also on the update time.

34.4 RANGE SPACES

"Pointwise bounds" of the form in Equation 34.1.1 can be proven in the axiomatic framework of range spaces, which then leads to immediate application to a wide variety of geometric settings.

GLOSSARY

Range space: A pair (X, Γ), with X a universe (possibly infinite), and Γ a family of subsets of X. The elements of Γ are called ***ranges***. Typical examples of range spaces are of the form $(\mathbb{R}^d, \Gamma)$, where Γ is a set of geometric figures, such as all line segments, halfspaces, simplices, balls, etc. (Cf. Section 31.2.)

Shattered: A set $A \subseteq X$ is shattered if every subset A' of A can be expressed as $A' = A \cap \gamma$, for some range $\gamma \in \Gamma$.

In the range space $(\mathbb{R}^2, \mathcal{H})$, where $\mathcal{H}$ is the set of all closed halfplanes, a set of three points in convex position is shattered. However, no set of four points is shattered. See Figure 34.4.1: whether the point set is in convex position or not, there always is a subset (encircled) that cannot be expressed as $A \cap h$ for any halfplane h.

FIGURE 34.4.1
No set of four points can be shattered by halfplanes.

In the range space $(\mathbb{R}^2, \mathcal{C})$, where $\mathcal{C}$ is the set of all convex polygons, any set of points lying on a circle is shattered.

Vapnik-Chervonenkis dimension (VC-dimension): The VC-dimension of a range space (X, Γ) is the smallest integer d such that there is no shattered subset $A \subseteq X$ of size $d + 1$. If no such d exists, the VC-dimension is said to be infinite.

Range spaces $(\mathbb{R}^d, \Gamma)$, where Γ is the set of line segments, of simplices, of balls, or of halfspaces, have finite VC-dimension. For example, the range space $(\mathbb{R}^2, \mathcal{H})$ has VC-dimension 3. The range space $(\mathbb{R}^2, \mathcal{C})$, however, has infinite VC-dimension.

Shatter function: For a range space (X, Γ), the shatter function $\pi_\Gamma(m)$ is defined as

$$\pi_\Gamma(m) = \max_{A \subset X, |A| = m} |\{A \cap \gamma \mid \gamma \in \Gamma\}| .$$

If the VC-dimension of the range space is infinite, then $\pi_\Gamma(m) = 2^m$. Otherwise the shatter function is bounded by $O(m^d)$, where d is the VC-dimension. (So the shatter function of any range space is either exponential or polynomially bounded.) If the shatter function is polynomial, the VC-dimension is finite. The order of magnitude of the shatter function is not necessarily the same as the VC-dimension; for instance, the range space $(\mathbb{R}^2, \mathcal{H})$ has VC-dimension 3 and shatter function $O(m^2)$.

ϵ-net: A subset $N \subseteq X$ is called an ϵ-net for the range space (X, Γ) if $N \cap \gamma \neq \emptyset$ for every $\gamma \in \Gamma$ with $|\gamma|/|X| > \epsilon$ (here, $\epsilon \in [0, 1)$ and X is finite). It is often more convenient to write $1/r$ for ϵ, with $r > 1$.

ϵ-approximation: A subset $A \subseteq X$ is called an ϵ-approximation for the range space (X, Γ) if, for every $\gamma \in \Gamma$, we have

$$\left| \frac{|A \cap \gamma|}{|A|} - \frac{|\gamma|}{|X|} \right| \leq \epsilon \, .$$

An ϵ-approximation is also an ϵ-net, but not necessarily vice versa.

ϵ-NETS AND ϵ-APPROXIMATIONS

The pointwise bound translates into the abstract framework of range spaces as follows (cf. Theorem 31.2.2):

THEOREM 34.4.1

Let (X, Γ) be a range space with X finite and of finite VC-dimension d. Then a random sample $R \subset X$ of size $C(d)r \log r$ is a $1/r$-net for (X, Γ) with probability whose complement to 1 is exponentially small in r. The constant $C(d)$ depends only on d.

This theorem forms the basis for "traditional" randomized divide-and-conquer algorithms, such as the one for the trapezoidal map of line segments sketched in Section 34.1. The pointwise bound used there follows from the theorem. Consider the range space (S, Γ), where $\Gamma := \{\gamma(\Delta) \mid \Delta \text{ an open trapezoid}\}$, and $\gamma(\Delta)$ is the set of all segments in S intersecting Δ. The VC-dimension of this range space is finite. The easiest way to see this is by looking at the shatter function. Consider a set of m line segments. Extend them to full lines, pass $2m$ vertical lines through all endpoints, and look at the arrangement of these $3m$ lines. Clearly, for any two trapezoids Δ and Δ' whose corners lie in the same faces of this arrangement we have $\gamma(\Delta) = \gamma(\Delta')$. Consequently, there are at most $O(m^8)$ different ranges, and that crudely bounds the shatter function as $O(m^8)$. Thus the VC-dimension is finite and Theorem 34.4.1 applies: with probability increasing rapidly with r, the sample R of size r is an ϵ-net for S with $\epsilon = \Omega((1/r) \log r)$. Assume this is the case, and consider some trapezoid $\Delta \in \mathcal{T}(R)$. The interior of Δ does not intersect any segment in R, so by the property of ϵ-nets, the range $\gamma(\Delta)$ can intersect at most ϵn segments of S. And so we have $n_\Delta = O((n/r) \log r)$.

The construction of ϵ-nets has been so successfully derandomized that ϵ-nets now are used routinely in deterministic algorithms. At least in theory, the top-down sampling algorithm of Section 34.1 need no longer be considered a randomized algorithm.

ϵ-approximations are important for the deterministic computation of ϵ-nets. They are also interesting in their own right, since some geometric problems—for instance, the computation of centerpoints or ham-sandwich cuts (see Section 11.2)—can be solved approximately by solving them exactly for an ϵ-approximation. A $1/r$-approximation can be found by taking a random sample of size $O(r^2 \log r)$. This bound is not tight.

VC-dimension and ϵ-nets are also frequently used in statistics (from which they derive) and in learning theory.

OPEN PROBLEM

For general range spaces, the bound $O(r \log r)$ in Theorem 34.4.1 is the best possible. However, for many geometrically-defined spaces the best lower bound is $\Omega(r)$. Can the upper bound be improved for some geometric range spaces? This is perhaps a difficult problem [MSW90].

34.5 CONFIGURATION SPACES

The framework of configuration spaces is somewhat more complicated than range spaces, but facilitates proving higher-moment bounds as in Equation 34.1.2. Terminology, axiomatics, and notation vary widely between authors.

GLOSSARY

Configuration space: A four-tuple $(X, \mathcal{T}, D, K)$. X is a finite set of geometric objects (the universe of size n). $\mathcal{T}$ is a mapping that assigns to every subset $S \subseteq X$ a set $\mathcal{T}(S)$; the elements of $\mathcal{T}(S)$ are called ***configurations***. $\Pi(X) := \bigcup_{S \subseteq X} \mathcal{T}(S)$ is the set of all configurations occuring over some subset of X. D and K assign to every configuration $\Delta \in \Pi(X)$ subsets $D(\Delta)$ and $K(\Delta)$ of X. Elements of the set $D(\Delta)$ are said to ***define*** the configuration (they are also called ***triggers***) and the elements of the set $K(\Delta)$ are said to ***kill*** the configuration (they are also said to be in ***conflict*** with the configuration and are sometimes called ***stoppers***).

Conflict size of Δ: The number of elements of $K(\Delta)$.

We will require the following axioms:

(i) The number $d = \max\{\, |D(\Delta)| \;\mid\; \Delta \in \Pi(X)\}$ is a constant (called the ***maximum degree*** or the ***dimension*** of the configuration space). Moreover, the number of configurations sharing the same defining set is bounded by a constant.

(ii) For any $\Delta \in \mathcal{T}(S)$, $D(\Delta) \subseteq S$ and $S \cap K(\Delta) = \emptyset$.

(iii) If $\Delta \in \mathcal{T}(S)$ and $D(\Delta) \subseteq S' \subseteq S$, then $\Delta \in \mathcal{T}(S')$.

(iii′) If $D(\Delta) \subseteq S$ and $K(\Delta) \cap S = \emptyset$, then $\Delta \in \mathcal{T}(S)$.

Note that axiom (iii) follows from (iii′); see below.

EXAMPLES

1. *Trapezoidal map.* The universe X is a set of segments in the plane, and $\mathcal{T}(S)$ is the set of trapezoids in the trapezoidal map of S. The defining set $D(\Delta)$ is the set of segments that are necessary to define Δ (at most four segments suffice, so $d = 4$), and the killing set $K(\Delta)$ is the set of segments that intersect the trapezoid. It is easy to verify that conditions (i), (ii), (iii), (iii′) all hold.

2. *Delaunay triangulation.* X is a set of points in the plane (assume that no four points lie on a circle), and $\mathcal{T}(S)$ is the set of triangles of the Delaunay triangulation of S. $D(\Delta)$ consists of the vertices of triangle Δ (so $d = 3$), while $K(\Delta)$ is the set of points lying inside the circumcircle of the triangle. Again, axioms (i), (ii), (iii), (iii′) all hold.

3. *Convex hulls in 3D.* The universe X is a set of points in 3D (assume that no four points are coplanar), and $\mathcal{T}(S)$ is the set of facets of the convex hull of S. The defining set of a facet Δ is the set of its vertices ($d = 3$), and the killing set is the set of points lying in the outer open halfspace defined by Δ. Note that there can be two configurations sharing the same defining set. Again, axioms (i)–(iii′) all hold.

4. *Single cell.* The universe X is a set of possibly intersecting segments in the plane, and $\mathcal{T}(S)$ is the set of trapezoids in the trapezoidal map of S that belongs to the cell of the line segment arrangement containing the origin (Figure 34.5.1). The defining and killing sets are defined as in the case of the trapezoidal map of the whole arrangement above. In this situation, axiom (iii′) does not hold. Whether or not a given trapezoid appears in $\mathcal{T}(S)$ depends on segments other than the ones in $D(\Delta) \cup K(\Delta)$. Axioms (i), (ii), (iii) are nevertheless valid.

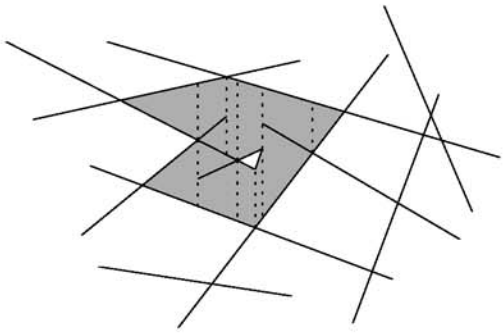


FIGURE 34.5.1
A single cell in an arrangement of line segments.

5. *Counterexample.* Let X be a set of line segments, and let $\mathcal{T}(S)$ be a decomposition of the arrangement that is obtained by drawing vertical extensions for faces with an even number of edges, and horizontal extensions for faces with an odd number of edges. Axioms (i) and (ii) hold, but neither (iii) nor (iii′) is satisfied.

Note that when (ii) and (iii′) both hold, then $\Delta \in \mathcal{T}(S)$ if and only if $D(\Delta) \subseteq S$ and $K(\Delta) \cap S = \emptyset$. In other words, the mapping $\mathcal{T}$ is then completely defined by

the functions D and K. In fact, in the first three examples we can decide from local information alone whether or not a configuration appears in $\mathcal{T}(S)$. For instance, a triangle Δ is in the Delaunay triangulation of S if and only if the vertices of Δ are in S, and no point of S lies in the circumcircle of Δ.

As mentioned above, axiom (iii) follows from (iii′), but not conversely. Axiom (iii) requires a kind of monotonicity: if Δ occurs in $\mathcal{T}(S)$ for some S, then we cannot destroy it by removing elements from S unless we remove some element in $D(\Delta)$.

We may say that the configuration spaces of the first three examples are defined *locally* and *canonically*. The fourth example is *canonical*, but *nonlocal*. The last example is not canonical and cannot be treated with the methods described here. (Fortunately, this is an artificial example with no practical use—but see the open problems below.)

HIGHER-MOMENTS BOUND

THEOREM 34.5.1

Let $(X, \mathcal{T}, D, K)$ be a configuration space satisfying axioms (i), (ii), (iii), and let R be a random sample of X of size r. For any constant c, we have

$$E[\sum_{\Delta \in \mathcal{T}(S)} |K(\Delta)|^c] = O((n/r)^c E[|\mathcal{T}(S)|]).$$

In other words, as far as the cth-degree average is concerned, the conflict size behaves as if it were $O(n/r)$ (instead of $O((n/r)\log r)$ from the pointwise bound). The specific bound for trapezoidal maps, Equation 34.1.2, follows immediately.

RANDOMIZED INCREMENTAL CONSTRUCTION

Many, if not most, randomized incremental algorithms in the literature can be analyzed using the configuration space framework. Given the set X, the goal of the randomized incremental algorithm is to compute $\mathcal{T}(X)$. This is done by maintaining $\mathcal{T}(X^i)$, for $1 \leq i \leq n$, where $X^i = \{x_1, x_2, \ldots, x_i\}$ and the x_i form a random permutation of X.

To bound the number of configurations created during the insertion of x_i into X^{i-1}, we observe that by axiom (iii) these configurations are exactly those $\Delta \in \mathcal{T}(X^i)$ with $x_i \in D(\Delta)$. The expected number of these can be bounded by

$$\frac{d}{i} E[|\mathcal{T}(X^i)|]$$

using backwards analysis. Here, d is the maximum degree of the configuration space.

The expected total change in the conflict graph or history graph can be bounded by summing $|K(\Delta)|$ over all Δ created during the course of the algorithm. Using axioms (i) to (iii′), we can derive the following bound:

$$\sum_{i=1}^{n} d^2 \frac{n-i}{i} \frac{E[|\mathcal{T}(X^i)|]}{i}.$$

(The exact form of this expression depends on the model used.) The book [Mul94] treats randomized incremental algorithms systematically using the configuration space framework (assuming axiom (iii′)).

LAZY RANDOMIZED INCREMENTAL CONSTRUCTION

In problems that have nonlocal definition, such as the computation of a single cell in an arrangement of segments, single cells in arrangements of surface patches, or zones in arrangements, the update step of a randomized incremental construction becomes more difficult. Besides the local updates in the neighborhood of the newly inserted object, there may also be global changes. For instance, when a line segment is inserted into an arrangement of line segments, it may cut the single cell being computed into several pieces, only one of which is still interesting. The technique of lazy randomized incremental construction [dBDS94] deals with these problems by simply postponing the global changes to a few "clean-up" stages. Since the setting of all these problems is nonlocal, the analysis uses only axioms (i), (ii), (iii).

OPEN PROBLEM

The canonical framework of randomized incremental algorithms sketched above is sometimes too restrictive. For instance, to make a problem fit into the framework, one often has to assume that objects are in general position. While many algorithms could deal with special cases (e.g., four points on a circle in the case of Delaunay triangulations) directly, the analysis does not hold for those situations, and one has to resort to a symbolic perturbation scheme to save the analysis. Can a more relaxed framework for randomized incremental construction be given [Sei93, MSS96]?

34.6 BETTER GUARANTEES

Bounds for the expected performance of randomized algorithms are usually available. Sometimes stronger results are desired. If the analysis of the algorithm cannot be extended to provide such bounds, then some techniques may help to achieve them:

Randomized space vs. deterministic space. Any randomized algorithm using expected space S and expected time T can be converted to an algorithm that uses deterministic space $2S$, and whose expected running time is at most $2T$. We simply need to maintain a count of the memory allocated by the algorithm. Whenever it exceeds $2S$, we stop the computation and restart it again with fresh choices for the random variables. The expected number of retrials is one.

Tail estimates. Note that the knowledge that the expected running time of a given program is one second does not exclude the possibility that it sometimes takes one hour. Markov's inequality implies that the probability that this happens is at most 1/3600. While this seems innocuous, it implies that it is likely to occur if we repeat this particular computation, say, 10000 times.

A simple modification to the algorithm can yield stronger bounds. We run it for two seconds. If it does not finish the computation within two seconds, then we

abandon the computation and restart with fresh choices for the random variables. Clearly, the probability that the algorithm does not terminate within one hour is at most 2^{-1800}. Alt et al. [AGM+] work out this technique in detail.

34.7 PROBABILISTIC PROOF TECHNIQUES

Randomized algorithms are related to probabilistic proofs and constructions in combinatorics, which precede them historically. Conversely, the concepts developed to design and analyze randomized algorithms in computational geometry can be used as tools in proving purely combinatorial results. Many of these results are based on the following theorem:

THEOREM 34.7.1

Let $(X, \mathcal{T}, D, K)$ be a configuration space satisfying axioms (i), (ii), (iii), and (iii') of Section 34.5. For $S \subseteq X$ and $0 \leq k \leq n$, let

$$\Pi^k(S) := \{\Delta \in \Pi(X) \mid |K(\Delta) \cap S| \leq k\}$$

denote the set of configurations with at most k conflicts in S.

Then $|\Pi^k(S)| = O(k^d)E[|\mathcal{T}(R)|]$, where R is a random sample of S of size n/k, and d is the maximum degree (or dimension) of the configuration space, $d = \max|D(\Delta)|$.

Note that $\Pi^0(S) = \mathcal{T}(S)$. The theorem relates the number of configurations with at most k conflicts to those without conflict.

An immediate application is to prove a bound on the number of vertices of level at most k in an arrangement of lines in the plane (the level of a vertex is the number of lines lying above it; see Section 21.2). We define a configuration space $(X, \mathcal{T}, D, K)$ where X is the set of lines, $\mathcal{T}(S)$ is the set of vertices of the upper envelope of the lines, $D(\Delta)$ are the two lines forming the vertex Δ (so $d = 2$), and $K(\Delta)$ is the set of lines lying above Δ. Theorem 34.7.1 implies that the number of vertices of level up to k is bounded by $O(nk)$. The same argument works in any dimension.

Sharir and others have proven a number of combinatorial results using this technique [Sha94, AES95, ASS96, SS96]. They define a configuration space and need to bound $|\mathcal{T}(S)|$. They do this by proving a geometric relationship between the configurations with zero conflicts (the ones appearing in $\mathcal{T}(S)$) and the configurations with at most k conflicts. Applying Theorem 34.7.1 yields a recursion that bounds $|\mathcal{T}(S)|$ in terms of $|\mathcal{T}(R)|$. A refined approach that uses a sample of size $n - 1$ (instead of n/k) has been suggested by Tagansky [Tag95].

34.8 DERANDOMIZATION

Even when an efficient randomized algorithm for a problem is known, researchers still find it worthwhile to obtain a deterministic algorithm of the same efficiency. The reasons for doing this are varied, from scientific curiosity (what is the real power of randomness?), to practical reasons (truly random bits are quite expensive), to a

preference for "deterministicity" that may not be strictly rational.

Sometimes a deterministic algorithm for a given problem may be obtained by "simulating" or "derandomizing" a randomized algorithm. A deterministic algorithm obtained this way often appears quite mysterious unless one appeals to the underlying randomized algorithm.

Derandomization has turned out to be a powerful tool; the only known worst-case optimal deterministic algorithm for computing the convex hull of n points in d-dimensional space, for instance, is obtained by derandomizing a randomized algorithm (Section 19.3.1).

At the current state of the art, randomized divide-and-conquer algorithms (and data structures based on random samples) in fixed dimension can usually be derandomized with at most a slight decrease in asymptotic efficiency. Randomized incremental algorithms are harder to derandomize. Efficient derandomization of algorithms for nonconstant dimension remains elusive. Randomized algorithms for linear programming still beat the best deterministic ones known by a factor exponential in d.

METHOD OF CONDITIONAL PROBABILITIES

The ***method of conditional probabilities*** (also called the ***Raghavan-Spencer method***) implements a binary search of the original probability space. Assume that an algorithm uses n random bits $X = (x_1, x_2, \ldots, x_n)$, and let $T(X)$ be the running time of the algorithm assuming a particular choice of bits. Our goal is to find a particular vector X^* such that $T(X^*) \leq E[T]$. The first step is to determine a value x_1^* such that $E[T \mid x_1 = x_1^*] \leq E[T]$. This is done by evaluating the two conditional expectancies $E[T \mid x_1 = 0]$ and $E[T \mid x_1 = 1]$ and choosing the x_1^* minimizing this expectancy. We then continue to choose values x_i^* for x_i, by evaluating the two conditional expectancies

$$E[T \mid (x_1 = x_1^*), (x_2 = x_2^*), \ldots, (x_{i-1} = x_{i-1}^*), (x_i = 0)]$$

and

$$E[T \mid (x_1 = x_1^*), (x_2 = x_2^*), \ldots, (x_{i-1} = x_{i-1}^*), (x_i = 1)].$$

It can be proven that this process guarantees that at any stage,

$$E[T \mid (x_1 = x_1^*), (x_2 = x_2^*), \ldots, (x_i = x_i^*)] \leq E[T]\,,$$

so that at termination we have found a vector X^* such that $T(X^*) \leq E[T]$.

Computing the exact conditional expectancies is often too difficult. When an easier-to-compute upper bound is used, this is called the ***method of pessimistic estimators***.

k-WISE INDEPENDENT DISTRIBUTIONS

Truly random bits should be mutually independent. Often it suffices for an algorithm to have its n bits k-wise independent. This means that for every k-tuple of bits, each one of the 2^k combinations has probability 2^{-k}. While full independence requires a probability space of 2^n 0/1-vectors, a k-wise independent probability

space of size roughly $n^{k/2}$ can be constructed. If k is constant, a randomized algorithm can then be derandomized by executing it for each of the vectors in this probability space, incurring a polynomial overhead only.

Often the analysis of an algorithm goes through even when the probability of some k-tuple deviates from 2^{-k} by some absolute error $\epsilon > 0$. In that case, the size of the probability space can be decreased even further, and an algorithm can be derandomized with little overhead by simply executing it for every element of this probability space.

34.9 SOURCES AND RELATED MATERIAL

SURVEYS

All results not given an explicit reference above may be traced in these surveys.

[Cla92, Mul]: General surveys of randomized algorithms in computational geometry.

[Sei93]: An introduction to randomized incremental algorithms using backwards analysis.

[GS93]: Surveys computations with arrangements, including randomized algorithms.

[Aga91]: A survey on geometric partitions.

[Mul94]: This monograph is an extensive treatment of randomized algorithms in computational geometry.

[Mat]: An introduction to derandomization for geometric algorithms, with many references.

[Kar91, MR95]: A survey and a book on randomized algorithms and their analysis in computer science, including derandomization techniques.

[AS93]: This monograph is a good reference on probabilistic proof techniques in combinatorics. It also deals with derandomization.

RELATED CHAPTERS

Because randomized algorithms have been used successfully in nearly all areas of computational geometry, they are mentioned throughout Parts C and D of this Handbook. Areas where randomization plays a particularly important role include:

Chapter 19: Convex hull computations
Chapter 21: Arrangements
Chapter 31: Range searching
Chapter 38: Linear programming in low dimensions

REFERENCES

[AES95] P.K. Agarwal, A. Efrat, and M. Sharir. Vertical decomposition of shallow levels in

3-dimensional arrangements and its applications. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 39–50, 1995.

[Aga91] P.K. Agarwal. Geometric partitioning and its applications. In J.E. Goodman, R. Pollack, and W. Steiger, editors, *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, pages 1–37, volume 6 of *DIMACS Series in Discrete Math. and Theor. Comput. Sci.*. Amer. Math. Soc., Providence, 1991.

[AGM+] H. Alt, L. Guibas, K. Mehlhorn, R. Karp, and A. Widgerson. A method of obtaining randomized algorithms with small tail probabilities. *Algorithmica*, to appear.

[AS93] N. Alon and J. Spencer. *The Probabilistic Method*. Wiley, New York, 1993.

[ASS96] P.K. Agarwal, O. Schwarzkopf, and M. Sharir. The overlay of lower envelopes and its applications. *Discrete Comput. Geom.*, 15:1–13, 1996.

[Cla92] K.L. Clarkson. Randomized geometric algorithms. In D.-Z. Du and F.K. Hwang, editors, *Computing in Euclidean Geometry*, volume 1 of *Lecture Notes Series on Computing*, pages 117–162. World Scientific, Singapore, 1992.

[dBDS94] M. de Berg, K. Dobrindt, and O. Schwarzkopf. On lazy randomized incremental construction. In *Proc. 26th Annu. ACM Sympos. Theory Comput.*, pages 105–114, 1994.

[GS93] L. Guibas and M. Sharir. Combinatorics and algorithms of arrangements. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms Combin.*, pages 9–36. Springer-Verlag, Berlin, 1993.

[Kar91] R. Karp. An introduction to randomized algorithms. *Discrete Appl. Math.*, 34:165–201, 1991.

[Mat] J. Matoušek. Derandomization in computational geometry. In J.-R. Sack and J. Urrutia, editors, *Handbook on Computational Geometry*. North Holland, Amsterdam, to appear. Earlier version in *J. Algorithms* 20:545–580, 1995.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[MSS96] J. Matoušek, O. Schwarzkopf, and J. Snoeyink. Non-canonical randomized incremental construction. Manuscript, 1996.

[MSW90] J. Matoušek, R. Seidel, and E. Welzl. How to net a lot with little: small ϵ-nets for disks and halfspaces. In *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, pages 16–22, 1990.

[Mul] K. Mulmuley. Randomized algorithms in computational geometry. In J.-R. Sack and J. Urrutia, editors, *Handbook on Computational Geometry*. North Holland, Amsterdam, to appear.

[Mul94] K. Mulmuley. *Computational Geometry: An Introduction through Randomized Algorithms*. Prentice Hall, Englewood Cliffs, 1994.

[Sei93] R. Seidel. Backwards analysis of randomized geometric algorithms. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms Combin.*, pages 37–68. Springer-Verlag, Berlin, 1993.

[Sha94] M. Sharir. Almost tight upper bounds for lower envelopes in higher dimensions. *Discrete Comput. Geom.*, 12:327–345, 1994.

[SS96] O. Schwarzkopf and M. Sharir. Vertical decomposition of a single cell in a three-dimensional arrangement of surfaces and its applications. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 20–29, 1996.

[Tag95] B. Tagansky. A new technique for analyzing substructures in arrangements. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 200–210, 1995.

35 ROBUST GEOMETRIC COMPUTATION

Chee K. Yap

INTRODUCTION

Nonrobustness refers to qualitative or catastrophic failures in geometric algorithms arising from numerical errors. Section 35.1 provides background on these problems. Although nonrobustness is already an issue in "purely numerical" computation, the problem is compounded in "geometric computation." In Section 35.2 we characterize such computations. Researchers trying to create robust geometric software have tried two approaches: making fixed-precision computation robust (Section 35.3), and making the exact approach viable (Section 35.4). There is another source of nonrobustness, and that is the phenomenon of "degenerate inputs." General methods for treating degenerate inputs are described in Section 35.5.

35.1 NUMERICAL NONROBUSTNESS ISSUES

Numerical nonrobustness in scientific computing is a well-known and widespread phenomenon. The root cause is the use of ***fixed-precision number*** to represent all real numbers, usually fixed by the machine word size (e.g., 32 bits). The unpredictability of floating-point code across architectural platforms in the 1980's was resolved through a general adoption of the IEEE standard 754-1985. But this standard only makes the errors predictable and consistent across platforms; the errors are still present. None of the ad hoc methods for fixing these errors (such as treating numbers smaller than some ϵ as zero) can guarantee their elimination.

If nonrobustness is problematic in purely numerical computation, it is more intractable in "geometric" computation. In Section 35.2, we elucidate the concept of geometric computations. Based on this understanding, we conclude that nonrobustness problems in fixed-precision computation cannot be solved by purely arithmetic solutions (better arithmetic packages, etc.). Rather, a suitable *fixed-precision geometry* is needed to substitute for the original geometry (which is usually Euclidean). We describe such approaches in Section 35.3.

In Section 35.4, we describe the *exact approach* for achieving robust geometric computation. This demands some type of *big number* package as well as further considerations.

In the final Section, 35.5, we address a different but common cause of numerical nonrobustness, namely, *data degeneracy*. Although this problem has some connection to fixed-precision arithmetic, it is an issue even with the exact approach.

GLOSSARY

Fixed-precision computation: A mode of computation in which every number is represented using some fixed number L of bits, usually 32 or 64. For floating

point numbers, L is partitioned into $L = L_M + L_E$ for the mantissa and the exponent respectively. ***Double precision mode*** is a relaxation of fixed precision: the intermediate values are represented in $2L$ bits, but these are finally truncated back to L bits.

Nonrobustness: The property of code failing on certain kinds of inputs. Here we are mainly interested in nonrobustness that has a numerical origin: the code fails on inputs containing certain patterns of numerical values. Degenerate inputs are just extreme cases of these "bad patterns."

Benign vs. catastrophic errors: Fixed-precision numerical errors are fully expected and so are normally considered to be "benign." In purely numerical computations, errors become "catastrophic" when there is a severe loss of precision. In geometric computations, errors are "catastrophic" when the computed results are qualitatively different from the true answer (e.g., the combinatorial structure is wrong) or when they lead to unexpected or *inconsistent* states of the programs.

Big number packages: Software packages for representing arbitrary precision numbers (usually integers or rational numbers), and in which some basic operations on these numbers are performed exactly. For instance, $+, -, \times$ are implemented exactly with *BigIntegers*. With *BigRationals*, division can also be exact. Other operations such as $\sqrt{\ }$ still need approximations or rounding.

35.2 THE NATURE OF GEOMETRIC COMPUTATION

If the root cause of numerical nonrobustness is arithmetic, then it may appear that the problem can be solved with the right kind of arithmetic package. We may roughly divide the approaches into two camps, depending on whether one uses finite precision arithmetic or insists on exactness (or at least the possibility of computing to arbitrary precision). While arithmetic is an important topic in its own right, our focus here will be on geometric rather than purely arithmetic approaches for achieving robustness.

To understand why nonrobustness is especially problematic for geometric computation, we need to understand what makes a computation "geometric." Geometric computation clearly involves numerical computation, but there is something more. We use the aphorism GEOMETRIC = NUMERIC + COMBINATORIAL to capture this. But the mere presence of combinatorial structures in a numerical computation does not make the computation geometric. There must be some nontrivial ***consistency condition*** holding between the numerical data and the combinatorial data. Thus, we would not consider the classical shortest-path problems on graphs to be geometric: the numerical weights assigned to edges of the graphs are not restricted by any consistency condition. Note that common restrictions on the weights (positivity, integrality, etc.) are not consistency restrictions. But the related ***Euclidean shortest-path problem*** (Chapter 24) is geometric. See Table 35.2.1 for further examples from well-known problems.

Alternatively, we can characterize a computation as "geometric" if it involves constructing or searching a geometric structure (which may only be implicit). Informally, the incidence graph of an arrangement of hyperplanes (Chapter 21), with suitable additional labels and constraints, is an example. A ***geometric structure***

TABLE 35.2.1 Examples of geometric and nongeometric problems.

PROBLEM	GEOMETRIC?
Matrix multiplication, determinant	no
Hyperplane arrangements	yes
Shortest paths on graphs	no
Euclidean shortest paths	yes
Point location	yes
Convex hulls, linear programming	yes
Minimum circumscribing circles	yes

can be viewed as comprising four components:

$$D = (G, \lambda, \Phi(\mathbf{z}), I), \tag{35.2.1}$$

where $G = (V, E)$ is a directed graph, λ is a labeling function on the vertices and edges of G, Φ is the consistency predicate, and I the input parameters. Intuitively, G is the combinatorial part, λ the geometric part, and Φ constrains λ based on the structure of G. We view the input as a sequence of variables $\mathbf{z} = (z_1, \ldots, z_n)$ together with the assignment $I : \mathbf{z} \rightarrow \mathbf{c} = (c_1, \ldots, c_n) \in \mathbb{R}^n$. The z_i's are called ***structural variables*** and the c_i's are ***parameters***. If $u \in V \cup E$, then $\lambda(u)$ is geometric in the sense that it represents a semialgebraic set (Chapter 29) parameterized by the structural variables. More precisely, $\lambda(u)$ is a Tarski formula $\xi(\mathbf{x}, \mathbf{z})$ where $\mathbf{x} = (x_1, \ldots, x_n)$, and represents the set $\{\mathbf{a} \in \mathbb{R}^n \mid \xi(\mathbf{a}, \mathbf{c}) \text{ holds}\}$. We are following Tarski here in identifying semialgebraic sets with geometric objects. The consistency relation $\Phi(\mathbf{z})$ is a Tarski formula. In practice $\Phi(\mathbf{z})$ has the form $(\forall x_1, \ldots, x_n)\phi(\lambda(u_1), \ldots, \lambda(u_m))$ where $u_1, \ldots, u_m$ ranges over elements of $V \cup E$ and ϕ can be systematically constructed from the graph G.

As an example of this notation, consider an arrangement S of hyperplanes in $\mathbb{R}^d$. The combinatorial structure $D(S)$ is the incidence graph $G = (V, E)$ between faces (V) of the arrangement. The parameter vector $\mathbf{c}$ consists of the coefficients of the input hyperplanes. Let the structural parameters $\mathbf{z}$ and $I : \mathbf{z} \mapsto \mathbf{c}$ be the corresponding assignment. The geometric data associate to each node v of the graph the Tarski formula $\lambda(v)$ involving $\mathbf{x}, \mathbf{z}$. When $\mathbf{c}$ is substituted for $\mathbf{z}$, then the formula $\lambda(v)$ defines a face $f_{\lambda,\mathbf{c}}(v)$ (or $f(v)$ for short) of the arrangement. We use the convention that an edge $(u, v) \in E$ represents an "incidence" from $f(u)$ to $f(v)$, where the dimension of $f(u)$ is one more than that of $f(v)$. So $f(v)$ is contained in the closure of $f(u)$. Let $\text{aff}(X)$ denote the affine span of X. Then $(u, v) \in E$ implies $\text{aff}(f(v)) \subseteq \text{aff}(f(u))$ and $f(u)$ lies on one of the two open halfspaces in $\text{aff}(f(u))$ defined by $\text{aff}(f(v))$. We let $\lambda(u, v)$ be the Tarski formula involving $\mathbf{x}, \mathbf{z}$ that defines this open halfspace in $\text{aff}(f(u))$ that contains $f(u)$. As usual, let $f(u, v) = f_{\lambda,\mathbf{c}}(u, v)$ denote this open halfspace. The consistency requirement is that (a) the set $\{f(v) : v \in V\}$ is a partition of $\mathbb{R}^d$, and (b) for each $u \in V$, the set $f(u)$ is nonempty with an irredundant representation of the form

$$f(u) = \text{aff} f(u) \cap \bigcap \{f(u, v) \mid (u, v) \in E\}.$$

Although the above definition is complicated, all of its elements are necessary in order to capture the following additional concepts. We can suppress the parameters

c_i in I and replace I by the list of structural variables $\mathbf{z}$: we call the modified quadruple

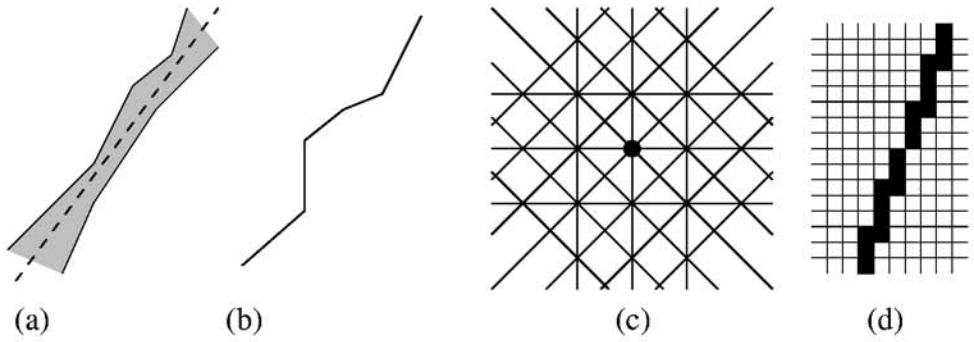
$$\widehat{D} = (G, \lambda, \Phi(\mathbf{z}), \mathbf{z})$$

an ***abstract geometric structure*** and call $D = (G, \lambda, \Phi(\mathbf{z}), I)$ an ***instance*** of $\widehat{D}$. Two geometric structures are ***structurally similar*** if they are instances of a common abstract geometric structure. The structure D in Equation 35.2.1 is ***consistent*** if the predicate $\Phi(\mathbf{c})$ holds. Finally, an abstract geometric structure $\widehat{D}$ is ***realizable*** if there is some consistent instance of $\widehat{D}$.

35.3 FIXED-PRECISION APPROACHES

This section surveys the various approaches within the fixed-precision paradigm. We may classify the approaches into several basic groups. We first illustrate our classification by considering the simple question: "What is the concept of a line in fixed-precision geometry?" Four basic answers to this question are illustrated in Figure 35.3.1 and in Table 35.3.1.

FIGURE 35.3.1
Finite-precision lines.



WHAT IS A FINITE-PRECISION LINE?

We call the first approach ***interval geometry*** because it is the geometric analogue of interval arithmetic. Segal and Sequin [SS85] (among others) define a zone surrounding the line composed of all points within some ϵ distance from the actual line.

The second approach is called ***topologically consistent distortion***. Greene and Yao [GY86] distorted their lines into polylines, where the vertices of these polylines are constrained to be at grid points. Interestingly, although the "fixed-precision representation" is preserved, the number of bits used to represent these polylines can have arbitrary complexity.

The third approach follows a tack of Sugihara [Sug89]. An ideal line is specified by a linear equation, $ax + by + c = 0$. Sugihara interprets a "fixed-precision line"

TABLE 35.3.1 Concepts of a finite-precision line.

	APPROACH	SUBSTITUTE FOR IDEAL LINE	SOURCE
(a)	Interval geometry	a line fattened into a tubular region	[SS85]
(b)	Topological distortion	a polyline	[GY86]
(c)	Rounded geometry	a line whose equation has bounded coefficients	[Sug89]
(d)	Discretization	a suitable set of pixels	computer graphics

to mean that the coefficients in this equation are integer and bounded: $|a|, |b| < K, |c| < K^2$ for some constant K. Call such lines *representable* (see Figure 35.3.1(c) for the case $K = 2$). There are $O(K^4)$ representable lines. An arbitrary line must be "rounded" to the closest (or some nearby) representable line in our algorithms. Hence we call this ***rounded geometry***.

The last approach is based on *discretization*: in traditional computer graphics and in the pattern recognition community, a "line" is just a suitable collection of pixels. This is natural in areas where pixel images are the central objects of study, but less applicable in computational geometry, where relatively compact line representations are desired. This approach will not be considered further in this chapter.

INTERVAL GEOMETRY

In interval geometry, we thicken a geometric object into a zone containing the object. Thus a point may become a disk, and a line becomes a strip between two parallel lines: this is the simplest case and is treated by Segal and Sequin [SS85, Seg90]. They called these "toleranced objects," and in order to obtain correct predicates, they enforce *minimum feature separations*. To do this, features that are too close must be merged (or pushed apart).

Guibas, Salesin, and Stolfi [GSS89] treat essentially the same class of thick objects as Segal and Sequin, although their analysis is mostly confined to geometric data based on points. Instead of insisting on minimum feature separations, their predicates are allowed to return the DON'T KNOW truth value. Geometric predicates (called ϵ-predicates) for objects are systematically treated in this paper.

In general we can consider zones with nonconstant descriptive complexity, e.g., a planar zone with polygonal boundaries. As with interval arithmetic, a zone is generally a conservative estimate because the precise region of uncertainty may be too complicated to compute or to maintain. In applications where zones expand rapidly, there is danger of the zone becoming catastrophically large: Segal [Seg90] reports that a sequence of duplicate-rotate-union operations repeated eleven times to a cube eventually collapsed it to a single vertex.

TOPOLOGICALLY-CONSISTENT DISTORTION

The polylines that Green and Yao use as substitutes for ideal lines (or segments) are determined by the presence of other lines: they construct a line arrangement that preserves certain properties (e.g., two lines intersect in a *connected* subset).

They introduce the concept of a "hook": every intersection point p of two lines introduces a hook (p, q) where q is a closest grid point to p. The idea is to pull p to q. But this may generate other kinds of intersections (derived hooks) and the cascaded effects need to be carefully controlled; this is achieved in their paper. In particular, their polylines lie within a natural "zone" of the ideal line and, in some sense, every point is perturbed a distance $\leq \sqrt{2}$. Further developments include the numerically stable algorithms in [FM91]. The interesting twist here is the use of pseudolines rather than polylines.

Hoffmann, Hopcroft, and Karasick [HHK88] address the problem of intersecting polygons. Phrased in terms of our notion of "geometric structure" (Section 35.2) their goal is to compute a combinatorial structure G that is *consistent* in the sense that G is the structure underlying a consistent geometric structure $D = (G, \lambda, \Phi, \mathbf{c}')$. Here, $\mathbf{c}'$ need not equal the actual input parameter vector $\mathbf{c}$. They show that the intersection of two polygons R_1, R_2 can be efficiently computed, i.e., a consistent G representing $R_1 \cap R_2$ can be computed. However, in their framework, $R_1 \cap (R_2 \cap R_3) \neq (R_1 \cap R_2) \cap R_3$. Hence they need to consider the triple intersection $R_1 \cap R_2 \cap R_3$. Unfortunately, this operation seems to require a nontrivial amount of geometric theorem proving ability.

This suggests that the problem of verifying consistency of combinatorial structures (the "reasoning paradigm" [HHK88]) is generally hard. Indeed, the NP-hard existential theory of reals can be reduced to such problems. But it should be noted that one can avoid this complexity trap by using weaker notions of consistency, for example, by replacing (ideal) lines with pseudolines, as in [FM91].

STABILITY

This is a metric form of topological distortion where we place a priori bounds on the amount of distortion. It is analogous to backwards error analysis in numerical analysis. Framed as the problem of computing the graph G underlying some geometric structure D (as above, for [HHK88]), we could say an algorithm is ***ϵ-stable*** there is a consistent geometric structure $D = (G, \lambda, \Phi, \mathbf{c}')$ such that $\|\mathbf{c} - \mathbf{c}'\| < \epsilon$ where $\mathbf{c}$ is the input parameter vector. We say an algorithm has ***strong*** (resp. ***linear***) stability if ϵ is a constant (resp., $O(n)$) where n is the input size. Fortune and Milenkovic [FM91] provide both linearly stable and strongly stable algorithms for line arrangements. Stable algorithms have been achieved for two other problems on planar point sets: maintaining a triangulation of a point set [For89], and Delaunay triangulations [For92, For95]. The latter problem can be solved stably using either an incremental or a diagonal-flipping algorithm that is $O(n^2)$ in the worst case. Sugihara and Iri [SI92] described a robust algorithm to compute the Voronoi diagram of a large set of points, but it has not been proved stable. Jaromczk and Wasilkowski [JW94] presented stable algorithms for convex hulls.

GEOMETRIC ROUNDING

Sugihara [Sug89] shows that the problem of rounding to representable (but ideal) geometric objects can generally be reduced to the classical problem of *simultaneous approximation by rationals*: given real numbers $a_1, \ldots, a_n$, find integers $p_1, \ldots, p_n$ and q such that $\max_{1 \leq i \leq n} |a_i q - p_i|$ is minimized. There are no efficient algorithms to

solve this exactly, although lattice reduction techniques yield good approximations. The approach of Greene and Yao can be viewed as a geometric rounding problem, where we allow each point of the ideal line to be individually rounded to the closest grid point (but without introducing self-intersections). Milenkovic and Nackman [MN90] show that rounding a collection of disjoint simple polygons while preserving their combinatorial structure is NP-complete.

ARITHMETICAL APPROACHES

Certain approaches might be described as mainly based on arithmetic considerations (as opposed to geometric considerations). Ottmann, Thiemt, and Ullrich [OTU87] show that the use of an accurate scalar product operator leads to improved robustness in segment intersection algorithms; that is, the onset of qualitative errors is delayed. A case study of Dobkin and Silver [DS88] shows that permutation of operations combined with random rounding (up or down) can give accurate predictions of the total round-off error. By coupling this with a multiprecision arithmetic package that is invoked when the loss in significance is too severe, they are able to improve the robustness of their code. This is one of the few studies on the loss of significance in iterated operations.

35.4 EXACT APPROACH

As the name suggests, this approach proposes to compute every geometric object "exactly." This entails the following:

Input is exact. This may be an issue if the input is inherently approximate. Sometimes we can simply treat the approximate inputs as exact, as in the case of an input set of points without any constraints. Otherwise, there are two options: (1) "clean up" the inexact input, transforming it to data exact in some sense, before feeding to the exact algorithm; or (2) formulate a related problem in which the inexact input can be treated as exact (e.g., inexact input points can be viewed as centers of small balls). The cleaning up process in (1) may be nontrivial; it is outside our present scope.

Numerical quantities may be implicitly represented. This is clearly necessary if we hope to compute exactly with values such as $\sqrt{2}$. Here, "explicit" representations of numbers refer to the usual positional notations (binary or decimal notations, or their use in floating-point or rational numbers).

All branching decisions in a computation are errorless. This is an important point: all "critical" phenomena in geometric computation are determined by the particular branches taken in a *computation tree*. Hence if we make only errorless branches, the combinatorial part of a geometric structure D (see Section 35.2) will be correctly computed. Thus we only need to evaluate *test values* to one bit of relative precision, i.e., enough to determine the sign correctly.

For problems (such as convex hulls) requiring only rational numbers, exact computation is possible. In other applications rational arithmetic is not enough. The

most general setting in which exact computation is known to be possible is the framework of *algebraic problems* [Yap97a].

GLOSSARY

Computation tree: A geometric algorithm in the algebraic framework can be viewed as an infinite sequence $T_1, T_2, T_3, \ldots$ of computation trees. Each T_n is restricted to inputs of size n, and is a finite tree with two kinds of nodes: (a) nonbranching nodes, (b) branching nodes. Assume the input to T_n is a sequence of n real parameters $x_1, \ldots, x_n$. A nonbranching node at depth i computes a value v_i, say $v_i \leftarrow f_i(v_1, \ldots, v_{i-1}, x_1, \ldots, x_n)$. A branching node tests a previous computed value v_i and makes a 3-way branch depending on the sign of v_i. Call any v_i that is used solely in a branching node a ***test value***. The branch corresponding to a zero test value is the ***degenerate branch***.

Constructible expressions: Expressions involving $+, -, \times, \div, \sqrt{\ }$.

Rational transformations: Transformations whose defining matrices have only rational numbers.

NAIVE APPROACH

For lack of a better term, we call the approach to exact computation in which every numerical quantity is computed exactly (explicitly if possible) the *naive approach*. Thus an exact algorithm that relies solely on the use of a big number package is probably naive. In general, this approach faces the "bugbear of exact computation," namely, excessive numerical precision. For the class of "bounded-depth problems," this bugbear seems manageable [Yap97a]. Still, using an off-the-shelf big number package appears to be unnecessarily inefficient [FvW93, KLN91]. Recent work (cited in [YD95]) suggests that just improving current big number packages alone is unlikely to gain a factor of more than 10.

BIG EXPRESSION PACKAGES

Expression packages represent the next step after number packages. This stems from the observation that the numerical part in geometric computations can usually be reduced to repeated evaluations of a few expressions (each time with different values of the variables). Examples of expressions include the squared distance $\sum_{i=1}^{n}(p_i - q_i)^2$ between two points p, q, and determinants. If the values computed by these expressions are test values, then it is sufficient to compute them to one bit of relative precision. A technique of Clarkson [Cla92] can be used in case the test values are determinants. Some current implementation efforts are shown in Table 35.4.1, which we briefly highlight.

One of `LN`'s goals is to remove all overhead associated with function calls or dynamic allocation of space for numbers with unknown sizes. It incorporates an effective floating-point filter based on static error analysis. The `LEA` system philosophy is to delay evaluating an expression until forced to, and to maintain intervals of uncertainty for values. Upon complete evaluation, the expression is discarded. The exact computation work in `LEDA` is part of a much more ambitious system

TABLE 35.4.1 Expression packages.

SYSTEM	DESCRIPTION	REFERENCES
LN	Little Numbers	[FvW96]
LEA	Lazy ExAct Numbers	[BJMM93]
LEDA	Library of Efficient Data structures and Algorithms	[BKM+95]
Real/Expr	precision-driven exact expressions	[YD95]

of data structures for combinatorial and geometric computing (see Section 52.1). It uses root bounds to achieve exactness and floating point filters for speed. The Real/Expr design attaches user-specified precisions to each expression, and drives this precision down to the leaves. Expression sharing and time stamps are supported. Both LEDA and Real/Expr implement exact comparisons for constructible expressions by maintaining root bounds.

Shewchuk [She96a] implements a floating-point arithmetic package that uses adaptive-precision techniques. Although this falls short of an expression package, he shows how the underlying techniques can give adaptive evaluation of geometric predicates. For some common geometric predicates, his implementations have impressive timing.

APPLICATIONS

Burnikel, Mehlhorn, and Schirra use exact techniques to compute Voronoi diagrams in [BMS94a]. They show by direct arguments that the precision needed for exact computation in this case is much better than that promised by general root bounds. Chang and Milenkovic [CM93] report their experience with using LN in a solid modeler. Benouamer, Michelucci, and Péroche [BMP94] describe applications to solid modeling (Chapter 47). Dubé and Yap [DY93] describe the implementation of a bigfloat package with dynamic error bounds, and show its usefulness in exact computations (specifically, in Fortune's sweepline algorithm for Voronoi diagrams). Shewchuk [She96b] demonstrates the use of his adaptive bigfloat system for problems such as Delaunay triangulation. Algorithms and experiments for the exact computation of the smallest enclosing cylinder problem appear in Schömer et al. [SSTY96].

THEORY

We have noted that the framework of algebraic problems has exact solutions. This class encompasses most problems in contemporary computational geometry. Algebraic problems can be solved exactly in singly-exponential space [Yap97a]. This result is based on recent progress on the decision problem for Tarski's language, and on the associated cell decomposition problems (Chapter 29). The derivation of effective root bounds for various geometric computations is important for exact algebraic algorithms [SSTY96, BMS94a, CSY94]. The root bounds derived by Canny

[Can88] are extended in [Yap97b, Lecture XI].

A related endeavor is developing techniques to control the cost of high-precision computations. An example is the Euclidean shortest-paths problem in 3-space (3ESP), known to be NP-hard (Section 24.5). Two approaches have been proposed towards practical solutions for 3ESP:

Approximation algorithms. We would like to find a path with length within $(1+\epsilon)$ times the length of the shortest. Here ϵ is an input parameter. The first solution was given by Papadimitriou [Pap85]. Choi, Sellen, and Yap [CSY94] give an alternative scheme and also analyze this in the bit-complexity model.

Precision-sensitive algorithm. Choi, Sellen, and Yap [CSY95] introduce precision-sensitivity as the bit-complexity analogue of "output-sensitive algorithms" in algebraic models of computation. They present such an algorithm for 3ESP, and give some experimental results.

Another issue in exact computation is the avoidance of transcendental functions or values. Rigid transformations are important in constructive solid modeling, but they are in general nonalgebraic. We can get arbitrarily good approximations by using *rational rigid transformations*. Solutions in 2 and 3 dimensions are given in [CDR92] and [MM93], respectively.

35.5 TREATMENT OF DEGENERACIES

Suppose the input to an algorithm is a set of planar points. Depending on the context, any of the following scenarios might be considered "degenerate": two covertical points, three collinear points, four cocircular points. Intuitively, these are degenerate because arbitrarily small perturbations can result in qualitatively different geometric structures. Clearly this has an impact on robustness of algorithms. Degeneracy is basically a discontinuity [Yap90b, Sei94]. Sedgewick [Sed83] calls degeneracies the "bugbear of geometric algorithms."

GLOSSARY

Inherent and induced degeneracy: This is illustrated by the planar convex hull problem: an input set S with three collinear points p, q, r is inherently degenerate if it lies entirely in one halfplane determined by the line through p, q, r. If p, q, r are collinear but S does not lie on one side of the line through p, q, r, then we may have an induced degeneracy for a divide-and-conquer algorithm. This happens when the algorithm solves a subproblem $S' \subseteq S$ containing p, q, r with all the remaining points on one side. Induced degeneracy is algorithm-dependent. In this chapter, we simply say "degeneracy" for induced degeneracy. More precisely, an input is ***degenerate*** if it leads to a path containing a vanishing test value in the computation tree [Yap90b]. A nondegenerate input is also said to be ***generic***.

Generic algorithm: One that is only guaranteed to be correct on generic inputs.

General algorithm: One that works correctly for all (legal) inputs. Note that

"general" and "generic" are often used synonymously in other literature (e.g., "generic inputs" often means inputs in general position).

THE BASIC ISSUES

1. One basic goal of this field is to provide a *systematic transformation* of a generic algorithm A into a general algorithm A'. Since generic algorithms are widespread in the literature, the availability of general tools for this $A \mapsto A'$ transformation is useful for implementing robust algorithms.

2. Underlying any transformations $A \mapsto A'$ is some kind of perturbation of the inputs. This raises the issue of *controlled perturbations*. For example, if A is an algorithm for intersecting two convex polytopes, then we would like the perturbation to expand the input polytopes so that the incidence of a vertex in the relative interior of a face will be detected by A'.

3. There is a *postprocessing issue*: although A' is "correct" in some technical sense, it need not necessarily produce the same outputs as an ideal algorithm A^*. For example, suppose A computes the Voronoi diagram of a set of points in the plane. Four cocircular points are a degeneracy and are not treated by A. The transformed A' can handle four cocircular points but it may output two Voronoi vertices that have identical coordinates and are connected by a Voronoi edge of length 0. This may arise if we use infinitesimal perturbations. The postprocessing problem amounts to cleaning up the output of A' (removing the length-0 edges in this example) so that it conforms to the ideal output of A^*.

CONVERTING GENERIC TO GENERAL ALGORITHMS

There are essentially two general methods for converting a generic algorithm to a general one:

Blackbox sign evaluation schemes. We postulate a ***sign blackbox*** that takes as input a function $f(\mathbf{x}) = f(x_1, \ldots, x_n)$ and parameters $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{R}^n$, and outputs a nonzero sign (either $+$ or $-$). In case $f(\mathbf{a}) \neq 0$, this sign is guaranteed to be the sign of $f(\mathbf{a})$, but the interesting fact is that we get a nonzero sign even if $f(\mathbf{a}) = 0$. We can formulate a consistency property for the blackbox, both in an algebraic setting [Yap90b] or in a geometric setting [Yap90a]. The transformation $A \mapsto A'$ amounts to replacing all evaluations of test values by calls to this blackbox. In the paper [Yap90b] formulating this approach, a family of ***admissible schemes*** for blackboxes is given in case the functions $f(\mathbf{x})$ are polynomials.

Perturbation towards a nondegenerate instance. A fundamentally different approach is provided by Seidel [Sei94], based on the following idea. For any problem, if we know one nondegenerate input $\mathbf{a}^*$ for the problem, then every other input $\mathbf{a}$ can be made nondegenerate by perturbing it in the direction of $\mathbf{a}^*$. We can take the perturbed input to be $\mathbf{a} + \epsilon\mathbf{a}^*$ for some infinitesimal

ϵ. For example, for the convex hull of points in $\mathbb{R}^n$, we can choose $\mathbf{a}^*$ to be distinct points on the moment curve $(t, t^2, \ldots, t^n)$.

There are pros and cons to these two approaches. We currently only have blackbox schemes for polynomials (and rational functions), while Seidel's method would apply even in nonalgebraic settings. On the other hand, the blackbox schemes are independent of particular problems, while the nondegenerate instances $\mathbf{a}^*$ depend on the problem (and on the input size).

The first work in this area is the SoS ("simulation of simplicity") technique of Edelsbrunner and Mücke [EM90]. The method amounts to adding powers of an indeterminate ϵ to each input parameter. Such ϵ-methods were first used in linear programming in the 1950's. The SoS scheme (which was described for perturbing determinants) is an admissible scheme [Yap90b].

Intuitively, sign blackbox invocations should be almost as fast as the actual evaluations with high probability [Yap90b]. But the worst-case behavior is exponential and this prompted Emiris and Canny to propose more efficient numerical approaches [EC95]. To each input parameter a_i in $\mathbf{a}$, they add a perturbation $b_i\epsilon$ (where $b_i \in \mathbb{Z}$ and ϵ is again an infinitesimal): these are called ***linear perturbations***. In case the test values are determinants, they show that a simple choice of the b_i's will ensure nondegeneracy and efficient computation. For general rational function tests, they apply a lemma of Schwartz to show that a random choice of the b_i's will likely lead to nondegeneracy. Emiris, Canny, and Seidel [ECS, Sei94] give a general theorem for the validity of linear perturbations, and apply it to some common test polynomials.

APPLICATIONS AND PRACTICE

Michelluci [Mic95] describes implementations of blackbox schemes, based on the concept of "ϵ-arithmetic." One advantage of his approach is the possibility of controlling the perturbations. Experiences with the use of perturbation in the beneath-beyond convex hull algorithm in arbitrary dimensions are reported in [ECS].

In solid modeling systems, it is very useful to systematically avoid degenerate cases (numerous in this setting). Fortune [For97] uses symbolic perturbation to allow an "exact manifold representation" of nonregularized polyhedral solids (see Section 47.1). The idea is that a dangling rectangular face (for instance) can be perturbed to look like a very flat rectangular solid, which has a manifold representation. Here, controlling the perturbation is clearly necessary.

Hertling and Weihrauch [HW94] define "levels of degeneracy" and use this to obtain lower bounds on the size of decision computation trees.

Burnikel, Mehlhorn, and Schirra [BMS94b] report on the implementation of a line segment intersection algorithm and the convex hull maintenance (no deletion) algorithm in arbitrary dimensions. From this experience, they question the usefulness of perturbation methods, based on three observations: (i) perturbations may increase the running time of an algorithm by an arbitrary amount; (ii) the postprocessing problem can be significant; and (iii) it is not hard to handle degeneracies directly. This may appear to contradict other remarks in the literature, but on balance, this discrepancy can easily be reconciled or explained. This work serves as a reminder that the development of general techniques for perturbation does not mean that direct treatment of degeneracies is no longer needed for particular

applications, and that users must understand the consequences of applying these techniques. Cf. [Sch94].

35.6 OPEN PROBLEMS

1. The ϵ-approximate algorithms for shortest paths in [Pap85, Cla87] are intuitively reasonable, and yet they fit into neither the bit complexity nor the algebraic complexity model of computation. It is a challenge to give a proper formal model for such results (the other alternative would be to convert them into results in the bit model, but this is probably nontrivial [CSY94]).

2. Can the NP-hardness result of Milenkovic and Nackman [MN90] be strengthened to show that rounding a single simple polygon is NP-hard?

3. Dobkin and Silver [DS88] is an empirical study of the complexity of iterated geometric operations; it would be useful to have also formal complexity results. In particular, does applying their *in* operation n times yield a figure that requires $\Omega(1)^n$ bits of precision? It is known that $O(1)^n$ bits suffice.

4. It is known from an example of Paterson [Yap90a] that there are consistent blackboxes that do not arise from admissible orderings. Thus the question of constructing a family of complete blackbox schemes is open.

35.7 SOURCES AND RELATED MATERIAL

SURVEYS

Forrest [For87] is an influential overview of the field of computational geometry. He deplores the gap between theory and practice and describes the open problem of robust intersection of line segments (expressing a belief that robust solutions do not exist). Fortune treats implementation issues in geometric algorithms in [For93]. A survey of exact geometric computation is found in Yap and Dubé [YD95]. Robust geometric modelers are surveyed in [PCH+95].

RELATED CHAPTERS

Chapter 21: Arrangements
Chapter 24: Shortest paths and networks
Chapter 29: Computational real algebraic geometry
Chapter 47: Solid modeling
Chapter 52: Computational geometry software

REFERENCES

[BJMM93] M.O. Benouamer, P. Jaillon, D. Michelucci, and J.-M. Moreau. A lazy arithmetic library. In *Proc. 11th Annu. IEEE Sympos. Comput. Arith.*, pages 242–269, 1993.

[BKM+95] C. Burnikel, J. Könnemann, K. Mehlhorn, S. Näher, S. Schirra, and C. Uhrig. Exact geometric computation in LEDA. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages C18–C19, 1995.

[BMP94] M.O. Benouamer, D. Michelucci, and B. Péroche. Error-free boundary evaluation based on a lazy rational arithmetic: a detailed implementation. *Comput. Aided Design*, 26:403–416, 1994.

[BMS94a] C. Burnikel, K. Mehlhorn, and S. Schirra. How to compute the Voronoi diagram of line segments: Theoretical and experimental results. In *Proc. 2nd Annu. European Sympos. Algorithms*, volume 855 of *Lecture Notes in Comput. Sci.*, pages 227–239. Springer-Verlag, Berlin, 1994.

[BMS94b] C. Burnikel, K. Mehlhorn, and S. Schirra. On degeneracy in geometric computations. In *Proc. 5th ACM-SIAM Sympos. Discrete Algorithms*, pages 16–23, 1994.

[Can88] J.F. Canny. *The Complexity of Robot Motion Planning*. Ph.D. Thesis, MIT, Cambridge, 1988.

[CDR92] J.F. Canny, B. Donald, and E.K. Ressler. A rational rotation method for robust geometric algorithms. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 251–160, 1992.

[Cla87] K.L. Clarkson. Approximation algorithms for shortest path motion planning. In *Proc. 19th Annu. ACM Sympos. Theory Comput.*, pages 56–65, 1987.

[Cla92] K.L. Clarkson. Safe and effective determinant evaluation. In *Proc. 33rd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 387–395, 1992.

[CM93] J.D. Chang and V. Milenkovic. An experiment using LN for exact geometric computations. In *Proc. 5th Canad. Conf. Comput. Geom.*, Waterloo, pages 67–72, 1993.

[CSY94] J. Choi, J. Sellen, and C.K. Yap. Approximate Euclidean shortest path in 3-space. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 41–48, 1994.

[CSY95] J. Choi, J. Sellen, and C.K. Yap. Precision-sensitive Euclidean shortest path in 3-space. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 350–359, 1995.

[DS88] D. Dobkin and D. Silver. Recipes for geometry and numerical analysis—Part I: An empirical study. In *Proc. 4th Annu. ACM Sympos. Comput. Geom.*, pages 93–105, 1988.

[DY93] T. Dubé and C.K. Yap. A basis for implementing exact geometric algorithms (extended abstract), 1993. URL `http://cs.nyu.edu/cs/faculty/yap`.

[EC95] I.Z. Emiris and J.F. Canny. A general approach to removing degeneracies. *SIAM J. Comput.*, 24:650–664, 1995.

[ECS] I.Z. Emiris, J.F. Canny, and R. Seidel. Efficient perturbations for handling geometric degeneracies. *Algorithmica*, to appear.

[EM90] H. Edelsbrunner and E.P. Mücke. Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graph.*, 9:66–104, 1990.

[FM91] S.J. Fortune and V.J. Milenkovic. Numerical stability of algorithms for line arrangements. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 334–341, 1991.

[For87] A.R. Forrest. Computational geometry and software engineering: Towards a geometric computing environment. In D.F. Rogers and R.A. Earnshaw, editors, *Techniques for Computer Graphics*, pages 23–37. Springer-Verlag, Berlin, 1987.

[For89] S.J. Fortune. Stable maintenance of point-set triangulations in two dimensions. In *Proc. 30th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 494–499, 1989.

[For92] S.J. Fortune. Numerical stability of algorithms for 2-d Delaunay triangulations. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 83–92, 1992.

[For93] S. Fortune. Progress in computational geometry. In R. Martin, editor, *Directions in Geometric Computing*, pages 81–128. Information Geometers, Winchester, 1993.

[For95] S.J. Fortune. Numerical stability of algorithms for 2-d Delaunay triangulations. *Internat. J. Comput. Geom. Appl.*, 5:193–213, 1995.

[For97] S.J. Fortune. Polyhedral modeling with multiprecision integer arithmetic. *Comput. Aided Design*, 29:123–133, 1997.

[FvW93] S.J. Fortune and C.J. van Wyk. Efficient exact arithmetic for computational geometry. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 163–172, 1993.

[FvW96] S.J. Fortune and C.J. van Wyk. Static analysis yields efficient exact integer arithmetic for computational geometry. *ACM Trans. Graph.*, 15:223–248, 1996.

[GSS89] L. Guibas, D. Salesin, and J. Stolfi. Epsilon geometry: building robust algorithms from imprecise computations. In *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, pages 208–217, 1989.

[GY86] D.H. Greene and F.F. Yao. Finite-resolution computational geometry. In *Proc. 27th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 143–152, 1986.

[HHK88] C. Hoffmann, J. Hopcroft, and M. Karasick. Towards implementing robust geometric computations. In *Proc. 4th Annu. ACM Sympos. Comput. Geom.*, pages 106–117, 1988.

[HW94] P. Hertling and K. Weihrauch. Levels of degeneracy and exact lower complexity bounds for geometric algorithms. In *Proc. 6th Canad. Conf. Comput. Geom.*, pages 237–242, 1994.

[JW94] J.W. Jaromczyk and G.W. Wasilkowski. Computing convex hull in a floating point arithmetic. *Comput. Geom. Theory Appl.*, 4:283–292, 1994.

[KLN91] M. Karasick, D. Lieber, and L.R. Nackman. Efficient Delaunay triangulation using rational arithmetic. *ACM Trans. Graphics*, 10:71–91, 1991.

[Mic95] D. Michelucci. An epsilon-arithmetic for removing degeneracies. In *Proc. 12th Annu. IEEE Sympos. Comput. Arith.*, pages 230–237, 1995.

[MM93] V. Milenkovic and V. Milenkovic. Rational orthogonal approximations to orthogonal matrices. In *Proc. 5th Canad. Conf. Comput. Geom.*, Waterloo, pages 485–490, 1993.

[MN90] V. Milenkovic and L. Nackman. Finding compact coordinate representations for polygons and polyhedra. In *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, pages 244–252, 1990.

[OTU87] T. Ottmann, G. Thiemt, and C. Ullrich. Numerical stability of geometric algorithms. In *Proc. 3rd Annu. ACM Sympos. Comput. Geom.*, pages 119–125, 1987.

[Pap85] C.H. Papadimitriou. An algorithm for shortest-path motion in three dimensions. *Inform. Process. Lett.*, 20:259–263, 1985.

[PCH+95] N.M. Patrikalakis, W. Cho, C.-Y. Hu, T. Maekawa, E.C. Sherbrooke, and J. Zhou. Towards robust geometric modellers, 1994 progress report. In *Proc. 1995 NSF Design and Manufac. Grantees Conf.*, pages 139–140, 1995.

[Sch94] P. Schorn. Degeneracy in geometric computation and the perturbation approach. *Comput. J.*, 37:35–42, 1994.

[Sed83] R. Sedgewick. *Algorithms.* Addison-Wesley, Reading, 1983.

[Seg90] M.G. Segal. Using tolerances to guarantee valid polyhedral modeling results. *Comput. Graph.* (Proc. SIGGRAPH '90), 24:105–114, 1990.

[Sei94] R. Seidel. The nature and meaning of perturbations in geometric computing. In *Proc. 11th Sympos. Theoret. Aspects Comput. Sci.*, volume 775 of *Lecture Notes in Comput. Sci.*, pages 3–17, Springer-Verlag, Berlin, 1994.

[She96a] J.R. Shewchuk. Robust adaptive floating-point geometric predicates. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 141–150, 1996.

[She96b] J.R. Shewchuk. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In *Proc. 1st ACM Workshop Appl. Comput. Geom.*, volume 1148 of *Lecture Notes in Comput. Sci.*, pages 124–133, Springer-Verlag, Berlin, 1996.

[SI92] K. Sugihara and M. Iri. Construction of the Voronoi diagram for 'one million' generators in single-precision arithmetic. *Proc. IEEE*, 80:1471–1484, 1992.

[SS85] M.G. Segal and C.H. Sequin. Consistent calculations for solids modelling. In *Proc. 1st Annu. ACM Sympos. Comput. Geom.*, pages 29–38, 1985.

[SSTY96] E. Schömer, J. Sellen, M. Teichmann, and C.K. Yap. Smallest enclosing cylinders. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages C13–14, 1996.

[Sug89] K. Sugihara. On finite-precision representations of geometric objects. *J. Comput. Syst. Sci.*, 39:236–246, 1989.

[Yap90a] C.K. Yap. A geometric consistency theorem for a symbolic perturbation scheme. *J. Comput. Syst. Sci.*, 40:2–18, 1990.

[Yap90b] C.K. Yap. Symbolic treatment of geometric degeneracies. *J. Symbolic Comput.*, 10:349–370, 1990.

[Yap97a] C.K. Yap. Towards exact geometric computation. *Internat. J. Comput. Geom. Appl.*, 7:3–23, 1997.

[Yap97b] C.K. Yap. *Fundamental Problems in Algorithmic Algebra.* Princeton University Press, 1997. URL `ftp://cs.nyu.edu/pub/local/yap/algebra-bk`.

[YD95] C.K. Yap and T. Dubé. The exact computation paradigm. In D.-Z. Du and F.K. Hwang, editors, *Computing in Euclidean Geometry*, 2nd edition, pages 452–486. World Scientific, Singapore, 1995.

36 PARALLEL ALGORITHMS IN GEOMETRY

Michael T. Goodrich

INTRODUCTION

The goal of parallel algorithm design is to develop parallel computational methods that run very fast with as few processors as possible, and there is an extensive literature of such algorithms for computational geometry problems. There are several different parallel computing models, and in order to maintain a focus in this chapter, we will describe results in the Parallel Random Access Machine (PRAM) model, which is a synchronous parallel machine model in which processors share a common memory address space (and all inter-processor communication takes place through this shared memory). Although it does not capture all aspects of parallel computing, it does model the essential properties of parallelism. Moreover, it is a widely-accepted model of parallel computation, and all other reasonable models of parallel computation can easily simulate a PRAM.

The PRAM model is subdivided into submodels based on how one wishes to handle concurrent memory access to the same location. The Exclusive-Read, Exclusive-Write (EREW) variant does not allow for concurrent access. The Concurrent-Read, Exclusive-Write (CREW) variant permits concurrent memory reads, but memory writes must be exclusive. Finally, the Concurrent-Read, Concurrent-Write (CRCW) variant allows for both concurrent memory reading and writing, with concurrent writes being resolved by some simple rule, such as having an arbitrary member of a collection of conflicting writes succeed. One can also define randomized versions of each of these models ("rand-EREW," etc.), where in addition to the usual arithmetic and comparison operations, each processor can generate a random number from 1 to n in one step.

In Section 36.1 we give a brief discussion of general techniques for parallel geometric algorithm design. We then partition the research in parallel computational geometry into problems dealing with convexity (Section 36.2), arrangements and decompositions (Section 36.3), proximity (Section 36.4), geometric searching (Section 36.5), and visibility, envelopes, and geometric optimization (Section 36.6).

36.1 SOME PARALLEL TECHNIQUES

The design of efficient parallel algorithms for computational geometry problems often depends upon the use of powerful general parallel techniques (e.g., see [AL93, J92, KR90, Rei93]). We review some of these techniques below.

PARALLEL DIVIDE-AND-CONQUER

Possibly the most general technique is parallel divide-and-conquer. In applying this technique one divides a problem into two or more subproblems, solves the

subproblems recursively in parallel, and then merges the subproblem solutions to solve the entire problem. As an example application of this technique, consider the problem of constructing the upper convex hull of a S set of n points in the plane presorted by x-coordinates. Divide the list S into $\lceil\sqrt{n}\rceil$ contiguous sublists of size $\lfloor\sqrt{n}\rfloor$ each and recursively construct the upper convex hull of the points in each list. Assign a processor to each pair of sublists and compute the common upper tangent line for the two upper convex hulls for these two lists, which can be done in $O(\log n)$ time using a well-known binary-search computation. By maximum computations on the left and right common tangents, respectively, for each subproblem S_i, one can determine which vertices on the upper convex hull of S_i belong to the upper convex hull of S. Compressing all the vertices identified to be on the upper convex hull of S constructs an array representation of this hull, completing the construction.

The running time of this method is characterized by the recurrence relation $T(n) \leq T(\sqrt{n}) + O(\log n)$, which implies that $T(n)$ is $O(\log n)$. It is important to note that the coefficient for the $T(\sqrt{n})$ term is 1 even though we had $\lceil\sqrt{n}\rceil$ subproblems, for all these subproblems were processed simultaneously in parallel. The number of processors needed for this computation can be characterized by the recurrence relation $P(n) \leq \max\{\lceil\sqrt{n}\rceil P(\sqrt{n}), n\}$, which implies that $P(n)$ is $O(n)$. Thus, the *work* needed for this computation is $O(n \log n)$, which is not quite optimal. Still, this method can be adapted to result in an optimal work bound (e.g., [Che95]).

BUILD-AND-SEARCH

Another important technique in parallel computational geometry is the build-and-search technique. It is a paradigm that often yields efficient parallel adaptations of sequential algorithms designed using the powerful plane sweeping technique. In the build-and-search technique, the solution to a problem is partitioned into a *build* phase, where one constructs in parallel a data structure built from the geometric data present in the problem, and a *search* phase, where one searches this data structure in parallel to solve the problem at hand. An example of an application of this technique is for the trapezoidal decomposition problem: given a collection of nonintersecting line segments in the plane, determine the first segments intersected by vertical rays emanating from each segment endpoint (cf. Figure 34.0.1). The existing efficient parallel algorithm for this problem is based upon first building in parallel a data structure on the input set of segments that allows for such vertical ray-shooting queries to be answered in $O(\log n)$ time by a single processor, and then querying this structure for each segment endpoint in parallel. This results in a parallel algorithm with an efficient $O(n \log n)$ work bound and fast $O(\log n)$ query time.

36.2 CONVEXITY

Results on the problem of constructing the convex hull of n points in $\mathbb{R}^d$ are summarized in Table 36.2.1, for various fixed values of d, and, in the case of $d = 2$, under assumptions about whether the input is presorted. We restrict our attention to parallel algorithms with efficient work bounds, where we use the term ***work***

of an algorithm here to refer to the product of its running time and the number of processors used by the algorithm. A parallel algorithm has an ***optimal*** work bound if the work used asymptotically matches the sequential lower bound for the problem. In the table, h denotes the size of the hull, and c is some fixed constant. Also, we use (throughout this chapter) $\bar{O}(f(n))$ to denote an asymptotic bound that holds with high probability.

TABLE 36.2.1 Parallel convex hull algorithms.

PROBLEM	MODEL	TIME	WORK	REF
2-dim presorted	rand-CRCW	$\bar{O}(\log^* n)$	$\bar{O}(n)$	[GG91]
2-dim presorted	CRCW	$O(\log\log n)$	$O(n)$	[BSV96]
2-dim presorted	EREW	$O(\log n)$	$O(n)$	[Che95]
2-dim polygon	EREW	$O(\log n)$	$O(n)$	[Che95]
2-dim	rand-CRCW	$\bar{O}(\log n)$	$\bar{O}(n\log h)$	[GG91]
2-dim	EREW	$O(\log n)$	$O(n\log n)$	[MS88]
2-dim	EREW	$O(\log^2 n)$	$O(n\log h)$	[GG91]
3-dim	rand-CRCW	$\bar{O}(\log n)$	$\bar{O}(n\log n)$	[RS92]
3-dim	CREW	$O(\log n)$	$O(n^{1+1/c})$	[AP93]
3-dim	EREW	$O(\log^2 n)$	$O(n\log n)$	[AGR94]
3-dim	EREW	$O(\log^3 n)$	$O(n\log h)$	[AGR94]
Fixed $d \geq 4$	rand-EREW	$\bar{O}(\log^2 n)$	$\bar{O}(n^{\lfloor d/2\rfloor})$	[AGR94]
Even $d \geq 4$	EREW	$O(\log^2 n)$	$O(n^{\lfloor d/2\rfloor})$	[AGR94]
Odd $d > 4$	EREW	$O(\log^2 n)$	$O(n^{\lfloor d/2\rfloor}\log^c n)$	[AGR94]

We discuss a few of these algorithms to illustrate their flavor.

2-DIMENSIONAL CONVEX HULLS

The two-dimensional convex hull algorithm of Miller and Stout [MS88] is based upon a parallel divide-and-conquer scheme where one presorts the input and then divides it into many subproblems ($O(n^{1/4})$ in their case), solves each subproblem independently in parallel, and then merges all the subproblem solutions together in $O(\log n)$ parallel time. Of course, the difficult step is the merge of all the subproblems, with the principal difficulty being the computation of common tangents between hulls. The total running time is characterized by the recurrence

$$T(n) \leq T(n^{1/4}) + O(\log n),$$

which solves to $T(n) = O(\log n)$.

3-DIMENSIONAL CONVEX HULLS

All of the 3-dimensional convex hull algorithms listed in Table 36.2.1 are also based upon this many-way, divide-and-conquer paradigm, except that there is no notion of presorting in three dimensions, so the subdivision step also becomes nontrivial. Reif and Sen [RS92] use a random sample to perform the division, and the methods

of Amato, Goodrich, and Ramos [AGR94] derandomize this approach. Amato and Preparata [AP93] use parallel separating planes, an approach extended to higher dimension in [AGR94].

LINEAR PROGRAMMING

A problem strongly related to convex hull construction, which has also been addressed in a parallel setting, is d-dimensional linear programming, for fixed dimensions d (see Chapter 38). Of course, one could solve this problem by transforming it to its dual problem, constructing a convex hull in this dual space, and then evaluating each vertex in the simplex that is dual to this convex hull. This would be quite inefficient, however, for $d \geq 4$. The best parallel bounds for this problem are listed in Table 36.2.2. See Section 38.6 for a detailed discussion.

TABLE 36.2.2 d-dimensional parallel linear programming.

MODEL	TIME	WORK	REF
Rand-CRCW	$\bar{O}(1)$	$\bar{O}(n)$	[AM90]
CRCW	$O((\log\log n)^{d+1})$	$O(n)$	[Goo96]
EREW	$O(\log n(\log\log n)^{d-1})$	$O(n)$	[Goo96]

OPEN PROBLEMS

There are a number of interesting open problems regarding convexity:

1. Can d-dimensional linear programming be solved (deterministically) in $O(\log n)$ time using $O(n)$ work in the CREW model?
2. Is there an efficient output-sensitive parallel convex hull algorithm for $d \geq 4$?
3. Is there an optimal-work $O(\log^2 n)$-time CREW convex hull algorithm for odd dimensions greater than 4?

36.3 ARRANGEMENTS AND DECOMPOSITIONS

Another important class of geometric problems that has been addressed in the parallel setting are arrangement and decomposition problems, which deal with ways of partitioning space. We review the best parallel bounds for such problems in Table 36.3.1.

GLOSSARY

Arrangement: The partition of space determined by the intersections of a collection of geometric objects, such as lines, line segments, or (in higher dimensions)

hyperplanes. In this chapter, algorithms for constructing arrangements produce the *incidence graph*, which stores all adjacency information between the various primitive topological entities determined by the partition, such as intersection points, edges, faces, etc. See Section 21.3.1.

Red-blue arrangement: An arrangement defined by two sets of objects A and B such that the objects in A (resp. B) are nonintersecting.

Axis-parallel: All segments/lines are parallel to one of the coordinate axes.

Polygon triangulation: A decomposition of the interior of a polygon into triangles by adding diagonals between vertices. See Section 23.2.

Trapezoidal decomposition: A decomposition of the plane into trapezoids (and possibly triangles) by adding appropriate vertical line segments incident to vertices. See Section 30.3.

Star-shaped polygon: A (simple) polygon that is completely visible from a single point. A polygon with nonempty kernel. See Section 23.1.

$1/r$-cutting: A partition of $\mathbb{R}^d$ into $O(r^d)$ simplicies such that each simplex intersects at most n/r hyperplanes. See Sections 31.2 and 34.1.

TABLE 36.3.1 Parallel arrangement and decomposition algorithms.

PROBLEM	MODEL	TIME	WORK	REF
d-dim hyperplane arr	EREW	$O(\log n)$	$O(n^d)$	[AGR94]
2-dim seg arr	rand-CRCW	$\bar{O}(\log n)$	$\bar{O}(n \log n + k)$	[CCT92]
2-dim axis-par seg arr	CREW	$O(\log n)$	$O(n \log n + k)$	[Goo91]
2-dim red-blue seg arr	CREW	$O(\log n)$	$O(n \log n + k)$	[GSG92]
2-dim seg arr	EREW	$O(\log^2 n)$	$O(n \log n + k)$	[AGR95]
Polygon triangulation	CRCW	$O(\log n)$	$O(n)$	[Goo92]
Polygon triangulation	CREW	$O(\log n)$	$O(n \log n)$	[Goo89]
2-dim nonint seg trap decomp	CREW	$O(\log n)$	$O(n \log n)$	[ACG89]

We sketch the one randomized algorithm in Table 36.3.1 to illustrate how randomization and parallel computation can be mixed. Let S be a set of segments in the plane with k intersecting pairs. The goal is to construct $\mathcal{A}(S)$, the arrangement induced by S. First, an estimate $\hat{k}$ for k is obtained from a random sample. Then a random subset $R \subset S$ of a size r dependent on $\hat{k}$ is selected. $\mathcal{A}(R)$ is constructed using a suboptimal parallel algorithm, and processed (in parallel) for point location. Next the segments intersecting each cell of $\mathcal{A}(R)$ are found using a parallel point-location algorithm, together with some ad hoc techniques. Visibility information among the segments meeting each cell is computed using another suboptimal parallel algorithm. Finally, the resulting cells are merged in parallel. Because various key parameters in the suboptimal algorithms are kept small by the sampling, optimal expected work is achieved.

All of the algorithms for computing segment arrangements are ***output-sensitive***, in that their work bounds depend upon both the input size and the output size. In these cases we must slightly extend our computational model to allow for the machine to request additional processors if necessary. In all these algorithms,

this request may originate only from a single "master" processor, however, so this modification is not that different from our assumption that the number of processors assigned to a problem can be a function of the input size. Of course, to solve a problem on a real parallel computer, one would simulate one of these efficient parallel algorithms to achieve an optimal speed-up over what would be possible using a sequential method.

A related class of intersection-related problems is the class of problems dealing with methods for detecting intersections. Testing if a collection of objects has at least one intersection is frequently easier than finding all such intersections, and Table 36.3.2 reviews such results in the parallel domain.

TABLE 36.3.2 Parallel intersection detection algorithms.

PROBLEM	MODEL	TIME	WORK	REF
2 convex polygons	CREW	$O(1)$	$O(n^{1/c})$	[DK89]
2 star-shaped polygons	CREW	$O(\log n)$	$O(n)$	[GM91]
2 convex polyhedra	CREW	$O(\log n)$	$O(n)$	[DK89]

Given a collection of n hyperplaces in $\mathbb{R}^d$, another important decomposition problem is the construction of a $(1/r)$-cutting. Here an EREW algorithm running in $O(\log n \log r)$ time using $O(nr^{d-1})$ work has been obtained [Goo93].

OPEN PROBLEMS

1. Is there an optimal-work $O(\log n)$-time polygon triangulation algorithm that doesn't use concurrent writes?
2. Can a line segment arrangement be constructed in $O(\log n)$ time using $O(n \log n + k)$ work in the CREW model?

36.4 PROXIMITY

An important property of Euclidean space is that it is a metric space, and distance plays an important role in many computational geometry applications. For example, computing a closest pair of points can be used in collision detection, as can the more general problem of computing the nearest neighbor of each point in a set S, a problem we will call the ***all-nearest neighbors (ANN)*** problem. Perhaps the most fundamental problem in this domain is the subdivision of space into regions where each region $V(s)$ is defined by a *site* s in a set S of geometric objects such that each point in $V(s)$ is closer to s than to any other object in S. This subdivision is the ***Voronoi diagram*** (Chapter 20); its graph-theoretic dual, which is also an important geometric structure, is the ***Delaunay triangulation*** (Section 22.1). For a set of points S in $\mathbb{R}^d$, there is a simple "lifting" transformation that takes each

point $(x_1, x_2, \ldots, x_d) \in S$ to the point $(x_1, x_2, \ldots, x_d, x_1^2 + x_2^2 + \ldots + x_d^2)$, forming a set of points S' in $\mathbb{R}^{d+1}$ (Section 20.1). Each simplex on the convex hull of S' with a negative $(d+1)$-st component in its normal vector projects back to a simplex of the Delaunay triangulation in $\mathbb{R}^d$. Thus, any $(d+1)$-dimensional convex hull algorithm immediately implies a d-dimensional Voronoi diagram (VD) algorithm. Table 36.4.1 summarizes the bounds of efficient parallel algorithms for constructing Voronoi diagrams in this way, as well as methods that are designed particularly for Voronoi diagram construction or other specific proximity problems. (In the table, the underlying objects are points unless stated otherwise.)

GLOSSARY

Convex position: A set of points that are all on the boundary of their convex hull.

Voronoi diagram for line segments: A Voronoi diagram that is defined by a set of nonintersecting line segments, with distance from a point p to a segment s being defined as the distance from p to a closest point on s. See Section 20.3.

TABLE 36.4.1 Parallel proximity algorithms.

PROBLEM	MODEL	TIME	WORK	REF
2-dim ANN in convex pos	EREW	$O(\log n)$	$O(n)$	[CG92]
2-dim ANN	EREW	$O(\log n)$	$O(n \log n)$	[CG92]
d-dim ANN	CREW	$O(\log n)$	$O(n \log n)$	[Cal93]
2-dim VD in L_1 metric	CREW	$O(\log n)$	$O(n \log n)$	[WC90]
2-dim VD	rand-CRCW	$\tilde{O}(\log n)$	$\tilde{O}(n \log n)$	[RS92]
2-dim VD	CRCW	$O(\log n \log\log n)$	$O(n \log n \log\log n)$	[CGÓ90]
2-dim VD	EREW	$O(\log^2 n)$	$O(n \log n)$	[AGR94]
2-dim VD for segments	CREW	$O(\log^2 n)$	$O(n \log^2 n)$	[GÓY93]
3-dim VD	EREW	$O(\log^2 n)$	$O(n^2)$	[AGR94]

OPEN PROBLEMS

1. Can a 2-dimensional Voronoi diagram be constructed in $O(\log n)$ time using $O(n \log n)$ work under either the CREW or EREW models?
2. Can one construct general Voronoi diagrams, including Voronoi diagrams for sets of nonintersecting line segments, in $O(n \log n)$ work and at most $O(\log^2 n)$ time?
3. Is there an efficient output-sensitive parallel algorithm for constructing 3-dimensional Voronoi diagrams?

36.5 GEOMETRIC SEARCHING

Given a subdivision of space by a collection S of geometric objects, such as line segments, the point location problem is to build a data structure for this set that can quickly answer *vertical ray-shooting queries*, where one is given a point p and asked to report the first object in S hit by a vertical ray from p. We summarize efficient parallel algorithms for planar point location in Table 36.5.1. The time and work bounds listed, as well as the computational model, are for building the data structure to achieve an $O(\log n)$ query time. We do not list the space bounds for any of these methods in the table since, in every case, they are equal to the preprocessing work bounds.

GLOSSARY

Arbitrary planar subdivision: A subdivision of the plane (not necessarily connected), defined by a set of line segments that intersect only at their endpoints.

Monotone subdivision: A connected subdivision of the plane in which each face is intersected by a vertical line in a single segment.

Triangulated subdivision: A connected subdivision of the plane into triangles whose corners are vertices of the subdivision (see Chapter 22).

Shortest path in a polygon: The shortest path between two points that does not go outside of the polygon (see Section 23.4).

Ray-shooting query: A query whose answer is the first object hit by a ray oriented in a specified direction from a specified point.

TABLE 36.5.1 Parallel geometric searching algorithms.

QUERY PROBLEM	MODEL	TIME	WORK	REF
Point loc in arb subdivision	CREW	$O(\log n)$	$O(n \log n)$	[ACG89]
Point loc in monotone subdivision	EREW	$O(\log n)$	$O(n)$	[TV91]
Point loc in triangulated subdivision	CREW	$O(\log n)$	$O(n)$	[CZ90]
Point loc in d-dim hyp arr	EREW	$O(\log n)$	$O(n^d)$	[AGR94]
Shortest path in triangulated polygon	CREW	$O(\log n)$	$O(n)$	[GSG92]
Ray shooting in triangulated polygon	CREW	$O(\log n)$	$O(n)$	[HS93]
Line & convex polyhedra intersection	CREW	$O(\log n)$	$O(n)$	[CZ90]

OPEN PROBLEMS

1. Is there an efficient data structure that allows n simultaneous point locations to be performed in $O(\log n)$ time using $O(n)$ processors in the EREW model?

2. Is there an efficient data structure for 3-dimensional point location in convex subdivisions that can be constructed in $O(n \log n)$ work and at most $O(\log^2 n)$ time and which allows for a query time that is at most $O(\log^2 n)$?

36.6 VISIBILITY, ENVELOPES, AND OPTIMIZATION

We summarize efficient parallel methods for various visibility and lower envelope problems for a simple polygon in Table 36.6.1. In the table, m denotes the number of edges in a visibility graph. For definitions see Chapter 25.

TABLE 36.6.1 Parallel visibility algorithms for a simple polygon.

PROBLEM	MODEL	TIME	WORK	REF
Kernel	EREW	$O(\log n)$	$O(n)$	[Che95]
Vis from a point	EREW	$O(\log n)$	$O(n)$	[ACW91]
Vis from an edge	CRCW	$O(\log n)$	$O(n)$	[Her92]
Vis from an edge	CREW	$O(\log n)$	$O(n \log n)$	[GSG92]
Vis graph	CREW	$O(\log n)$	$O(n \log^2 n + m)$	[GSG92]

We sketch the algorithm for computing the point visibility polygon [ACW91], which is notable for two reasons: first, it is employed as a subprogram in many other algorithms; and second, it requires much more intricate processing and analysis than the relatively simple optimal sequential algorithm (Section 25.3). The parallel algorithm is recursive, partioning the boundary into $n^{1/4}$ subchains, and computing *visibility chains* from the source point of visibility x. Each of these chains is star-shaped with respect to x, i.e., effectively "monotone" (see Section 23.1). This monotonicity property is, however, insufficient to intersect the visibility chains quickly enough in the merge step to obtain optimal bounds. Rather, the fact that the chains are subchains of the boundary of a simple polygon must be exploited to achieve logarithmic-time computation of the intersection of two chains. This then leads to the optimal bounds quoted in Table 36.6.1.

The bounds of efficient parallel methods for visibility problems on general sets of segments and curves in the plane are summarized in Table 36.6.2.

GLOSSARY

Lower envelope: The function $F(x)$ defined as the point-wise minimum of a collection of functions $\{f_1, f_2, \ldots, f_n\}$: $F(x) = \min_i f_i(x)$ (see Section 21.2).

k-intersecting curves: A set of curves every two of which intersect at most k times (where they cross).

$\lambda_s(n)$: The maximum length of a Davenport-Schinzel sequence of order s on n symbols. If s is a constant, $\lambda_s(n)$ is $o(n \log^* n)$. See Section 40.4.

TABLE 36.6.2 General parallel visibility and enveloping algorithms.

PROBLEM	MODEL	TIME	WORK	REF
Lower env for segments	EREW	$O(\log^2 n)$	$O(n \log n)$	[Her89]
Lower env for k-int curves	EREW	$O(\log^2 n)$	$O(\lambda_{k+2}(n) \log n)$	[BM87]

Finally, we summarize some efficient parallel algorithms for solving several geometric optimization problems in Table 36.6.3.

GLOSSARY

Largest-area empty rectangle: For a collection S of n points in the plane, the largest-area rectangle that does not contain any point of S in its interior.

All-farthest neighbors problem in a simple polygon: Determine for each vertex p of a simple polygon the vertex q such that the shortest path from p is longest.

Closest visible-pair between polygons: A closest pair of mutually-visible vertices between two nonintersecting simple polygons in the plane.

Minimum circular-arc cover: For a collection of n arcs of a given circle C, a minimum-cardinality subset that covers C.

Optimal-area inscribed/circumscribed triangle: For a convex polygon P, the largest-area triangle inscribed in P, or, respectively, the smallest-area triangle circumscribing P.

Min-link path in a polygon: A piecewise-linear path of fewest "links" inside a simple polygon between two given points p and q; see Sections 23.4 and 24.3.

TABLE 36.6.3 Parallel geometric optimization algorithms.

PROBLEM	MODEL	TIME	WORK	REF
Largest-area empty rectangle	CREW	$O(\log^2 n)$	$O(n \log^3 n)$	[AKPS90]
All-farthest neighbors in polygon	CREW	$O(\log^2 n)$	$O(n \log^2 n)$	[Guh92]
Closest visible-pair btw polygons	CREW	$O(\log n)$	$O(n \log n)$	[HCL92]
Min circular-arc cover	EREW	$O(\log n)$	$O(n \log n)$	[AC89]
Opt-area inscr/circum triangle	CRCW	$O(\log \log n)$	$O(n)$	[CM92]
Opt-area inscr/circum triangle	CREW	$O(\log n)$	$O(n)$	[CM92]
Min-link path in a polygon	CREW	$O(\log n \log \log n)$	$O(n \log n \log \log n)$	[CGM+90]

OPEN PROBLEMS

1. Can the visibility graph of a set of n nonintersecting line segments be constructed using $O(n \log n + m)$ work in time at most $O(\log^2 n)$ in the CREW model, where m is the size of the graph?

2. Can the visibility graph of a triangulated polygon be computed in $O(\log n)$ time using $O(n+m)$ work in the CREW model?

36.7 SOURCES AND RELATED MATERIAL

FURTHER READING

Our presentation has been results-oriented and has not given much in the way of problem intuition or algorithmic techniques. There are several excellent surveys available in the literature [Ata92, AC96, AG93, RS93] that are more techniques-oriented. Another good location for related material is the book by Akl and Lyons [AL93].

RELATED CHAPTERS

Chapter 19: Convex hull computations
Chapter 20: Voronoi diagrams and Delaunay triangulations
Chapter 21: Arrangements
Chapter 23: Polygons
Chapter 30: Point location
Chapter 33: Geometric intersection
Chapter 34: Randomized algorithms
Chapter 38: Linear programming in low dimensions

REFERENCES

[AC89] M.J. Atallah and D.Z. Chen. An optimal parallel algorithm for the minimum circle-cover problem. *Inform. Process. Lett.*, 34:159–165, 1989.

[AC96] M.J. Atallah and D.Z. Chen. Parallel computational geometry. In A.Y. Zomaya, editor, *Parallel Computations: Paradigms and Applications*, pages 162–197, International Thomson Computer Press, Boston, 1996.

[ACG89] M.J. Atallah, R. Cole, and M.T. Goodrich. Cascading divide-and-conquer: A technique for designing parallel algorithms. *SIAM J. Comput.*, 18:499–532, 1989.

[ACW91] M.J. Atallah, D.Z. Chen, and H. Wagener. Optimal parallel algorithm for visibility of a simple polygon from a point. *J. Assoc. Comput. Mach.*, 38:516–553, 1991.

[AG93] M.J. Atallah and M.T. Goodrich. Deterministic parallel computational geometry. In J.H. Reif, editor, *Synthesis of Parallel Algorithms*, pages 497–536. Morgan Kaufmann, San Mateo, 1993.

[AGR94] N.M. Amato, M.T. Goodrich, and E.A. Ramos. Parallel algorithms for higher-dimensional convex hulls. In *Proc. 35th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 683–694, 1994.

[AGR95] N.M. Amato, M.T. Goodrich, and E.A. Ramos. Computing faces in segment and simplex arrangements. In *Proc. 27th Annu. ACM Sympos. Theory Comput.*, pages 672–682, 1995.

[AKPS90] A. Aggarwal, D. Kravets, J.K. Park, and S. Sen. Parallel searching in generalized Monge arrays with applications. In *Proc. 2nd Annu. ACM Sympos. Parallel Algorithms Architect.*, pages 259–268, 1990.

[AL93] S.G. Akl and K.A. Lyons. *Parallel Computational Geometry.* Prentice-Hall, Englewood Cliffs, 1993.

[AM90] N. Alon and N. Megiddo. Parallel linear programming in fixed dimension almost surely in constant time. In *Proc. 31st Annu. IEEE Sympos. Found. Comput. Sci.*, pages 574–582, 1990.

[AP93] N.M. Amato and F.P. Preparata. An NC^1 parallel 3D convex hull algorithm. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 289–297, 1993.

[Ata92] M.J. Atallah. Parallel techniques for computational geometry. *Proc. IEEE*, 80:1435–1448, 1992.

[BM87] L. Boxer and R. Miller. Parallel dynamic computational geometry. Report 87-11, Dept. Comput. Sci., SUNY-Buffalo, 1987.

[BSV96] O. Berkman, B. Schieber, and U. Vishkin. A fast parallel algorithm for finding the convex hull of a sorted point set. *Internat. J. Comput. Geom. Appl.*, 6:231–242, 1996.

[Cal93] P.B. Callahan. Optimal parallel all-nearest-neighbors using the well-seated pair decomposition. In *Proc. 34th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 332–340, 1993.

[CCT92] K.L. Clarkson, R. Cole, and R.E. Tarjan. Randomized parallel algorithms for trapezoidal diagrams. *Internat. J. Comput. Geom. Appl.*, 2:117–133, 1992. Erratum: 2:341–343, 1992.

[CG92] R. Cole and M.T. Goodrich. Optimal parallel algorithms for polygon and point-set problems. *Algorithmica*, 7:3–23, 1992.

[CGM+90] V. Chandru, S.K. Ghosh, A. Maheshwari, V.T. Rajan, and S. Saluja. *NC*-algorithms for minimum link path and related problems. Technical Report CS-90/3, TATA Inst., Bombay, 1990.

[CGÓ90] R. Cole, M.T. Goodrich, and C. Ó'Dúnlaing. Merging free trees in parallel for efficient Voronoi diagram construction. In *Proc. 17th Internat. Colloq. Automata Lang. Program.*, volume 443 of *Lecture Notes in Comput. Sci.*, pages 432–445. Springer-Verlag, Berlin, 1990.

[Che95] D. Chen. Efficient geometric algorithms on the EREW PRAM. *IEEE Trans. Parallel Distrib. Syst.*, 6:41–47, 1995.

[CM92] S. Chandran and D.M. Mount. A parallel algorithm for enclosed and enclosing triangles. *Internat. J. Comput. Geom. Appl.*, 2:191–214, 1992.

[CZ90] R. Cole and O. Zajicek. An optimal parallel algorithm for building a data structure for planar point location. *J. Parallel Distrib. Comput.*, 8:280–285, 1990.

[DK89] N. Dadoun and D.G. Kirkpatrick. Cooperative subdivision search algorithms with applications. In *Proc. 27th Allerton Conf. Commun. Control Comput.*, pages 538–547, 1989.

[GG91] M. Ghouse and M.T. Goodrich. In-place techniques for parallel convex hull algorithms. In *Proc. 3rd Annu. ACM Sympos. Parallel Algorithms Architect.*, pages 192–203, 1991.

[GM91] S.K. Ghosh and A. Maheshwari. An optimal parallel algorithm for determining the intersection type of two star-shaped polygons. In *Proc. 3rd Canad. Conf. Comput. Geom.*, Vancouver, pages 2–6, 1991.

[Goo89] M.T. Goodrich. Triangulating a polygon in parallel. *J. Algorithms*, 10:327–351, 1989.

[Goo91] M.T. Goodrich. Intersecting line segments in parallel with an output-sensitive number of processors. *SIAM J. Comput.*, 20:737–755, 1991.

[Goo92] M.T. Goodrich. Planar separators and parallel polygon triangulation. In *Proc. 24th Annu. ACM Sympos. Theory Comput.*, pages 507–516, 1992.

[Goo93] M.T. Goodrich. Geometric partitioning made easier, even in parallel. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 73–82, 1993.

[Goo96] M.T. Goodrich. Fixed-dimensional parallel linear programming via relative epsilon-approximations. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pages 132–141, 1996.

[GÓY93] M.T. Goodrich, C. Ó'Dúnlaing, and C. Yap. Computing the Voronoi diagram of a set of line segments in parallel. *Algorithmica*, 9:128–141, 1993.

[GSG92] M.T. Goodrich, S. Shauck, and S. Guha. Parallel methods for visibility and shortest path problems in simple polygons. *Algorithmica*, 8:461–486, 1992. Addendum: 9:515–516, 1993.

[Guh92] S. Guha. Parallel computation of internal and external farthest neighbours in simple polygons. *Internat. J. Comput. Geom. Appl.*, 2:175–190, 1992.

[HCL92] F.R. Hsu, R.C. Chang, and R.C.T. Lee. Parallel algorithms for computing the closest visible vertex pair between two polygons. *Internat. J. Comput. Geom. Appl.*, 2:135–162, 1992.

[Her89] J. Hershberger. Finding the upper envelope of n line segments in $O(n \log n)$ time. *Inform. Process. Lett.*, 33:169–174, 1989.

[Her92] J. Hershberger. Optimal parallel algorithms for triangulated simple polygons. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 33–42, 1992.

[HS93] J. Hershberger and S. Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. In *Proc. 4th ACM-SIAM Sympos. Discrete Algorithms*, pages 54–63, 1993.

[J92] J. JáJá. *An Introduction to Parallel Algorithms.* Addison-Wesley, Reading, 1992.

[KR90] R.M. Karp and V. Ramachandran. Parallel algorithms for shared memory machines. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 869–941. Elsevier/The MIT Press, Amsterdam, 1990.

[MS88] R. Miller and Q.F. Stout. Efficient parallel convex hull algorithms. *IEEE Trans. Comput.*, C-37:1605–1618, 1988.

[Rei93] J.H. Reif. *Synthesis of Parallel Algorithms.* Morgan Kaufmann, San Mateo, 1993.

[RS92] J.H. Reif and S. Sen. Optimal parallel randomized algorithms for three-dimensional convex hulls and related problems. *SIAM J. Comput.*, 21:466–485, 1992.

[RS93] S. Rajasekaran and S. Sen. Random sampling techniques and parallel algorithms design. In J.H. Reif, editor, *Synthesis of Parallel Algorithms*, pages 411–452. Morgan Kaufmann, San Mateo, 1993.

[TV91] R. Tamassia and J.S. Vitter. Parallel transitive closure and point location in planar structures. *SIAM J. Comput.*, 20:708–725, 1991.

[WC90] Y.C. Wee and S. Chaiken. An optimal parallel L_1-metric Voronoi diagram algorithm. In *Proc. 2nd Canad. Conf. Comput. Geom.*, Ottawa, pages 60–65, 1990.

37 PARAMETRIC SEARCH

Jeffrey S. Salowe

INTRODUCTION

Parametric search is a technique that can sometimes be used to solve an optimization problem when there is an efficient algorithm for the related decision problem. If successful, one creates an optimization algorithm that makes only a small number of calls to the decision algorithm. We provide a general description (Section 37.1) and four examples (Sections 37.2–37.5) to illustrate the technique.

37.1 PARAMETRIC SEARCH OVERVIEW

GLOSSARY

Monotonic function: A function $f(x)$ having the property that $f(y) \geq f(x)$ if $y > x$.

Root-finding problem: Determining the largest value θ^* of θ with the property that $f(\theta^*) = 0$.

Monotonic root-finding problem: A root-finding problem where $f(\theta)$ is monotonically increasing in θ.

Fixed-value problem: Evaluating $f(\theta)$ for a given value of θ.

Parametric search: A technique to solve efficiently suitable monotonic root-finding problems.

WHAT IS PARAMETRIC SEARCH?

The parametric search technique was invented by Megiddo [Meg79, Meg83] as a technique to solve certain optimization problems. Parametric search is particularly effective if the optimization problem can be phrased as a monotonic root-finding problem and if an efficient algorithm for the fixed-value problem can be constructed.

More specifically, let $f(\theta)$ be a monotonic function with a root, and suppose our optimization problem is to determine $\theta^* = \sup\{\theta \mid f(\theta) = 0\}$. (Our notation emphasizes the dependence on the parameter θ, but it obscures the dependence of certain functions on the problem inputs.) Let $A(\theta)$ be an algorithm that computes $f(\theta)$, written in the form of a binary *decision tree* whose nodes s correspond to inequalities $g_s(\theta) \geq 0$. The parametric search technique evaluates $f(\theta^*)$, and in the process discovers θ^*, by evaluating the sign of $f(\theta)$ at some of the roots of $g_s(\theta)$.

The technique works best when each $g_s(\theta)$ has at most a constant number of roots and when $A(\theta)$ is an efficient parallel algorithm.

WHAT IS ITS EFFECT?

Parametric search generally yields the following results. Suppose that the optimization problem has n inputs and the decision problem has $n+1$ inputs, the additional input being for the parameter θ. If $A(\theta)$ computes $f(\theta)$ sequentially in $S(n)$ time, θ^* can be found in $O((S(n))^2)$ time. If $B(\theta)$ is an efficient parallel algorithm to compute $f(\theta)$ that runs in $T(n)$ time using $P(n)$ processors, θ^* can be found in $O(S(n)T(n)\log P(n) + T(n)P(n))$ time. Under favorable conditions, parametric search solves an optimization problem in $O(\log^c n)$ $f(\theta)$ evaluations, where c is a small constant.

HOW IS IT APPLIED?

It is sometimes difficult to determine whether a given problem can be phrased as a root-finding problem suitable for parametric search. As a guideline, we illustrate the parametric search technique through a series of examples. The examples are picked for their illustrative value, and we do not necessarily derive the most efficient results known. Instead, we demonstrate the efficacy of the technique by obtaining surprisingly efficient solutions. Parametric search was used on the problems mentioned in Sections 37.3–37.5 to substantially improve the time complexity over previous techniques.

37.2 EXAMPLE 1: QUARTERING THE PLANE

GLOSSARY

Planar ham sandwich cut: A line that simultaneously bisects two planar sets. (See Sections 11.2 and 31.2.)

Median: A number $x \in A$ with the property that at most half of the numbers in A are less than x, and at most half of the numbers in A are greater than x.

General position: A condition on a set of points that forbids certain configurations. A typical general position assumption is that no three points in the plane are collinear.

PROBLEM STATEMENT

- Input: Set $U = \{u_1, \ldots, u_n\}$, consisting of n points in the plane, each point satisfying $y(u_i) > 0$, where $y(u)$ is the y-coordinate of point u. Set $L = \{l_1, \ldots, l_n\}$, a set of n points in the plane, each point satisfying $y(l_i) < 0$.

- Output: A planar ham sandwich cut for U and L.

We assume that the points are in general position (no three points collinear and no two points with the same y-coordinates), that the input values are rational, and that n is an odd positive integer. In this case, the ham sandwich cut is unique. These conditions simplify the explanation of the algorithm.

CHOICE OF MONOTONIC FUNCTION

The quartering problem is not immediately in the form of a monotonic root-finding problem, but it can be converted to one in the following manner. Let θ be an angle with respect to the x-axis, $0 < \theta < \pi$, measured in the usual way, and let $m(\theta, U)$ denote the intersection of the x-axis with the line at angle θ that bisects U. Let $m(\theta, L)$ be the analogous quantity for set L. We seek an angle θ such that $f(\theta) = m(\theta, U) - m(\theta, L) = 0$. Because a ham sandwich cut exists, there is a value θ^* of θ for which $f(\theta^*) = 0$; our assumptions above ensure that θ^* is unique.

With this choice of the $f(\theta)$ function, the quartering problem seems to be a good candidate for parametric search. The function $f(\theta)$ is monotonic in θ, the quartering problem is solved if and only if $f(\theta^*) = 0$, and the value of $f(\theta)$ is easily computed, as described below.

FIXED-VALUE EVALUATION

To compute $f(\theta)$, first consider the U points. If these points are projected onto the x-axis along lines at an angle of θ, we have n one-dimensional points. The median of these projected points is precisely $m(\theta, U)$. Similarly, the median of the projected L points is $m(\theta, L)$. The evaluation of $f(\theta)$ amounts to:

1. Determining $m(\theta, U)$ by a median-find procedure.
2. Determining $m(\theta, L)$ by a median-find procedure.
3. Calculating $f(\theta) = m(\theta, U) - m(\theta, L)$.

The median-find procedure is a comparison-based algorithm that runs sequentially in $O(n)$ time and in parallel in $O(\log n)$ time using $O(n/\log n)$ processors. This is our algorithm $A(\theta)$.

THE DECISION-TREE ALGORITHM

We now rewrite $A(\theta)$ as a decision-tree algorithm and examine its comparisons. The median-find algorithm is central to $A(\theta)$. The generic step $s(i, j)$ of the median-find algorithm is to compare α_i and α_j, where α_i and α_j are two of the inputs; here the input values $\alpha_i(\theta)$ and $\alpha_j(\theta)$ are the projections of points $u_i = (x_i, y_i)$ and $u_j = (x_j, y_j)$ along a line with angle θ. It is apparent that $\alpha_i(\theta) = x_i - y_i \cot\theta$ and $\alpha_j(\theta) = x_j - y_j \cot\theta$. The decision tree node $s(i, j)$ corresponds to $g_{s(i,j)}(\theta) = x_i - x_j + (y_j - y_i)\cot\theta \geq 0$. There are no other branch points in the algorithm that depend on θ.

The function $g_{s(i,j)}$ has one root, $\theta_{s(i,j)} = \tan^{-1}\frac{y_j - y_i}{x_j - x_i}$. This is because the function $\cot(\theta)$ is monotonically decreasing in the range $0 < \theta < \pi$ and takes on all

values. Although the exact numerical value of $\theta_{s(i,j)}$ is generally unavailable, the sign of $f(\theta_{s(i,j)})$ can be evaluated. Consider comparison $s(m,n)$ in the computation of $f(\theta_{s(i,j)})$. The value of the function

$$g_{s(m,n)}(\theta_{s(i,j)}) = x_m - x_n + (y_n - y_m)\frac{x_j - x_i}{y_j - y_i}$$

is rational if the inputs are rational. Furthermore, the truth value of $\theta_{s(i,j)} < \theta_{s(i',j')}$ can be determined without the actual numerical values of $\theta_{s(i,j)}$ and $\theta_{s(i',j')}$: the truth value of $\theta_{s(i,j)} < \theta_{s(i',j')}$ is the same as the truth value of

$$\frac{y_j - y_i}{x_j - x_i} < \frac{y_{j'} - y_{i'}}{x_{j'} - x_{i'}}.$$

These two observations are needed below.

EVALUATING $f(\theta^*)$

Recall that we seek θ^*, the value of θ for which $f(\theta^*) = 0$. Suppose we try to run the algorithm $A(\theta^*)$ for $f(\theta^*)$, even though we do not know θ^*. Our main difficulty is resolving comparisons that depend on the value of θ^*.

Algorithm $A(\theta^*)$ is in the form of a decision tree, where each node s is labeled with inequality $g_s(\theta^*) \geq 0$. In order to resolve these decisions, we must determine the truth values of $g_s(\theta^*) \geq 0$.

These truth values are determined as follows. (This is the crucial step in parametric search.) The function $g_s(\theta)$ has one root, θ_s. Furthermore, $g_s(\theta)$ is monotonically decreasing in θ, so we can therefore determine the truth value of $g_s(\theta^*) \geq 0$ by determining the relative values of θ^* and θ_s. The relative values of θ^* and θ_s can be inferred by evaluating the sign of the fixed-value problem $f(\theta_s)$. Because $f(\theta)$ is monotonic, $f(\theta_s) < 0$ implies that $\theta_s < \theta^*$, and $f(\theta_s) > 0$ implies that $\theta_s > \theta^*$. If $f(\theta_s) = 0$, then $\theta^* = \theta_s$, and we have the value we seek. As stated above, the sign of $f(\theta_s)$ can be determined at the roots of $g_s(\theta)$.

Let $A(\theta^*)$ be based on a sequential median-find algorithm. Algorithm $A(\theta^*)$ runs in $O(n)$ time, but each comparison s evaluates the truth value of inequality $g_s(\theta^*) \geq 0$ by computing the sign of $f(\theta_s)$. The sign of $f(\theta_s)$ can be found in $O(n)$ time, so $A(\theta^*)$ runs in $O(n^2)$ time, even though the exact value of θ^* is unknown until the end of the computation.

IMPROVEMENTS USING PARALLELISM

We can decrease the time complexity of the algorithm by replacing the usual median-find procedure with a sequentialized version of a parallel algorithm. It is possible to devise a median-find procedure that uses $O(n/\log n)$ processors, completes in $O(\log n)$ time, and can be simulated in $O(n)$ sequential time. (Note that there are algorithms with better bounds that cannot be simulated in $O(n)$ sequential time.)

The advantage of a parallel algorithm is that the comparisons on a particular time step can be evaluated in an arbitrary order. Let

$$g_{s_1}(\theta^*) \geq 0, g_{s_2}(\theta^*) \geq 0, \ldots, g_{s_{n/\log n}}(\theta^*) \geq 0$$

be the comparisons on time step j. Rather than evaluating each of them by computing $f(\theta_{s_i})$, $1 \leq i \leq n/\log n$, we evaluate the one with median θ_s value. (This is where we need to order the θ_s values.) This comparison can be used to infer the truth value of half of the remaining comparisons. That is, we evaluate the comparisons by performing a binary search for θ^* among the θ_{s_i} values.

The time complexity of the new algorithm is as follows. A total of $O(n)$ comparisons must be evaluated, organized so that $O(n/\log n)$ comparisons are made per time step for a duration of $O(\log n)$ time steps. During each time step, binary search resolves $O(\log n)$ comparisons by actually computing the sign of $f(\theta_s)$, and the rest of the comparisons are decided by transitivity. There are consequently $O(n \log n)$ operations per time step, multiplied by $O(\log n)$ time steps, giving a total of $O(n \log^2 n)$ operations.

FURTHER IMPROVEMENTS

This problem can be attacked with the related "prune-and-search" technique. If the proper comparisons are done, it is possible to reduce the size of the original problem and solve a substantially-smaller subproblem. The resulting time complexity is $O(n)$.

37.3 EXAMPLE 2: SELECTING VERTICES IN ARRANGEMENTS

GLOSSARY

Selection problem: Given a totally ordered set S and an integer k, $1 \leq k \leq |S|$, the selection problem is to find θ^*, the kth smallest item in S.

Ranking problem: Given a totally ordered set S and a number θ, the ranking problem is to return the number of items rank(θ, S) in S whose value is less than or equal to θ.

Arrangement: The subdivision of space induced by a set of hyperplanes. (See Chapter 21.)

Permutation: A sequence of n distinct integers in the range 1 through n.

Inversion: A pair (i, j) occurring in a permutation where $i < j$ but j precedes i in the permuted sequence.

PROBLEM STATEMENT

- Input: Set $H = \{h_1, \ldots, h_n\}$ of lines in the plane, where h_i has equation $y = m_i x + b_i$, and the lines are indexed in order of increasing slope. Integer k, $1 \leq k \leq \binom{n}{2}$.
- Output: Let V be the intersection points (vertices) of the arrangement formed by H. The output is the vertex v^* whose x-coordinate has rank k among the x-coordinates in V.

We assume that m_i and b_i are rational, and that H is in general position, so that no three lines intersect in a single vertex, no two vertices have the same x-coordinate, no line is vertical, and no two lines are parallel.

CHOICE OF MONOTONIC FUNCTION

Consider the function $f(\theta) = \text{rank}(\theta, V) - k$. This function is monotonically nondecreasing in θ, and it has the property that θ^*, the x-coordinate of v^*, satisfies $\theta^* = \sup\{\theta \mid f(\theta) = 0\}$.

FIXED-VALUE EVALUATION

Evaluating $f(\theta) = \text{rank}(\theta, V)$ amounts to counting the number of vertices in V whose x-coordinates are less than input θ. This can be done in the following way. The y-intercepts of the intersections of H with the line $x = \theta$ are the numbers $m_i\theta + b_i$, $1 \le i \le n$. If these numbers are sorted in decreasing order and value $m_i\theta + b_i$ is replaced by index i, the result is a permutation $\pi(\theta)$. The key insight is that the number of inversions in $\pi(\theta)$ equals $\text{rank}(\theta, V)$.

Algorithm $A(\theta)$, the algorithm to determine $f(\theta)$, consists of:

1. Computing the permutation $\pi(\theta)$.
2. Counting the number of inversions in $\pi(\theta)$.
3. Subtracting k from this result.

The first step is essentially a sorting step, which can be done sequentially in $O(n \log n)$ time and in parallel in $O(\log n)$ time with $O(n)$ processors. The second step can be done by a mergesort-like procedure.

THE DECISION-TREE ALGORITHM

The first step of algorithm $A(\theta)$ depends on the value of θ. Once the permutation $\pi(\theta)$ is computed, the control flow of the second and third steps does not depend on θ.

The comparisons $s(i, j)$ in $A(\theta)$ ask whether i precedes j in the permutation: Is $m_i\theta + b_i \ge m_j\theta + b_j$? We rewrite this inequality as

$$g_{s(i,j)}(\theta) = (m_i - m_j)\theta + (b_i - b_j) \ge 0\,.$$

It is clear that $g_{s(i,j)}(\theta)$ has a root $\theta_{s(i,j)}$ at $\frac{b_j - b_i}{m_i - m_j}$ (recall that no two lines have the same slope). The sign of $m_i - m_j$ is negative, implying that the functions $g_{s(i,j)}(\theta)$ are monotonically nonincreasing. The root $\theta_{s(i,j)}$ is rational, so evaluating the sign of $f(\theta_{s(i,j)})$ or comparing $\theta_{s(i,j)}$ values poses no difficulty.

EVALUATING $f(\theta^*)$

Suppose we attempt to evaluate $f(\theta^*)$ at the unknown x-coordinate θ^*. The chief difficulty is resolving comparisons involving θ^*. These comparisons correspond to inequalities of the form $g_{s(i,j)}(\theta^*) \ge 0$.

The inequality $g_{s(i,j)}(\theta^*) \geq 0$ is the same as the inequality $\theta^* \geq \theta_{s(i,j)}$. We can determine the truth value of this inequality by evaluating $f(\theta_{s(i,j)})$. Because $f(\theta)$ is monotonic, $f(\theta_{s(i,j)}) < 0$ implies that $\theta_{s(i,j)} < \theta^*$, and $f(\theta_{s(i,j)}) > 0$ implies that $\theta_{s(i,j)} > \theta^*$. Otherwise, $f(\theta_{s(i,j)}) = 0$, and $\theta^* = \theta_{s(i,j)}$.

A sequential implementation of algorithm $A(\theta^*)$ evaluates $O(n \log n)$ comparisons. Each comparison at node $s(i,j)$ determines the sign of $f(\theta_{s(i,j)})$, an operation that takes $O(n \log n)$ time. Step one therefore takes $O(n^2 \log^2 n)$ time to simulate. The rest of the work, steps two and three, takes additional $O(n \log n)$ time steps. The total work is $O(n^2 \log^2 n)$.

IMPROVEMENTS USING PARALLELISM

There are efficient parallel sorting algorithms; it is possible to sort n numbers in $O(\log n)$ time using n processors. If we perform a binary search on the n comparisons per level, only $O(\log n)$ $f(\theta)$-evaluations are done, and the remaining comparisons are resolved by transitivity. The work per level is $O(n \log^2 n)$. There are $O(\log n)$ levels, so the time complexity of this algorithm is $O(n \log^3 n)$.

FURTHER IMPROVEMENTS

Cole [Col87b] gave a general technique that can be used to remove a log factor from the time complexity. If a parallel algorithm can be described by a circuit with constant fan-out gates (say fan-out two), then the following trick can be applied. Suppose that $\frac{c-1}{c}$ of the comparisons on the first time step have been resolved; then the inputs of at least $\frac{c-2}{c}$ of the comparisons on the second time step are available, and these comparisons are also ready to be resolved. Cole's idea is to combine these newly-ready comparisons with the unresolved comparisons. The total number of comparisons that need to be resolved by actually evaluating $f(\theta)$ becomes $O(\log P(n) + T(n))$. With respect to the sorting problem, the parallel sorting algorithm can be written as a circuit with fan-out two, so a total of $O(\log n)$ function evaluations need to be performed.

A second log factor can be removed by approximate ranking. Rather than computing the number of inversions exactly, the number is approximated. This approximation is sufficiently precise to determine the relative values of θ_s and θ^*. The resulting time complexity is $O(n \log n)$.

37.4 EXAMPLE 3: SELECTING INTERDISTANCES

GLOSSARY

L_p interdistance: Given points $a = (a_1, a_2, \ldots, a_d)$ and $b = (b_1, b_2, \ldots, b_d)$, $1 \leq p < \infty$, the L_p interdistance between a and b is given by

$$\|a-b\|_p = \left(\sum_{i=1}^{d} |a_i - b_i|^p \right)^{1/p}.$$

L_∞ interdistance: Given points a and b as above, the L_∞ interdistance between a and b is given by

$$\|a-b\|_\infty = \max_{1 \le i \le d} \{|a_i - b_i|\}.$$

$\tilde{\mathbf{O}}(f(n))$: The set of functions that are $O(f(n)^{1+\epsilon})$, for any $\epsilon > 0$.

PROBLEM STATEMENT

- Input: Set P of n points in the plane. Integer k, $1 \le k \le \binom{n}{2}$.
- Output: Let D be the L_p interdistances formed by the points in P. The output is the interdistance θ^* with rank k in D.

We assume that all interdistances are unique.

CHOICE OF MONOTONIC FUNCTION

As in the vertex selection problem, the function $f(\theta) = \text{rank}(\theta, D) - k$.

FIXED-VALUE EVALUATION

The ranking problem in either metric can be viewed as a problem involving balls and points. Place a ball of radius θ around each point in P; then $\text{rank}(\theta, D)$ is one-half times the number of point-ball containments. Do not include the center point-ball containments in this total. In the L_∞ metric, the unit ball is a square, and in the L_2 metric, the unit ball is a circle.

We deal with the L_∞ problem first. Ranking can be done efficiently by merging the x-coordinates of the vertical box sides of radius θ with the x-coordinates $\{x_1, \ldots, x_n\}$ of P, and then repeating this process with the y-coordinates. (Assume that $x_1, \ldots, x_n$ are presorted.) Given these sorted orders, we can simulate a sweep-line algorithm that counts the number of point-square containments.

The L_2 ranking problem is somewhat harder, but the basic strategy is identical to the L_∞ case. To rank θ, we form an arrangement of circles, each circle of radius θ and centered about a distinct point in P. Assume that this arrangement can be built and preprocessed for planar point location, and assume that each region of the arrangement is labeled with the number of circles that contain it. For each point in P, perform a point location query to determine how many circles contain it.

Suppose there are s circles and t points. The arrangement can be built in $O(s^2)$ time, and each point location query can be answered in $O(\log s)$ time. The total processing time is $O(s^2 + t \log s)$.

Our ranking problem consists of n circles and points. If we divide the set of circles into $O(\sqrt{n})$ groups of size $O(\sqrt{n})$ and perform the procedure above, ranking can be performed in $\tilde{O}(n^{3/2})$ time.

THE DECISION-TREE ALGORITHM

The first step in the L_∞ ranking algorithm is to sort the values $\{x_1, \ldots, x_n, x_1 - \theta, \ldots, x_n - \theta\}$ and to sort the analogous y-coordinates. Some of these comparisons $s(i,j)$ depend on θ; they are of the form $x_i \geq x_j - \theta$. This implies that $g_{s(i,j)}(\theta) = \theta + x_i - x_j$, and the root of $g_{s(i,j)}(\theta)$ is $\theta_{s(i,j)} = x_j - x_i$. After these two sorted orders are known, the remainder of the algorithm does not depend on θ.

The L_2 algorithm is more complicated. The construction of the circular arrangement contains some steps that depend on θ. A typical such step $s(z, C)$ involves the comparison of a point with a circle: Does point $z = (z_1, z_2)$ lie inside of circle C? Let the center of circle C be (c_1, c_2). Deciding if z lies on or inside circle C of radius θ is equivalent to determining the truth value of the inequality $(z_1 - c_1)^2 + (z_2 - c_2)^2 \leq \theta^2$, so $g_{s(z,C)}(\theta) = \theta^2 - (z_1 - c_1)^2 - (z_2 - c_2)^2$. Function $g_s(z, C)$ has roots at $\pm\theta_{s(z,C)} = \pm\sqrt{(z_1 - c_1)^2 + (z_2 - c_2)^2}$.

EVALUATING $f(\theta^*)$

As in vertex selection, we perform interdistance selection by ranking unknown interdistance θ^*. For the L_∞ problem, the only step that needs the value of θ^* is the merging step; here, comparisons of the form $x_i \geq x_j - \theta^*$ must be resolved. This comparison is precisely $\theta_{s(i,j)} \leq \theta^*$, which we can resolve by evaluating $f(\theta_{s(i,j)})$.

The cost of presorting the data is $O(n \log n)$, and there are $O(n)$ comparisons in the merging steps, each comparison taking $O(n)$ time. Parametric search takes $O(n \log n + n^2) = O(n^2)$ time.

For the L_2 problem, comparisons of the form $(z_1 - c_1)^2 + (z_2 - c_2)^2 \leq (\theta^*)^2$ must be resolved. This comparison is precisely $(\theta_{s(z,C)})^2 \leq (\theta^*)^2$. Since $f(-\theta_{s(z,C)}) = -k$, this comparison can be resolved by evaluating the sign of $f(+\theta_{s(z,C)})$. Note that the square root is not needed in this evaluation because θ is squared in the functions $g_{s(z,C)}$.

The description of the L_2 ranking problem included an analysis of its time complexity. The ranking algorithm makes $\tilde{O}(n^{3/2})$ comparisons, each taking $\tilde{O}(n^{3/2})$ time, for a total of $\tilde{O}(n^3)$ time.

IMPROVEMENTS USING PARALLELISM

In the L_∞ algorithm, only the merging step needs to be parallelized. This can be done in $O(\log n)$ time using $O(n/\log n)$ processors. A straightforward application of parametric search gives an $O(n \log^3 n)$ time algorithm.

With respect to the L_2 algorithm, it is possible to devise a parallel algorithm that uses $O(n^{3/2})$ processors and $O(\log n)$ time. Consequently, only $O(\log^2 n)$ comparisons need to be resolved by ranking. The total time is only $\tilde{O}(n^{3/2})$.

FURTHER IMPROVEMENTS

Cole's trick removes one log factor from the L_∞ algorithm, giving an $O(n \log^2 n)$ time algorithm. A different ranking scheme, one based on epsilon nets (Sections 31.2 and 34.4), is used to obtain better ranking results for the L_2 problem. The resulting time complexity is $\tilde{O}(n^{4/3})$.

37.5 EXAMPLE 4: RAY SHOOTING

GLOSSARY

Ray shooting: Determining the first object intersected by a ray (Chapter 32).

Partition tree: A data structure for simplex range queries (Section 31.2).

PROBLEM STATEMENT

- Input: A set H of n hyperplanes in k-dimensional space. A query ray ρ with origin o.
- Output: The first hyperplane of H that ρ intersects.

It is intended that the queries be repeated many times, so we want a data structure with small query time. We assume that o is not contained in a hyperplane of h and that $\rho \cap H \neq \emptyset$.

CHOICE OF MONOTONIC FUNCTION

Let ray ρ be given by its origin o and an arbitrary point $o(1)$ on ρ. For nonnegative θ, let $\rho(\theta)$ be the open subsegment of ρ given by $(1-\lambda)o + \lambda o(1)$, $0 < \lambda < \theta$. (We will call the nonorigin endpoint $o(\theta)$). Let

$$f(\theta) = \begin{cases} [\,|\{h \in H \mid \rho(\theta) \cap h \neq \emptyset\}| \geq 1\,], & \theta \geq 0 \\ 0, & \theta < 0\,. \end{cases}$$

Here, $[P(x)] = 1$ if predicate $P(x)$ is true and 0 if $P(x)$ is false. The set $\{h \in H \mid \rho(\theta) \cap h \neq \emptyset\}$ consists of the hyperplanes in H that $\rho(\theta)$ intersects.

FIXED-VALUE EVALUATION

We now address the issue of efficiently computing $f(\theta)$. A reasonable data structure for such a task is a partition tree, described in Chapter 31.

Let X be a set of points. Each node r in the partition tree corresponds to a set $X(r) \subseteq X$; the root corresponds to X. Furthermore, each node r is associated with a region of space $J(r)$, usually a simplex.

The partition tree can be used to compute $f(\theta)$ by determining whether the endpoints of $\rho(\theta)$, o and $o(\theta)$, lie in the same cell of arrangement $\mathcal{A}(H)$. To do this, the hyperplanes in H are dualized to a set of points $\mathcal{D}(H)$, and a partition tree is constructed for $\mathcal{D}(H)$. Points o and $o(\theta)$ lie in the same cell of $\mathcal{A}(H)$ if the double-wedge, the dual of the segment connecting o and $o(\theta)$, does not contain any points of $\mathcal{D}(H)$.

THE DECISION-TREE ALGORITHM

The basic step of the algorithm above compares the position of hyperplane $\mathcal{D}(o(\theta))$ to a point p, one of the vertices of $J(r)$. This is tantamount to deciding whether point $o(\theta)$ and o are on the same side of a particular hyperplane $h = \mathcal{D}(p)$.

Let h be given by $n_h \cdot x = \alpha_h$, where n_h is the unit normal of h in the direction of o. Then $o(\theta)$ and o are on the same side of h if $g_h(\theta) = n_h \cdot o(\theta) - \alpha_h \geq 0$.

The function $g_h(\theta)$ has a single root at

$$\theta_h = \frac{\alpha_h - (n_h \cdot o)}{n \cdot (o(1) - o)}.$$

The sign of $f(\theta_h)$ can be evaluated when the components of n and $o(1)$ are rational.

EVALUATING $f(\theta^*)$

Given ray ρ, we seek the value of $o(\theta^*)$, the location of the first intersection of ρ with a hyperplane in h. The number of points inside the double-wedge for $\rho(\theta^*)$ can be computed by resolving comparisons of the form

$$g_h(\theta^*) = n_h \cdot \rho(\theta^*) - \alpha_h \geq 0 = g_h(\theta_h).$$

This comparison is the same as determining the truth value of $\theta^* \geq \theta_h$, which is decided by evaluating the sign of $f(\theta_h)$. As stated above, the partition tree is used to evaluate $f(\theta_h)$.

Let $B(n)$ be the cost of constructing a partition tree on H, let $C(n)$ be the amount of storage needed, and let $Q(n)$ be the cost of querying the partition tree. Preprocessing does not depend on θ, and it takes $B(n)$ time. After preprocessing, the number of operations necessary to evaluate each $f(\theta)$ is $Q(n)$, and there are $Q(n)$ such evaluations in computing θ^*. The total number of operations in the parametric search is $O(B(n) + Q(n)^2)$.

PARALLEL ALGORITHM

Suppose that the query algorithm can be parallelized so that it runs in $T(n)$ time on $P(n)$ processors. In this case, $B(n)$ time is spent preprocessing the data structure, and $T(n)P(n)$ comparisons are made. Of these comparisons, $T(n) \log P(n)$ are resolved by computing the sign of $f(\theta_h)$, and the rest are resolved by transitivity. This version of the parametric ray-shooting algorithm takes $D(n) = O(Q(n)T(n) \log P(n) + T(n)P(n) + B(n))$ time.

Using results on partition trees, one can construct a family of parametric search algorithms parameterized by m, $n \leq m \leq n^d$, whose preprocessing, storage, and query requirements are $B(n) = \tilde{O}(m)$, $C(n) = \tilde{O}(m)$, and $D(n) = \tilde{O}(\frac{n}{m^{1/d}})$, respectively.

37.6 OTHER RESULTS

We summarize in Table 37.6.1 some of the results obtained with the parametric

search technique on computational geometry problems. Parametric search has been successfully applied in other domains as well.

TABLE 37.6.1 Selected parametric search results.

PROBLEM NAME	INPUT	COMPLEXITY	SOURCE
3-dim set diameter	n points in 3-dim	$O(n \log^3 n)$	[BCM93]
Minimum-width annulus	n points in plane	$\tilde{O}(n^{8/5})$	[AST94]
Collision btw two polyhedra	two polyhedra, n vertices total	$\tilde{O}(n^{8/5})$	[ST95]
Biggest stick	n-sided simple polygon	$\tilde{O}(n^{8/5})$	[AST94]
L_p interdistance selection	n points in plane, k	$\tilde{O}(n^{4/3})$	[AASS93]
L_∞ interdistance selection	n points in plane, k	$O(n \log^2 n)$	[Sal89]
Min Hausdorff dist btw polygons	n- and m-sided simple poly	$\tilde{O}((mn)^2)$	[AST94]
2-center	n points in plane	$O(n^2 \log n)$	[JK94]
Center point in plane	n points in plane	$O(n)$	[JM94]
Segment center in plane	n segments in plane	$\tilde{O}(n)$	[ES96]
Selecting verts in arrangements	n lines in plane, k	$O(n \log n)$	[CSSS89]

Parametric search is not limited to monotonic functions of a single parameter—there is also a multidimensional version of parametric search. An instance of its application in computational geometry appears in Matoušek [Mat93].

37.7 SOURCES AND RELATED MATERIAL

FURTHER READING

The example in Section 37.2 was drawn from Cole [Col87a], where it was used as the basis of a multidimensional partitioning algorithm. Megiddo [Meg85] discovered the linear-time algorithm for quartering the plane. The example in Section 37.3, usually known as *slope selection*, is from Cole et al. [CSSS89]. The interdistance examples in Section 37.4 are from Agarwal et al. [AASS93] and Salowe [Sal89]. Agarwal et al. discovered the L_2 algorithm, and Salowe described the L_∞ algorithm. Finally, the ray-shooting example in Section 37.5 is from Agarwal and Matoušek [AM93].

A good bibliography of parametric search in computational geometry appears in Agarwal et al. [AST94].

RELATED CHAPTERS

Chapter 30: Point location
Chapter 31: Range searching
Chapter 38: Linear programming in low dimensions

REFERENCES

[AASS93] P.K. Agarwal, B. Aronov, M. Sharir, and S. Suri. Selecting distances in the plane. *Algorithmica*, 9:495–514, 1993.

[AM93] P.K. Agarwal and J. Matoušek. Ray shooting and parametric search. *SIAM J. Comput.*, 22:794–806, 1993.

[AST94] P.K. Agarwal, M. Sharir, and S. Toledo. Applications of parametric searching in geometric optimization. *J. Algorithms*, 17:292–318, 1994.

[BCM93] H. Brönnimann, B. Chazelle, and J. Matoušek. Product range spaces, sensitive sampling, and derandomization. In *Proc. 34th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 400–409, 1993.

[Col87a] R. Cole. Partitioning point sets in arbitrary dimensions. *Theoret. Comput. Sci.*, 49:239–265, 1987.

[Col87b] R. Cole. Slowing down sorting networks to obtain faster sorting algorithms. *J. Assoc. Comput. Mach.*, 34:200–208, 1987.

[CSSS89] R. Cole, J. Salowe, W. Steiger, and E. Szemerédi. An optimal-time algorithm for slope selection. *SIAM J. Comput.*, 18:792–810, 1989.

[ES96] A. Efrat and M. Sharir. A near-linear algorithm for the planar segment center problem. *Discrete Comput. Geom.*, 16:239–258, 1996.

[JK94] J.W. Jaromczyk and M. Kowaluk. An efficient algorithm for the Euclidean two-center problem. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 303–311, 1994.

[JM94] S. Jadhav and A. Mukhopadhyay. Computing a centerpoint of a finite planar set of points in linear time. *Discrete Comput. Geom.*, 12:291–312, 1994.

[Mat93] J. Matoušek. Linear optimization queries. *J. Algorithms*, 14:432–448, 1993.

[Meg79] N. Megiddo. Combinatorial optimization with rational objective functions. *Math. Oper. Res.*, 4:414–424, 1979.

[Meg83] N. Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *J. Assoc. Comput. Mach.*, 30:852–865, 1983.

[Meg85] N. Megiddo. Partitioning with two lines in the plane. *J. Algorithms*, 6:430–433, 1985.

[Sal89] J. Salowe. L_∞ interdistance selection by parametric search. *Inform. Process. Lett.*, 30:9–14, 1989.

[ST95] E. Schömer and C. Thiel. Efficient collision detection for moving polyhedra. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 51–60, 1995.

APPLICATIONS OF DISCRETE AND COMPUTATIONAL GEOMETRY

38 LINEAR PROGRAMMING IN LOW DIMENSIONS

Martin Dyer and Nimrod Megiddo

INTRODUCTION

Linear programming has many important practical applications, and has also given rise to a wide body of theory. See Section 38.8 for recommended sources. Here we consider the linear programming problem in the form of maximizing a linear function of d variables subject to n linear inequalities. We focus on the relationship of the problem to computational geometry, i.e., we consider the problem in small dimension. More precisely, we concentrate on the case where $d \ll n$, i.e., $d = d(n)$ is a function that grows very slowly with n. By linear programming duality, this also includes the case $n \ll d$. This has been called *fixed-dimensional* linear programming, though our viewpoint here will not treat d as constant. In this case there are strongly polynomial algorithms, provided the rate of growth of d with n is small enough.

The plan of the chapter is as follows. In Section 38.2 we consider the simplex method, in Section 38.3 we review deterministic linear time algorithms, in Section 38.4 randomized algorithms, and in Section 38.5 we consider the derandomization of the latter. In Section 38.6 we examine parallel algorithms for this problem, and finally in Section 38.7 we briefly discuss related issues. The emphasis throughout is on complexity-theoretic bounds for the linear programming problem in the form 38.1.1.

38.1 THE BASIC PROBLEM

Any linear program (LP) may be expressed in the inequality form

$$\begin{aligned} \text{maximize} \quad & z = c.x \\ \text{subject to} \quad & Ax \leq b\,, \end{aligned} \tag{38.1.1}$$

where $c \in \mathbb{R}^d$, $b \in \mathbb{R}^n$, and $A \in \mathbb{R}^{n \times d}$ are the input data and $x \in \mathbb{R}^d$ the variables. Without loss of generality, the columns of A are assumed to be linearly independent. The vector inequality in (38.1.1) is with respect to the componentwise partial order on $\mathbb{R}^n$. We will write a_i for the ith row of A, so the constraint may also be expressed in the form

$$a_i.x = \sum_{j=1}^{d} a_{ij}x_j \leq b_i \quad (i = 1, \ldots, n). \tag{38.1.2}$$

GLOSSARY

Constraint: A condition that must be satisfied by a solution.

Inequality form: The formulation of the linear programming problem where all the constraints are weak inequalities $a_i.x \leq b_i$.

Feasible set: The set of points that satisfy all the constraints. In the case of linear programming, it is a convex polyhedron in $\mathbb{R}^d$.

Defining hyperplanes: The hyperplanes described by the equalities $a_i.x = b_i$.

Tight constraint: An inequality constraint is tight at a certain point if the point lies on the corresponding hyperplane.

Infeasible problem: A problem with an empty feasible set.

Unbounded problem: A problem with no finite maximum.

Vertex: A feasible point where at least d linearly independent constraints are tight.

Nondegenerate problem: A problem where at each vertex precisely d constraints are tight.

Strongly polynomial-time algorithm: An algorithm for which the total number of arithmetic operations and comparisons (on numbers whose size is polynomial in the input length) is bounded by a polynomial in n and d alone.

We observe that (38.1.1) may be infeasible or unbounded, or have multiple optima. A complete algorithm for linear programming must take account of these possibilities. In the case of multiple optima, we assume that we have merely to identify *some* optimum solution. (The task of identifying *all* optima is considerably more complex; see [Dye83].) An optimum of (38.1.1) will be denoted by x^0. At least one such solution (assuming one exists) is known to lie at a vertex of the feasible set. There is little loss in assuming nondegeneracy for theoretical purposes, since we may "infinitesimally perturb" the problem to ensure this using well known methods [Sch86]. However, a complete algorithm must recognize and deal with this possibility.

It is well known that linear programs can be solved in time polynomial in the total length of the input data. However, it is not known in general if there is a *strongly* polynomial-time algorithm. This is true even if randomization is permitted. (Algorithms mentioned below may be assumed deterministic unless otherwise stated.) The "weakly" polynomial algorithms make crucial use of the size of the numbers, so seem unlikely to lead to strongly polynomial methods. However, strong bounds are known in some special cases. For example, if all a_{ij} are bounded by a constant, then E. Tardos has given a strongly polynomial algorithm.

38.2 THE SIMPLEX METHOD

GLOSSARY

Simplex method: For a nondegenerate problem in inequality form, this method seeks an optimal vertex by iteratively moving from one vertex to a better neighboring vertex.

Pivot rule: The rule by which a neighboring vertex is chosen.

Randomized simplex algorithm: A randomized variant of the simplex method where the neighboring vertex is chosen uniformly at random.

Dantzig's simplex method is probably still the most commonly used method for solving large linear programs in practice, but (with standard pivot rules) Klee and Minty showed that Dantzig's pivot rule may require an exponential number of iterations in the worst case. For example, it may require 2^d iterations when $n = 2d$. Other variants were subsequently shown to have similar behavior. While it is not known for certain that all suggested variants of the simplex method have this bad worst case, there seems to be no reason to believe otherwise. In our case $d \ll n$, the simplex method may require $\Omega(n^{\lfloor d/2 \rfloor})$ iterations, and thus it is polynomial only for $d = O(1)$. This is asymptotically no better than enumerating all vertices of the feasible region.

By contrast, Kalai [Kal92] gave a randomized simplex-like algorithm that requires only $2^{O(\sqrt{d \log n})}$ iterations. (An identical bound was also given by Matoušek, Sharir, and Welzl [MSW92] for a closely related algorithm; see Section 38.4.) This is the best "strong" bound known, other than for various special problems. For $d \ll n$, Kalai's algorithm is evidently polynomial provided $d = O(\log n)$. No complete derandomization of Kalai's algorithm is known, and it is possible that randomization may genuinely help here. In this respect, the complexity of the so-called randomized simplex method (where the pivot is chosen uniformly at random) is an open question. See [BDF+95, GZ95] for some limited information.

38.3 LINEAR-TIME LINEAR PROGRAMMING

The study of linear programming within computational geometry was initiated by Shamos [Sha78] as an application of an $O(n \log n)$ convex hull algorithm for the intersection of halfplanes. Muller and Preparata [MP78] gave an $O(n \log n)$ algorithm for the intersection of halfspaces in $\mathbb{R}^3$. Dyer [Dye84] and Megiddo [Meg83] found, independently, an $O(n)$ time algorithm for the linear programming problem in the cases $d = 2, 3$.

Megiddo [Meg84] generalized the approach of these algorithms to arbitrary d, arriving at an algorithm of complexity $O(2^{2^d} n)$, which is polynomial for $d \leq \log \log n + O(1)$. This was subsequently improved by Clarkson [Cla86b] and Dyer [Dye86] to $O(3^{d^2} n)$, which is polynomial for $d = O(\sqrt{\log n})$. Megiddo [Meg84, Meg89] and Dyer [Dye86, Dye92] showed that Megiddo's idea could be used for many related problems: Euclidean one-center, separation of polyhedra, minimum ball containing balls, minimum volume ellipsoid, etc.

GLOSSARY

Multidimensional search: Given a set of hyperplanes and an oracle for locating a point relative to any hyperplane, locate the point relative to all the input hyperplanes.

The basic idea in these algorithms is as follows. It follows from convexity considerations that either the constraints in a linear program are tight (i.e., satisfied with equality) at x^0, or they are irrelevant. We need identify only d tight constraints to identify x^0. We do this by discarding a fixed proportion of the irrelevant constraints at each iteration. Determining whether the ith constraint is tight amounts to determining which case holds in $a_i.x^0 \gtreqless b_i$. This is embedded in a multidimensional search problem. Given *any* hyperplane $\alpha.x = \beta$, we can determine which case of $\alpha.x^0 \gtreqless \beta$ holds by (recursively) solving three linear programs in $d-1$ variables. These are (38.1.1) plus $a.x = \gamma$, where $\gamma \in \{\beta - \epsilon, \beta, \beta + \epsilon\}$ for "small" $\epsilon > 0$. (We need not define ϵ explicitly; it can be handled symbolically.) In each of the three linear programs we eliminate one variable to get $d-1$. The largest of the three objective functions tells us where x^0 lies with respect to the hyperplane. We call this an *inquiry* about $\alpha.x = \beta$. The problem now reduces to locating x^0 with respect to a proportion $P(d)$ of the n hyperplanes using only $N(d)$ inquiries.

The method recursively uses the following observation in $\mathbb{R}^2$. Given two lines through the origin with slopes of opposite sign, knowing which quadrant a point lies in allows us to locate it with respect to *at least one* of the lines (see Figure 38.3.1).

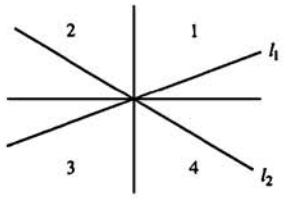


FIGURE 38.3.1
Quadrants $1, 3$ *locate for* l_2*; quadrants* $2, 4$ *locate for* l_1.

We use this on the first two coordinates of the problem in $\mathbb{R}^d$. First rotate until $\frac{1}{2}n$ defining hyperplanes have positive and $\frac{1}{2}n$ negative "slopes" on these coordinates. This can be done in $O(n)$ time using median-finding. Then arbitrarily pair a positive with a negative to get $\frac{1}{2}n$ pairs of the form

$$\begin{aligned} ax_1 + bx_2 + \cdots &= \cdots \\ cx_1 - dx_2 + \cdots &= \cdots, \end{aligned}$$

where a, b, c, d represent nonnegative numbers, and the $\cdots$ represent linear functions of $x_3, \ldots, x_d$ on the left and arbitrary numbers on the right. Eliminating x_2 and x_1 in each pair gives two families S_1, S_2 of $\frac{1}{2}n$ hyperplanes each in $d-1$ dimensions of the form

$$\begin{aligned} S_1: \quad x_1 \quad &+ \cdots = \cdots \\ S_2: \quad x_2 \quad &+ \cdots = \cdots. \end{aligned}$$

We recursively locate with respect to $\frac{1}{2}P(d-1)n$ with $N(d-1)$ inquiries in S_1, and then locate with respect to the corresponding paired hyperplanes in S_2. We have then located $\frac{1}{2}P(d-1)^2 n$ pairs with $2N(d-1)$ inquiries. Using the observation above, each pair gives us location with respect to at least one hyperplane in d

dimensions, i.e.,

$$P(d) = \tfrac{1}{2}P(d-1)^2, \qquad N(d) = 2N(d-1). \tag{38.3.1}$$

Since $P(1) = \frac{1}{2}, N(1) = 1$ (by locating with respect to the median in $\mathbb{R}^1$), (38.3.1) yields

$$P(d) = 2^{-(2^d-1)}, \qquad N(d) = 2^{d-1},$$

giving the following time bound $T(n,d)$ for solving (38.1.1).

$$T(n,d) \leq 3 \cdot 2^{d-1}T(n,d-1) + T((1-2^{-(2^d-1)})n,d) + O(nd),$$

with solution $T(n,d) = O(2^{2^d}n)$.

THE CLARKSON-DYER IMPROVEMENT

The Clarkson/Dyer improvement comes from repeatedly locating in S_1 and S_2 to increase $P(d)$ at the expense of $N(d)$.

38.4 RANDOMIZED ALGORITHMS

Dyer and Frieze [DF89] showed that, by applying an idea of Clarkson [Cla86a] to give a *randomized* solution of the multidimensional search in Megiddo's algorithm [Meg84], an algorithm of complexity $O(d^{3d+o(d)}n)$ was possible. Clarkson [Cla88] improved this dramatically. We describe this below, but first outline a simpler algorithm subsequently given by Seidel [Sei91].

Suppose we order the constraints randomly. At stage k, we have solved the linear program subject to constraints $i = 1, \ldots, k-1$. We now wish to add constraint k. If it is satisfied by the current optimum we finish stage k and move to $k+1$. Otherwise, the new constraint is clearly tight at the optimum over constraints $i = 1, \ldots, k-1$. Thus, recursively solve the linear program subject to this equality (i.e., in dimension $d-1$) to get the optimum over constraints $i = 1, \ldots, k$, and move on to $k+1$. Repeat until $k = n$.

The analysis hinges on the following observation. When constraint k is added, the probability it is not satisfied is exactly d/k (assuming, without loss, nondegeneracy). This is because only d constraints are tight at the optimum and this is the probability of writing one of these *last* in a random ordering of $1, 2, \ldots, k$. This leads to an expected time of $O(d!n)$ for (38.1.1). Megiddo observed that Seidel's algorithm is a form of the dual simplex method, with lexicographic choice of pivot row, if implemented in a natural way.

Sharir and Welzl [SW92] modified Seidel's algorithm and improved its analysis to $O(d^3 2^d n)$. They put their algorithm in a general framework of solving "LP-type" problems, and were able to solve smallest enclosing ball or ellipsoid and other problems [Wel91]. Matoušek, Sharir, and Welzl [MSW92] improved the analysis further, essentially obtaining the same bound as for Kalai's "primal simplex" algorithm. The LP-type framework was extended further by Gärtner [Gär92], with a similar time bound.

The basic idea is to choose a random set of r constraints, and solve the linear program subject to these. The solution will violate "few" constraints among the remaining $n - r$, and, moreover, one of these must be tight at x^0. We solve a new linear program subject to the violated constraints and a new random subset of the remainder. We repeat this procedure (aggregating the old violated constraints) until there are no new violated constraints, in which case we have found x^0. Each repetition gives an extra tight constraint for x^0, so we cannot perform more than d iterations.

Clarkson [Cla88] gave a different analysis, but using Seidel's idea we can easily bound the expected number of violated constraints. Imagine all the constraints ordered randomly, our sample consisting of the first r. For $i > r$, let $I_i = 1$ if constraint i is violated, $I_i = 0$ otherwise. Now $\Pr(I_i = 1) = \Pr(I_{r+1} = 1)$ for all $i > r$ by symmetry, and $\Pr(I_{r+1} = 1) = d/(r+1)$ from above. Thus the expected number of violated constraints is

$$\mathbf{E}\,(\sum_{i=r+1}^{n} I_i) = \sum_{i=r+1}^{n} \Pr(I_i = 1) = (n-r)d/(r+1) < nd/r.$$

(In the case of degeneracy, this will be an upper bound by a simple perturbation argument.)

Thus, if $r = \sqrt{n}$, say, there will be at most $d\sqrt{n}$ violated constraints in expectation. Hence, by Markov's inequality, with probability $\frac{1}{2}$ there will be at most $2d\sqrt{n}$ violated constraints in actuality. We must therefore recursively solve about $2(d+1)$ linear programs with at most $(2d^2+1)\sqrt{n}$ constraints. The "small" base cases can be solved by the simplex method in $d^{O(d)}$ time. This can now be applied recursively, as in [Cla88], to give a bound for (38.1.1) of

$$O(d^2 n) + (\log n)^{\log d + 2} d^{O(d)}.$$

Clarkson [Cla95] subsequently modified his algorithm using a different "iterative reweighting" algorithm to solve the $d+1$ small linear programs, obtaining a better bound on the execution time. This uses an idea of Welzl and Littlestone.

Each constraint receives an initial weight of 1. Random samples of total weight $10d^2$ (say) are chosen at each iteration, and solved by the simplex method. If W is the current total weight of all constraints, and W' the weight of the unsatisfied constraints, then $W' \le 2Wd/10d^2 = W/5d$ with probability at least $\frac{1}{2}$ by the discussion above, regarding the weighted constraints as a multiset. We now double the weights of all violated constraints and repeat until there are no violated constraints. This terminates in $O(d \log n)$ by the following argument. After k iterations we have

$$W \le \left(1 + \frac{1}{5d}\right)^k n \le n e^{k/5d},$$

and W^*, the total weight of the d optimal constraints, satisfies $W^* \ge 2^{k/d}$, since at least one is violated at each iteration. Now it is clear that $W^* < W$ only while $k < Cd \ln n$, for some constant C. Applying this to the $d+1$ small linear programs gives overall complexity

$$O(d^2 n + d^4 \sqrt{n} \log n) + d^{O(d)} \log n.$$

This is almost the best time known for fixed-dimensional linear programming, except that Kalai's algorithm can be used to solve the base cases rather than the simplex method. Then we get the improved bound

$$O(d^2 n) + 2^{O(\sqrt{d \log d})} \log n.$$

This is polynomial for $d = O(\log^2 n / \log \log n)$, and is the best bound to date.

38.5 DERANDOMIZED METHODS

Somewhat surprisingly, the randomized methods of Section 38.4 can also lead to the best *deterministic* algorithms for (38.1.1). Matoušek and Chazelle [CM93] produced a derandomized version of Clarkson's algorithm.

The idea, which has wider application, is based on finding (in linear time) *approximations* to the constraint set. If N is a constraint set, then for each $x \in \mathbb{R}^d$ let $V(x, N)$ be the set of constraints violated at x. A set $S \subseteq N$ is an ϵ-approximation to N if, for all x,

$$\left| \frac{|V(x,S)|}{|S|} - \frac{|V(x,N)|}{|N|} \right| < \epsilon.$$

(See also Sections 31.2 and 34.1.) Since $n = |N|$ hyperplanes partition $\mathbb{R}^d$ into only $O(n^d)$ regions, there is essentially only this number of possible cases for x, i.e., only this number of different sets $V(x, N)$. It follows from the work of Vapnik and Chervonenkis that a (d/r)-approximation of size $O(r^2 \log r)$ always exists, since a random subset of this size has the property with nonzero probability. If we can find such an approximation deterministically, then we can use it in Clarkson's algorithm in place of random sampling. If we use a (d/r)-approximation, then, if x^* is the linear programming optimum for the subset S, $|V(x^*, S)| = 0$, so that

$$|V(x^*, N)| < |N|d/r = nd/r,$$

as occurs in expectation in the randomized version. The implementation involves a refinement based on two elegant observations about approximations, which both follow directly from the definition.

(i) An ϵ-approximation of a δ-approximation is an $(\epsilon + \delta)$-approximation of the original set.

(ii) If we partition N into q equal sized subsets $N_1, \ldots, N_q$ and take an (equal sized) ϵ-approximation S_i in each N_i $(i = 1, \ldots, q)$, then $S_1 \cup \ldots \cup S_q$ is an ϵ-approximation of N.

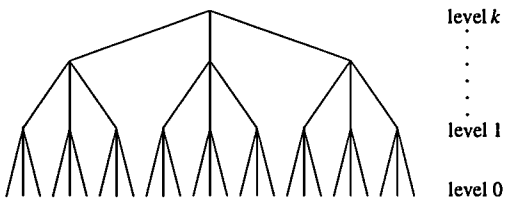


FIGURE 38.5.1
A partition tree of height k, with q = 3.

We then *recursively* partition N into q equal sized subsets, to give a "partition tree" of height k, say, as in Figure 38.5.1 (cf. Section 31.2). The sets at level 0 in the partition tree are "small." We calculate an ϵ_0-approximation in each. We now take the union of these approximations at level 1 and calculate an ϵ_1-approximation of this union. This is an $(\epsilon_0 + \epsilon_1)$-approximation of the whole level 1 set, by the above observations. Continuing up the tree, we obtain an overall $(\sum_{i=0}^{k} \epsilon_i)$-approximation of the entire set. At each stage, the sets on which we have to find the approximations remain "small" if the ϵ_i are suitably chosen. Therefore we can use a relatively inefficient method of finding an approximation. A suitable method is the *method of conditional probabilities* due to Raghavan and Spencer. It is (relatively) straightforward to implement this on a set of size m to run in $O(m^{d+1})$ time. However, since this has to be applied only to small sets (in comparison with n), the total work can be bounded by a linear function of n. Chazelle and Matoušek [CM93] used $q = 2$, and an ϵ_i that corresponds to roughly halving the union at each level $i = 1, \ldots, k$.

The algorithm cannot completely mimic Clarkson's, however, since we can no longer use $r = \sqrt{n}$. Such a large approximation cannot be determined in linear time by the above methods. But much smaller values of r suffice (e.g., $r = 10d^3$) simply to get *linear-time* behavior in the recursive version of Clarkson's algorithm. Using this observation, Chazelle and Matoušek [CM93] obtained a deterministic algorithm with time-complexity $d^{O(d)}n$. This is currently the best time bound known for solving (38.1.1), and remains polynomial for $d = O(\log n / \log\log n)$.

38.6 PARALLEL ALGORITHMS

GLOSSARY

PRAM: Parallel Random Access Machine. (See Section 36.1 for more information on this and the next two terms.)

EREW: Exclusive Read Exclusive Write.

CRCW: Concurrent Read Concurrent Write.

P: The class of polynomial time problems.

NC: The class of problems that have poly-logarithmic parallel time algorithms running a polynomial number of processors.

P-*complete problem:* A problem in P whose membership in NC implies P = NC.

Expander: A graph in which, for every set of nodes, the set of the neighbors of the nodes is relatively large.

Recently, parallel algorithms for (38.1.1) have been of interest in the literature. Here we will consider only PRAM algorithms. (See also Section 36.2.)

The general linear programming problem has long been known to be P-complete, so there is little hope of very fast parallel algorithms. However, the situation is different in the case $d \ll n$, where the problem is in NC if d grows slowly enough.

First, we note that there is a straightforward parallel implementation of Megiddo's algorithm [Meg83] that runs in $O((\log n)^d)$ time on an EREW PRAM.

However, this algorithm is rather inefficient in terms of processor utilization, since at the later stages, when there are few constraints remaining, most processors are idle. However, Deng [Den90] gave an "optimal" $O(n)$ work implementation in the plane running in $O(\log n)$ time on a CRCW PRAM with $O(n/\log n)$ processors. Deng's method does not seem to generalize to higher dimensions.

Alon and Megiddo [AM94] gave a *randomized* parallel version of Clarkson's algorithm which, with high probability, runs in constant time on a CREW PRAM in fixed dimension. Here the "constant" is a function of dimension only, and the probability of failure to meet the time bound is small for $n \gg d$.

Ajtai and Megiddo [AM96] attempted to improve the processor utilization in parallelizing Megiddo's algorithm for general d. They gave an intricate algorithm based on using an expander graph to select more nondisjoint pairs so as to utilize all the processors and obtain more rapid elimination. The resulting algorithm for (38.1.1) runs in $O((\log\log n)^d)$ time, but in a nonuniform model of parallel computation based on Valiant's comparison model. The model, which is stronger than the CRCW PRAM, requires $O(\log\log n)$ time median selection from n numbers using n processors, and employs an $O(\log\log n)$ time scheme for compacting the data after deletions, again based on a nonuniform use of expander graphs. A lower bound of $\Omega(\log n/\log\log n)$ time for median-finding on the CRCW PRAM follows from results of Beame and Hastad. Thus Ajtai and Megiddo's algorithm could not be implemented directly on the CRCW PRAM. Within Ajtai and Megiddo's model there is a lower bound $\Omega(\log\log n)$ for the case $d = 1$ implied by results of Valiant. This extends to the CRCW PRAM, and is the only lower bound known for solving (38.1.1) in this model.

Dyer [Dye95] gave a different parallelization of Megiddo's algorithm, which avoids the use of expanders. The method is based on forming groups of size $r \geq 2$, rather than simple pairs. As constraints are eliminated, the size of the groups is gradually increased to utilize the extra processors. Using this, Dyer [Dye95] establishes an $O(\log n(\log\log n)^{d-1})$ bound in the EREW model. It is easy to show that there is an $\Omega(\log n)$ lower bound for solving (38.1.1) on the EREW PRAM, even with $d = 1$. (See [KR90].) Thus improvements on Dyer's bound for the EREW model can only be made in the $\log\log n$ term. However, there was still an open question in the CRCW model, since exact median-finding and data compaction cannot be performed in time polynomial in $\log\log n$.

Goodrich [Goo93] solved these problems for the CRCW model by giving fast implementations of derandomization techniques similar to those outlined in Section 38.5. However, the randomized algorithm that underlies the method is not a parallelization of Clarkson's algorithm, but is similar to a parallelized version of that of Dyer and Frieze [DF89]. He achieves a work-optimal (i.e., $O(n)$ work) algorithm running in $O(\log\log n)^d$ time on the CRCW PRAM. The methods also imply a work-optimal EREW algorithm, but only with the same time bound as Dyer's. Neither Dyer nor Goodrich is explicit about the dependence on d of the execution time of their algorithms.

Independently of Goodrich's work, Sen [Sen95] has shown how to directly modify Dyer's algorithm to give a work-optimal algorithm with $O((\log\log n)^{d+1})$ execution time in the CRCW model. The "constant" in the running time is shown to be $2^{O(d^2)}$. To achieve this, he uses approximate median-finding and approximate data compaction operations, both of which can be done in time polynomial in $\log\log n$ on the common CRCW PRAM. These additional techniques are, in fact, both examples of derandomized methods and similar to those Goodrich uses for the same

purpose. Note that this places linear programming in NC provided $d = O(\sqrt{\log n})$. This is the best result known, although Goodrich's algorithm may give a better behavior once the "constant" has been explicitly evaluated. We may also observe that the Goodrich/Sen algorithms improve on Deng's result in $\mathbb{R}^2$.

There is still room for some improvements in this area, but there now seems to be a greater need for sharper lower bounds, particularly in the CRCW case.

38.7 RELATED ISSUES

GLOSSARY

Integer programming problem: A linear programming problem with the additional constraint that the solution must be integral.

Linear programming is a problem of interest in its own right, but it is also representative of a class of geometric problems to which similar methods can be applied. Many of the references given below discuss closely related problems, such as smallest enclosing balls and ellipsoids. Some of these applications have been mentioned in passing above, but for other examples see [Wel91] or, in the parallel case, [Sen95].

An important related area is integer programming. Here the size of the numbers cannot be relegated to a secondary consideration. In general this problem is NP-hard, but in fixed dimension is polynomial-time solvable. See [Sch86] for further information. It may be noted that Clarkson's methods are applicable in this situation [Cla95].

We have considered only the solution of a single linear program. However, there are some situations where one might wish to solve a sequence of closely related linear programs. In this case, it may be worth the computational investment of building a data structure to facilitate fast solution of the linear programs. For results of this type see, for example, [Epp90] and [Mat93].

Finally, we have concerned ourselves solely with theoretical issues so far. But, from a practical viewpoint, Clarkson's algorithm at least is certainly applicable in the case $d \ll n$. There is little practical work cited in the references below, but Welzl [Wel91] gives good results using an algorithm based on the LP-type solver of [SW92].

38.8 SOURCES AND RELATED MATERIAL

BOOKS AND SURVEYS

A good general introduction to linear programming may be found in Chvátal's book [Chv83]. A theoretical treatment is given in Schrijver's book [Sch86]. The latter is a very good source of additional references. Karp and Ramachandran [KR90] is a

good source of information on models of parallel computation. See [Mat96] for a recent survey of derandomization techniques for computational geometry.

RELATED CHAPTERS

Chapter 13: Basic properties of convex polytopes
Chapter 17: Polytope skeletons and paths
Chapter 27: Computational convexity
Chapter 36: Parallel algorithms in geometry
Chapter 37: Parametric search
Chapter 39: Mathematical programming
Chapter 52: Computational geometry software

REFERENCES

[AM96] M. Ajtai and N. Megiddo. A deterministic poly($\log\log n$)-time n-processor algorithm for linear programming in fixed dimension. *SIAM J. Comput.*, 25:1171–1195, 1996.

[AM94] N. Alon and N. Megiddo. Parallel linear programming in fixed dimension almost surely in constant time. *J. Assoc. Comput. Mach.*, 41:422–434, 1994.

[BDF+95] A. Broder, M.E. Dyer, A.M. Frieze, P. Raghavan, and E. Upfal. The worst case running time of the randomized simplex algorithm is exponential in the height. *Inform. Process. Lett.*, 56:79–82, 1995.

[Chv83] V. Chvátal. *Linear Programming.* Freeman, New York, 1983.

[Cla86a] K.L. Clarkson. Further applications of random sampling to computational geometry. In *Proc. 18th Annu. ACM Sympos. Theory Comput.*, pages 414–423, 1986.

[Cla86b] K.L. Clarkson. Linear programming in $O(n3^{d^2})$ time. *Inform. Process. Lett.*, 22:21–24, 1986.

[Cla88] K.L. Clarkson. Las Vegas algorithms for linear and integer programming when the dimension is small. In *Proc. 29th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 452–456, 1988.

[Cla95] K.L. Clarkson. Las Vegas algorithms for linear and integer programming when the dimension is small. *J. Assoc. Comput. Mach.*, 42:488–499, 1995. Improved version of [Cla88].

[CM93] B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization in fixed dimension. In *Proc. 4th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 281–289, 1993.

[Den90] X. Deng. An optimal parallel algorithm for linear programming in the plane. *Inform. Process. Lett.*, 35:213–217, 1990.

[DF89] M.E. Dyer and A.M. Frieze. A randomized algorithm for fixed-dimensional linear programming. *Math. Programming*, 44:203–212, 1989.

[Dye83] M.E. Dyer. The complexity of vertex enumeration methods. *Math. Oper. Res.*, 8:381–402, 1983.

[Dye84] M.E. Dyer. Linear time algorithms for two- and three-variable linear programs. *SIAM J. Comput.*, 13:31–45, 1984.

[Dye86] M.E. Dyer. On a multidimensional search problem and its application to the Euclidean one-centre problem. *SIAM J. Comput.*, 15:725–738, 1986.

[Dye92] M.E. Dyer. A class of convex programs with applications to computational geometry. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 9–15, 1992.

[Dye95] M.E. Dyer. A parallel algorithm for linear programming in fixed dimension. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 345–349, 1995.

[Epp90] D. Eppstein. Dynamic three-dimensional linear programming. *ORSA J. Comput.*, 4:360–368, 1990.

[Gär92] B. Gärtner. A subexponential algorithm for abstract optimization problems. In *Proc. 33rd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 464–472, 1992.

[Goo93] M.T. Goodrich. Geometric partitioning made easier, even in parallel. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 73–82, 1993.

[GZ95] B. Gärtner and G.M. Ziegler. Randomized simplex algorithms on Klee-Minty cubes. In *Proc. 35th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 502–510, 1995.

[Kal92] G. Kalai. A subexponential randomized simplex algorithm. In *Proc. 24th Annu. ACM Sympos. Theory Comput.*, pages 475–482, 1992.

[KR90] R. Karp and V. Ramachandran. Parallel algorithms for shared-memory machines. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Vol. A: Algorithms and Complexity*, pages 869–941. Elsevier, Amsterdam, 1990.

[Mat93] J. Matoušek. Linear optimization queries. *J. Algorithms*, 14:432–448, 1993.

[Mat96] J. Matoušek. Derandomization in computational geometry. *J. Algorithms*, 20:545–580, 1996.

[Meg83] N. Megiddo. Linear time algorithms for linear programming in R^3 and related problems. *SIAM J. Comput.*, 12:759–776, 1983.

[Meg84] N. Megiddo. Linear programming in linear time when dimension is fixed. *J. Assoc. Comput. Mach.*, 31:114–127, 1984.

[Meg89] N. Megiddo. On the ball spanned by balls. *Discrete Comput. Geom.*, 4:605–610, 1989.

[MP78] D.E. Muller and F.P. Preparata. Finding the intersection of two convex polyhedra. *Theoret. Comput. Sci.*, 7:217–236, 1978.

[MSW92] J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 1–8, 1992.

[Sch86] A. Schrijver. *Introduction to the Theory of Linear and Integer Programming.* Wiley, Chichester, 1986.

[Sei91] R. Seidel. Low dimensional linear programming and convex hulls made easy. *Discrete Comput. Geom.*, 6:423–434, 1991.

[Sen95] S. Sen. A deterministic poly(log log n) time optimal CRCW PRAM algorithm for linear programming in fixed dimension. Technical Report 95-08, Dept. of Comput. Sci., Univ. of Newcastle, Australia, 1995.

[Sha78] M.I. Shamos. *Computational Geometry.* PhD thesis, Yale Univ., New Haven, 1978.

[SW92] M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In *Proc. 9th Sympos. Theoret. Aspects Comput. Sci.*, volume 577 of *Lecture Notes in Comput. Sci.*, pages 569–579. Springer-Verlag, Berlin, 1992.

[Wel91] E. Welzl. Smallest enclosing disks (balls and ellipsoids). In H. Maurer, editor, *New Results and New Trends in Computer Science*, volume 555 of *Lecture Notes in Comput. Sci.*, pages 359–370. Springer-Verlag, Berlin, 1991.

39 MATHEMATICAL PROGRAMMING

Michael J. Todd

INTRODUCTION

Mathematical programming is concerned with minimizing a real-valued function of several variables, which may be either discrete or continuous, subject to equality and/or inequality constraints on other functions of the variables. Optimality conditions and computational schemes for such problems frequently rely on geometrical properties of the set of feasible solutions or subsidiary geometrical constructions. Here we consider these aspects of general nonlinear optimization problems (Section 39.1), general convex programming (Section 39.2, where we discuss the ellipsoid method and its relatives); linear programming (Section 39.3, where we consider the simplex algorithm and more recent interior-point methods); integer and combinatorial optimization (Section 39.4); and special convex programming problems (Section 39.5). The treatment here focuses mainly on methods involving geometric ideas, especially those for which global complexity estimates are known.

39.1 GENERAL NONLINEAR PROGRAMMING

Consider the problem of choosing x to

$$\begin{array}{lrcll} \text{minimize} & f(x) & & & \\ \text{subject to} & g_i(x) & \leq & 0, & i = 1, \ldots, m, \\ & h_j(x) & = & 0, & j = 1, \ldots, p, \end{array} \qquad \text{(P)}$$

where f and all g_i's and h_j's are smooth functions on real n-dimensional Euclidean space $\mathbb{R}^n$ and x can vary over all $\mathbb{R}^n$. This is a ***general nonlinear programming*** problem.

Research on the general nonlinear programming problem seeks to characterize global or local minimizers by appropriate optimality conditions, and to compute or approximate a local minimizer or stationary point by some iterative method.

GLOSSARY

Objective function: f.

Constraint functions: g_i's and h_j's.

Feasible point: Point in $\mathbb{R}^n$ satisfying all constraints.

Active constraints: All equality constraints and those inequality constraints holding with equality at a given feasible point.

Global minimizer: Feasible point with objective function value at most that of any other feasible point.

Local minimizer: Feasible point with objective function value at most that of any other sufficiently close feasible point.

Optimality conditions: Conditions that are necessary or sufficient, perhaps under regularity restrictions, for a given point to be a local or global minimizer.

Stationary point: Point satisfying optimality conditions.

Lagrangian function: Function $L(x,u,v) := f(x) + u^T g(x) + v^T h(x)$.

39.1.1 OPTIMALITY CONDITIONS

These are based on Taylor approximations of the objective and constraint functions. Let $\bar{x}$ be a feasible point. We seek conditions that are necessary or sufficient for $\bar{x}$ to be a local minimizer. The first-order ***Karush-Kuhn-Tucker conditions***, *necessary* under a regularity condition (such as that the gradients of all constraints active at $\bar{x}$ are linearly independent), involve the Lagrangian function, but because of the presence of inequalities, are more general than the classical Lagrange conditions. They can be stated simply as follows: there exist multipliers $\bar{u}$, $\bar{v}$, such that

$$\begin{array}{rcl} \nabla_x L(\bar{x},\bar{u},\bar{v}) & = & 0, \\ \nabla_u L(\bar{x},\bar{u},\bar{v}) & \leq & 0, \quad \bar{u} \geq 0, \quad \bar{u}^T \nabla_u L(\bar{x},\bar{u},\bar{v}) = 0, \\ \nabla_v L(\bar{x},\bar{u},\bar{v}) & = & 0. \end{array}$$

THEOREM 39.1.1

Let $\bar{x}$ be a feasible point, and assume the regularity condition that the gradients of all constraints active at $\bar{x}$ are linearly independent. Then the Karush-Kuhn-Tucker conditions are necessary *for $\bar{x}$ to be a local minimizer for (P).*

Second-order conditions, involving the Hessian (second derivative matrix) of the Lagrangian, are also important because of the role of curvature in nonlinear optimization. For example:

THEOREM 39.1.2

Suppose that the first-order conditions above hold, and in addition that, for all nonzero directions d with $\nabla h_j(\bar{x})^T d = 0$ for all j, $\nabla g_i(\bar{x})^T d = 0$ for all i with $\bar{u}_i > 0$, and $\nabla g_i(\bar{x})^T d \leq 0$ for all other constraints g_i active at $\bar{x}$, $d^T \nabla^2_{xx} L(\bar{x},\bar{u},\bar{v}) d > 0$. Then $\bar{x}$ is a local minimizer. (Thus these are sufficient *conditions.)*

Note the following special case: For unconstrained problems, $\nabla f(\bar{x}) = 0$ is necessary for $\bar{x}$ to be a local minimizer, while this equality together with $\nabla^2 f(\bar{x})$ positive definite is sufficient.

39.1.2 ALGORITHMS

Methods to approximate stationary points or possibly local minimizers of general smooth nonlinear programming problems are often based on solving a sequence of simpler problems, using the final approximation or solution of the previous problem as a starting point for the new problem. Examples of simpler problems include unconstrained minimization, using barrier, penalty, or (augmented) Lagrangian functions to incorporate the constraints, or a quadratic minimization subject to linear

constraints, where the original nonlinear constraints are linearized, and the Hessian of the quadratic objective approximates that of the Lagrangian of the original problem. (Such ***quadratic programming*** problems can be solved exactly when the objective function is convex, by extensions of methods for linear programming or other algorithms.) If the original problem is unconstrained, and we make a quadratic approximation to the function at each iteration, we recover Newton's method for optimization if the Hessian is exact, and various quasi-Newton methods if approximations are iterated.

Let us describe some typical examples of such algorithms. We state these in simplified form without worrying about important subjects like step size selection, termination criteria, or implementation details. We also omit *globalization* techniques, designed to force convergence to a stationary point or local minimizer from arbitrary starting points (not guaranteeing global minimizers, which is in general much harder!). Subscript k here refers to the iteration number, not a component.

NEWTON'S METHOD FOR UNCONSTRAINED MINIMIZATION

Given iterate x_k, calculate $\nabla f(x_k)$ and $H_k := \nabla^2 f(x_k)$. Stop if H_k is not positive definite. Otherwise, compute the direction d_k as the solution to $H_k d_k = -\nabla f(x_k)$: note that $x_k + d_k$ minimizes the Taylor approximation

$$f(x_k) + \nabla f(x_k)^T (x - x_k) + (1/2)(x - x_k)^T H_k (x - x_k).$$

Let $x_{k+1} := x_k + \alpha_k d_k$ for some step size α_k chosen so that $f(x_{k+1}) < f(x_k)$, and repeat.

BFGS METHOD FOR UNCONSTRAINED MINIMIZATION

This is a very popular quasi-Newton method. Instead of the Hessian matrix being calculated, a positive definite approximation to it is *updated* at each iteration using new information obtained about f. This method is named for Broyden, Fletcher, Goldfarb, and Shanno, who independently developed the update formula below. Initially, choose H_0, say, as some positive multiple of the identity matrix. At the kth iteration, proceed as above but with H_k the updated approximation. The step size α_k is chosen so that $f(x_{k+1}) < f(x_k)$ and so that, with $y_k := \nabla f(x_{k+1}) - \nabla f(x_k)$ and $s_k := x_{k+1} - x_k$, we have $y_k^T s_k > 0$. Then update H_k to

$$H_{k+1} := H_k - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

(this formula maintains positive definiteness) and repeat. Note that we have $H_{k+1} s_k = y_k$, the so-called secant or quasi-Newton equation; clearly, if f is quadratic, its constant Hessian matrix satisfies this equation.

A SEQUENTIAL QUADRATIC PROGRAMMING METHOD FOR CONSTRAINED MINIMIZATION

Given the iterate x_k and estimates of the Lagrange multipliers u_k and v_k, evaluate the gradients of all functions and $\nabla^2_{xx} L(x_k, u_k, v_k)$. Then solve the quadratic programming subproblem:

$$\begin{array}{lrcll} \min_d & \nabla f(x_k)^T d & + & (1/2) d^T \nabla^2_{xx} L(x_k, u_k, v_k) d & \\ & g_i(x_k) & + & \nabla g_i(x_k)^T d & \leq 0, \quad \text{all } i \\ & h_j(x_k) & + & \nabla h_j(x_k)^T d & = 0, \quad \text{all } j \end{array}$$

to get d_k. Let $x_{k+1} := x_k + \alpha_k d_k$ for some step size α_k chosen, for example, so that the penalty function

$$f(x) + \mu \sum_i \max\{g_i(x), 0\} + \nu \sum_j |h_j(x)|$$

is reduced in moving from x_k to x_{k+1}, for suitable positive μ and ν. Replace u_k and v_k by the Lagrange multipliers for the constraints in the quadratic programming problem above, and repeat. There are also quasi-Newton versions of this method.

CONVERGENCE

Some global, local, or rate of convergence results can be established for such methods, for example:

THEOREM 39.1.3

If Newton's method is started sufficiently close to a point x_* satisfying the second-order sufficient conditions to be a local minimizer of f, then the iterates will converge to x_* and the convergence will be quadratic: $\|x_{k+1} - x_*\|/\|x_k - x_*\|^2$ remains bounded.

(A similar result holds for the sequential quadratic programming method using an exact Hessian.) For the quasi-Newton method, it is generally necessary to assume also that H_0 is sufficiently close to $\nabla^2 f(x_*)$, and the convergence is only superlinear: $\|x_{k+1} - x_*\|/\|x_k - x_*\|$ converges to zero. These are local convergence and rate of convergence results. For an example of a global convergence result, consider the unconstrained minimization problem. Then, assuming f is bounded below, if an algorithm of the form above has the angle between each search direction d_k and the direction of the negative gradient $-\nabla f(x_k)$ bounded away from 90°, and the step sizes are chosen appropriately, then $\nabla f(x_k)$ necessarily converges to zero, and so every limit point is a stationary point. However, no bounds on the total computation required for a prescribed precision are known in general (or to be expected lacking convexity). Vavasis [Vav91] describes what complexity results have been obtained for certain special nonconvex problems; for example, minimizing a general quadratic function subject to simple bound constraints is NP-hard, while minimizing such a function subject to lying in a ball is polynomially approximable.

39.2 CONVEX PROGRAMMING

Now we suppose that the functions f and all g_i's are convex, and that all h_j's are linear (affine). Then (P) is called a ***convex programming*** problem; it involves the minimization of a convex function over a convex set.

39.2.1 OPTIMALITY CONDITIONS

If all the functions involved are smooth, then the first-order conditions that are necessary (under a regularity condition) also turn out to be sufficient, not just

for local but for global optimality. In other words, stationary points are global minimizers:

THEOREM 39.2.1

Suppose $\bar{x}$ is feasible for the convex programming problem (P), and that there exist multipliers $\bar{u}$ and $\bar{v}$ such that the Karush-Kuhn-Tucker conditions hold. Then $\bar{x}$ is a global minimizer for (P).

There are also optimality conditions in the nonsmooth case, since convex functions admit subgradients (linear supports) even if they are not differentiable at a point. For a convex function k and a point $\bar{x}$,

$$\partial k(\bar{x}) := \{z \mid k(x) \geq k(\bar{x}) + z^T(x - \bar{x}) \text{ for all } x\}$$

is called the ***subdifferential*** of k at $\bar{x}$, and it is a nonempty compact convex set whose members are called ***subgradients*** of k at $\bar{x}$.

THEOREM 39.2.2

Consider the (modified) Karush-Kuhn-Tucker conditions at a point $\bar{x}$, where the first equation is replaced by

$$0 \in \partial_x L(\bar{x}, \bar{u}, \bar{v}).$$

These conditions are sufficient for $\bar{x}$ to be a global minimizer for (P). In the case that (P) satisfies the regularity condition that there is some feasible point $\hat{x}$ satisfying all inequality constraints strictly, these conditions are also necessary for $\bar{x}$ to be a local minimizer for (P).

In addition, optimality conditions can be stated as saddle-point properties of the Lagrangian. Indeed, $(\bar{x}, \bar{u}, \bar{v})$ satisfies these conditions iff

$$L(x, \bar{u}, \bar{v}) \geq L(\bar{x}, \bar{u}, \bar{v}) \geq L(\bar{x}, u, v)$$

for any $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m_+$, $v \in \mathbb{R}^p$. Whether the functions are smooth or not, local minimizers are always global minimizers. There is also a rich duality theory for convex programming, with many results mirroring those for linear programming. See [Roc70].

39.2.2 ALGORITHMS

As far as algorithms are concerned, in the smooth case one can again employ the general methods discussed above. Slightly stronger results are available about convergence; for example, for the unconstrained minimization of a convex function f, global convergence of the BFGS quasi-Newton method is assured for suitable step size rules, and it is no longer necessary to assume H_0 close to $\nabla^2 f(x_*)$ to obtain superlinear convergence. However, again no global estimates of the work required to attain a certain precision are known for such methods.

LOCALIZER ALGORITHMS

On the other hand, a different class of methods for which such guarantees are available can be applied, even in the nonsmooth case. Various methods are appropriate

in the case of smooth functions or the case of nonsmooth functions when the dimension n is high and the desired accuracy low. Here we'll briefly describe methods for nonsmooth problems where n is small and high accuracy is required; these are based on very geometrical ideas involving *localizers*. While more general problems can be treated, suppose we wish to minimize a convex function f over the cube $G := \{x \in \mathbb{R}^n \mid \|x\|_\infty \leq 1\}$, where the variation of f over G is at most 1 (this is just a normalization condition).

GLOSSARY

ϵ-optimal point: $x \in G$ with $f(x) \leq \inf_G f + \epsilon$.

Localizer: Pair (H, z), $H \subseteq \mathbb{R}^n$, $z \in G$, such that if $x \in G \setminus H$, $f(x) \geq f(z)$.

We mention three such methods here.

METHOD OF CENTRAL SECTIONS (MCS)

This algorithm generates a sequence $\{x_k\}$ of test points and a sequence $\{(Q_k, z_k)\}$ of localizers by the following rules. Choose $Q_0 := G$, z_0 arbitrary, and at the kth iteration, choose x_k as the center of gravity of Q_k and compute $f(x_k)$ and a subgradient g_k of f at x_k. If $g_k = 0$, x_k minimizes f; in this case, stop. Otherwise, $Q_k^+ := \{x \in Q_k \mid g_k^T x \leq g_k^T x_k\}$ contains all minimizers of f over G. Set $Q_{k+1} := Q_k^+$ and let z_{k+1} be whichever of z_k and x_k has the lower function value. It is easy to see by induction that all (Q_k, z_k)'s are localizers. (For $n = 1$, this amounts to just the well-known bisection method.)

The key fact is that a substantial reduction in volume is obtained in successive localizers: $\mathrm{vol}(Q_{k+1}) \leq (1 - 1/e)\mathrm{vol}(Q_k)$. (Here e denotes the base of the natural logarithm.) From this it is not hard to see that an ϵ-optimal point will be found within $O(n \ln \frac{1}{\epsilon})$ iterations. This is optimal from a worst-case viewpoint—no algorithm can ask fewer questions of f (up to a constant factor) and guarantee ϵ-optimality. Unfortunately, no good methods for finding or approximating centers of gravity are known for general n.

ELLIPSOID METHOD (EM)

The (circumscribing) ellipsoid method of Yudin-Nemirovskii and Shor is similar, with the following variations: Q_0 is taken to be the minimum-volume ellipsoid containing G, and Q_{k+1} the minimum-volume ellipsoid containing the semi-ellipsoid Q_k^+. The formulas for updating x_k and Q_k are then trivially implemented, thus removing the drawback of the MCS. However, the volume reduction is much less: $\mathrm{vol}(Q_{k+1}) \approx (1 - [2(n+1)]^{-1})\mathrm{vol}(Q_k)$, and this leads to a complexity bound of $O(n^2[\ln n + \ln \frac{1}{\epsilon}])$ iterations to get an ϵ-optimal point. An iteration of the ellipsoid method is shown in Figure 39.2.1.

METHOD OF INSCRIBED ELLIPSOIDS (MIE)

This final method chooses the localizers Q_k as in the MCS, but at each iteration takes x_k as the center of (an approximation to) the maximum-volume ellipsoid E_k contained in Q_k. It can be shown that $\mathrm{vol}(E_{k+1}) \leq (8/9)\mathrm{vol}(E_k)$, which leads to an $O(n \ln \frac{1}{\epsilon})$-iteration bound. After every $O(n \ln n)$ iterations, the polytope Q_k can be

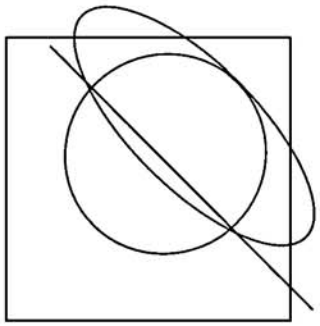


FIGURE 39.2.1
An iteration of the ellipsoid method.

enlarged slightly to one with $O(n)$ facets, so that all Q_l's can be restricted to only $O(n \ln n)$ facets, without changing the complexity. For these polytopes, x_k can be well approximated in $O(n^{3.5+\epsilon})$ arithmetic operations (see [KT93]).

Table 39.2.1 summarizes the complexities of all three methods.

TABLE 39.2.1 Localizer algorithms for convex programming.

ALGORITHM	COMPLEXITY
MCS	$O(n \ln[1/\epsilon])$
EM	$O(n^2[\ln n + \ln(1/\epsilon)])$
MIE	$O(n \ln[1/\epsilon])$

One last remark: In the EM (because possibly $x_k \notin G$) or when (convex) constraints are present, x_k may not be feasible. In this case, g_k is chosen so that all feasible solutions satisfy $g_k^T x \leq g_k^T x_k$, and $z_{k+1} := z_k$.

These methods are not practical for large n, and may not be as efficient for small dimensions as other smooth methods, since they are based on a worst-case perspective. For more efficient methods that have global complexity estimates for certain classes of convex programming problems, see Section 39.5.

39.3 LINEAR PROGRAMMING

Now we discuss the case where all functions defining (P) are linear (affine). By performing simple manipulations, we can express any such problem in standard form, where the constraints take the form of m equations in n nonnegative variables:

$$\begin{array}{lrcl} \min & c^T x & & \\ & Ax & = & b, \\ & x & \geq & 0. \end{array} \qquad \text{(LP)}$$

Here A is $m \times n$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and the variable $x \in \mathbb{R}^n$.

Such problems are of great interest in a wide variety of areas, and their solution comprises a not insignificant fraction of all scientific computation. Problems with

m of the order of 10^3 and n of the order of 10^4 are solved routinely, and larger instances can also be solved without too much difficulty in most cases. (Note the contrast with Chapter 38, where it is assumed that n is small.) In large-scale settings, the sparsity of the matrix A (typically it has at most 5–10 nonzero entries per column) is very important, and numerical methods must exploit sparse matrix technology.

39.3.1 DUALITY

Conditions for a feasible solution to be optimal in (LP) are best stated in terms of another linear programming problem, now involving n inequalities in m variables unrestricted in sign, constructed from the same data. This is the ***dual*** problem (we call (LP) the ***primal***), and is

$$\begin{array}{lll} \max & b^T y & \\ & A^T y & \leq \quad c. \end{array} \tag{LD}$$

It is easy to see that $c^T x \geq b^T y$ for any feasible solutions x for (LP) and y for (LD) (this is called ***weak duality***), so that if equality holds for $\bar{x}$ and $\bar{y}$ they must both be optimal. Indeed, the converse holds: a feasible point $\bar{x}$ is optimal in (LP) if and only if there is a $\bar{y}$ feasible in (LD) with $c^T \bar{x} = b^T \bar{y}$. This is ***strong duality***. There is also a nice geometric way of looking at the optimal dual solution. Let us write $\bar{s}$ for the vector $c - A^T \bar{y}$ (we shall see more of this dual ***slack vector*** later). Then $c^T \bar{x} = b^T \bar{y}$ implies that $\bar{x}^T \bar{s} = 0$, and feasibility in the dual implies that $\bar{s} \geq 0$. Then the dual constraints $A^T \bar{y} + \bar{s} = c$ can be viewed as expressing the gradient of the objective function of the primal problem as a linear combination of the gradients of the constraints that are active at $\bar{x}$ (since $\bar{s}_j$ is nonzero only if $\bar{x}_j$ is zero), where we can take arbitrary multiples of the equality constraint gradients but only nonnegative multiples of the inequality constraint gradients.

Besides its use in proving optimality, the dual problem has important economic interpretations in many instances, and its solution provides crucial sensitivity information about the effects of changes in the data on the solution and optimal value of the primal problem. In addition, it is much exploited in solution techniques for linear programming.

39.3.2 ALGORITHMS

Once again, algorithms for more general problems, i.e., from convex programming, can be applied to these more special problems. Indeed, the first polynomial-time algorithm for linear programming (for the case where all the data are integer-valued) was constructed by Khachiyan based on the ellipsoid method. This result has great theoretical significance (even more for combinatorial optimization, see below), but little practical importance, since the method is very inefficient even for small problems.

Efficient methods for solving (LP) rely on two geometrical ways of looking at its feasible region. This is a polyhedron, typically of dimension $d := n - m$, with at most n facets. The simplex method of Dantzig [Dan63] relies on the combinatorial properties of this polyhedron, in particular on its 1-skeleton, while interior-point methods originating from Karmarkar's projective algorithm (or the earlier method

of Dikin) use a more analytic view relying on a Riemannian metric at points of the interior of the feasible region that reflects its local geometry. (There are also methods based on completely different geometric ideas, for example that of Sharir and Welzl [SW92], but these are not yet used in practice and we shall not discuss them further. In addition, there are special methods appropriate for low-dimensional problems discussed in Chapter 38.)

The differences among the various classes of methods are summarized in Table 39.3.1.

TABLE 39.3.1 Classes of linear programming methods.

METHOD	GEOMETRY	ITERATES	TERMINATION
Ellipsoid	convex	arbitrary points	in the limit
Simplex	combinatorial	vertices	finite
Interior-point	analytic	interior points	in the limit

The "efficiencies" of the methods are shown in Table 39.3.2 (incidentally, this shows the difference between practical and theoretical efficiency).

TABLE 39.3.2 Complexities of linear programming methods.

METHOD	WORST-CASE COMPLEXITY	EXPECTED COMPLEXITY	PRACTICAL
Ellipsoid	polynomial	polynomial	no
Simplex	exponential/super-polynomial?	polynomial	yes
Interior-point	polynomial	polynomial	yes

Articles on the practical efficiency of algorithms for large-scale linear programming problems may be found in [ORJ94].

39.3.3 SIMPLEX METHODS

Given a vertex of the feasible region, a variant of the simplex method proceeds from vertex to vertex along edges while improving the objective function, stopping either at an indication that the objective function is unbounded below or at an optimal solution. (In order to find an initial vertex, the same process is applied to an artificial problem.) The choice of a particular edge to follow is called a pivot rule, and leads to a particular variant. For example, we can choose the edge yielding the maximal decrease in the objective function per unit length (in the Euclidean norm) moved; this is the steepest-edge rule. Various rules can be shown to terminate finitely, even for degenerate problems when the "edge" followed has zero length and merely leads to a different representation of the same vertex (this may happen when the feasible region is not a simple polyhedron). However, no rule is currently known for which the number of steps taken is always bounded by a polynomial in

the dimensions m and n or the input size (total number of bits to represent the data, assumed integer-valued). Indeed, for many rules, examples have shown the worst-case complexity to be exponential, although Kalai [Kal92] has described a randomized rule whose expected number of steps in the worst case is subexponential, though superpolynomial; see Section 38.2. Nevertheless, the simplex method using appropriate pivot rules works very well in practice, requiring only a small number of iterations (possibly $O(m \ln n)$) for typical problems. To explain this gap, some studies have shown that the expected number of steps for certain (deterministic) simplex variants, when applied to a random problem generated by a suitable probabilistic distribution, is polynomial; see [Bor87].

The number of steps taken by a simplex variant starting at an arbitrary vertex is clearly related to the *diameter* of the associated polyhedron of feasible solutions; this is the largest, over all pairs of vertices, of the smallest number of edges required to go from one to the other. The famous ***Hirsch conjecture*** states that this is at most $n-d$ for polytopes (bounded polyhedra) of dimension d with at most n facets. It is known to be true for $d \leq 3$ and $n - d \leq 5$, and for certain classes of polytopes arising in special applications, but the general case remains open; see [KK87]. A nice recent result [Nad89] shows that the conjecture is true for a polytope that is the convex hull of a set of (0,1)-vectors, as arises in combinatorial optimization. Note that knowing that the diameter of a polytope is small does not immediately lead to an efficient simplex method for optimizing a linear function over that polytope.

39.3.4 INTERIOR-POINT METHODS

Algorithms for linear programming of this type generate a sequence of iterates for (LP) that satisfy all the equations and satisfy all the inequalities strictly (we call such points ***strictly feasible***). At such an iterate, one can move in a direction of steepest descent restricted to the affine space $\{x \mid Ax = b\}$, but if the strictly feasible iterate is very close to the boundary of the feasible region, it is possible that only a very short step can be taken.

The first method of this kind, due to Dikin, applied an affine transformation (or ***scaling***) to move the current point to the vector e of ones. A steepest descent step was taken in the scaled space, and the resulting point was transformed back to yield the next iterate. This very simple algorithm performs surprisingly well, but polynomial convergence has not been established. Dikin's paper was not well-known, but was rediscovered soon after Karmarkar's independent work, which used a projective transformation and achieved a polynomial time bound (given in Table 39.3.3). The proof used an ingenious potential function, of the form

$$\phi(x, \zeta) := n \ln(c^T x - \zeta) - \sum_j \ln x_j,$$

with ζ a lower bound on the optimal value of (P), that is closely related to classical barrier functions used in nonlinear programming since the '50's.

Instead of performing the transformations, we can view the search directions in the original space as given by steepest descent directions for a certain function with respect to a Riemannian metric. At the strictly feasible point x, the length of a displacement d_x is defined as $\|d_x\|_x := (d_x^T X^{-2} d_x)^{1/2}$, where X is the diagonal matrix containing the components of x. Thus the metric is defined by the matrix

X^{-2}, which is the Hessian of the *barrier function*

$$F(x) := -\sum_j \ln x_j.$$

The Dikin, or affine-scaling, direction is the steepest direction for the objective function $c^T x$ in the null space of A with respect to this norm, whereas the Karmarkar, or projective-scaling, direction is a similar steepest descent direction for a certain linear combination of $c^T x$ and the barrier $F(x)$. This metric, and in the second case the presence of the barrier, steers the direction away from approaching the boundary of the feasible region too closely prematurely. We will not describe Karmarkar's algorithm in detail, since it is quite complicated and the method has been superseded by those in the next subsection.

39.3.5 PRIMAL-DUAL METHODS

Recent attention has focused on primal-dual methods, which iterate both primal and dual strictly feasible points. It is helpful to write the dual with explicit slack variables as

$$\begin{array}{lrcl} \max & b^T y & & \\ & A^T y & + \; s \; \geq & 0. \end{array} \qquad \text{(LD)}$$

GLOSSARY

Barrier function: Convex function defined on the relative interior of the feasible region, tending to infinity as the boundary is approached.

Interior point: Point satisfying all inequality constraints strictly.

Strictly feasible point: Interior point satisfying all constraints.

e: Vector of ones in $\mathbb{R}^n$.

X (resp. ***S***): Diagonal matrix containing the components of x (resp. s).

Central path: Set of pairs of strictly feasible points x and (y, s) with $XSe = \mu e$ for some $\mu > 0$.

Neighborhoods of central path: Sets of strictly feasible pairs with $XSe - \mu e$ suitably bounded for all iterates.

Primal-dual potential function: $\rho \ln x^T s + F(x) + F(s)$ for $\rho \geq n + \sqrt{n}$, defined for strictly feasible pairs.

Noting that for feasible solutions we have

$$c^T x - b^T y = (A^T y + s)^T x - (Ax)^T y = x^T s \geq 0$$

(a short proof of weak duality), we see that optimality conditions for (LP) and (LD) can be written as

$$\begin{array}{rcl} Ax & = & b \quad (x \geq 0), \\ A^T y + s & = & c \quad (s \geq 0), \\ XSe & = & 0. \end{array} \qquad \text{(OC)}$$

Indeed, the simplex method can be viewed as seeking to satisfy (OC) by maintaining all conditions except the nonnegativity of s at each iteration.

On the other hand, (OC) can be viewed as $m + 2n$ mildly nonlinear equations in $m + 2n$ variables, with nonegativity restrictions as side constraints, and then Newton's method for nonlinear equations seems appropriate. If we start with strictly feasible solutions $\hat{x}$ and $(\hat{y}, \hat{s})$ ($\hat{x} > 0$ and $\hat{s} > 0$) and compute the Newton step for (OC), we can take a partial step to maintain strict feasibility for the next iterates (*damped* Newton step). Indeed, we can take a damped Newton step for a *perturbed* system with the zero right-hand side replaced by $\gamma\hat{\mu}e$, where $\hat{\mu} := \hat{x}^T\hat{s}/n$ is the current duality gap divided by n and $0 \leq \gamma \leq 1$, to encourage the iterates to remain positive while taking a large step.

This primal-dual framework appears to be rather far from the steepest descent view of primal-only interior-point methods. However, the direction for x can be seen to be a steepest descent direction for $c^Tx + \gamma\hat{\mu}F(x)$ with respect to the norm $(d_x^T\hat{X}^{-1}\hat{S}d_x)^{1/2}$, rather than the expected $(d_x^T\hat{X}^{-2}d_x)^{1/2}$. Similarly, the directions for y and s are the steepest descent directions for $-b^Ty+\gamma\hat{\mu}F(s)$ with respect to the norm (on just d_s) $(d_s^T\hat{S}^{-1}\hat{X}d_s)^{1/2}$, rather than the expected $(d_s^T\hat{S}^{-2}d_s)^{1/2}$. Note that these two norms are dual, as is appropriate since x and s lie in dual spaces (the duality gap is $x^Ts = \langle s, x\rangle$), and that they are scalar multiples of the norms given by the Hessians of the barrier functions if $\hat{X}\hat{S}e = \hat{\mu}e$. (They can be viewed as the closest dual norms to the latter.)

Several algorithms are based on taking such damped perturbed Newton steps. We next describe a generic algorithm of this type.

A GENERIC PRIMAL-DUAL INTERIOR-POINT METHOD

Suppose we are given the strictly feasible points x_k and (y_k, s_k) at the kth iteration. Let $\mu_k := x_k^Ts_k/n$ and choose $\gamma \in [0, 1]$. Solve for the directions d_x, d_y, and d_s from

$$\begin{array}{lllll} Ad_x & & & = & 0, \\ & A^Td_y + d_s & & = & 0, \\ S_kd_x & + X_kd_s & = & \gamma\mu_ke - X_kS_ke, \end{array}$$

where X_k and S_k denote the diagonal matrices corresponding to x_k and s_k. Then set $x_{k+1} := x_k + \alpha_P d_x$ and $(y_{k+1}, s_{k+1}) := (y_k, s_k) + \alpha_D(d_y, d_s)$, where the positive step sizes α_P and α_D are chosen so that the new iterates are strictly feasible.

A practical version of this method (which takes long steps to try to converge fast, but which lacks a polynomial bound) might choose $\gamma = 1/n$ and α_P and α_D as .999 of the maximum values that would maintain primal and dual feasibility, respectively.

Most theoretically attractive primal-dual methods try to stay close to the *central path*. For each $\mu > 0$, there is a unique pair x and (y, s) of strictly feasible solutions with $XSe = \mu e$, and the set of all these forms the central path, which leads (as $\mu \to 0$) to the set of optimal solutions. Some methods maintain $\|XSe/\mu - e\| \leq \beta$ with either the ℓ_2- or ℓ_∞-norm, and others only require $XSe \geq (1-\beta)(x^Ts/n)e$, for some $0 < \beta < 1$, for all iterates. When the ℓ_2-neighborhood is used, we get a close path-following method, whereas the other restrictions yield loose path-following methods. For example, one can choose $\alpha_P = \alpha_D = 1$ and $\gamma \in [0, 1]$ as small as possible to maintain $\|XSe/\mu - e\|_2 \leq 1/4$ in the generic algorithm above. (One can

show that $\gamma \leq 1 - 1/(4\sqrt{n})$ with this choice.) The first close path-following method was due to Renegar and operated in the dual space alone; this was also the first method with the improved complexity of $O(\sqrt{n} \ln \frac{1}{\epsilon})$ steps to attain ϵ-optimality. Also, primal-dual methods can be based on a primal-dual potential function and require no neighborhood. All the methods described in this subsection, with the exception of the affine-scaling method and the practical primal-dual algorithm outlined above, are polynomial. However, the methods with the better complexity bounds (see Table 39.3.3) seem not to be as useful practically, as they tend to force short steps.

INFEASIBLE-INTERIOR-POINT METHODS

Finally, as with the simplex method, initial feasible (here, strictly feasible) solutions are rarely known. However, a damped perturbed Newton step can also be taken from iterates $\hat{x}$ and $(\hat{y}, \hat{s})$ with $x > 0$ and $\hat{s} > 0$ (*infeasible interior points*) even if the equations in (LP) and (LD) are violated. These infeasible-interior-point methods strive for feasibility and optimality at the same time, and are the basis for most interior-point codes. Polynomial bounds have recently been obtained (see below). Another approach [YTM94] applies a "feasible" method to an artificial homogeneous self-dual problem and either generates optimal solutions or proves primal or dual infeasibility in the limit.

COMPLEXITY OF INTERIOR-POINT METHODS

The types of algorithms and their complexities to obtain ϵ-optimality given suitable starting points are given in Table 39.3.3. Note that all except the last two assume that a feasible solution is at hand.

TABLE 39.3.3 Complexities of interior-point methods.

ALGORITHM		COMPLEXITY
Primal	affine-scaling	—
	projective-scaling	$O\big(n \ln(1/\epsilon)\big)$
Primal-dual	close path-following	$O\big(\sqrt{n} \ln(1/\epsilon)\big)$
	loose path-following	$O\big(n \ln(1/\epsilon)\big)$
	potential-reduction	$O\big(\sqrt{n} \ln(1/\epsilon)\big)$
	infeasible-interior-point	$O\big(n \ln(1/\epsilon)\big)$
	homogeneous self-dual	$O\big(\sqrt{n} \ln(1/\epsilon)\big)$

All these methods require $O(n^3)$ arithmetical operations at each iteration (although an acceleration trick can reduce this to $O(n^{2.5})$ operations on average for some methods), assuming dense linear algebra is used. To get the complexity of solving exactly a linear programming problem with integer data of length L, replace $(1/\epsilon)$ in Table 39.3.3 by L; we then get polynomial complexity. Roughly, from an ϵ-optimal solution with $\epsilon = 2^{-O(L)}$ we can obtain an exact solution.

39.4 INTEGER AND COMBINATORIAL OPTIMIZATION

Now suppose we wish to optimize a linear function subject to linear constraints and the requirement that the variables be integer. Such a problem is called an integer (linear) programming problem. For simplicity, we assume that all variables must be 0 or 1. Then the problem can be formulated as:

$$\begin{array}{lrcll} \max & & c^T x & & \\ & & Ax & \leq & b, \\ & 0 \leq & x & \leq & e, \\ & & x & & \text{integer}. \end{array} \qquad \text{(IP)}$$

(Recall that e denotes the vector of ones.) Clearly, such problems (sometimes with only some of the variables required to be integer) arise from applications like those leading to linear programming instances—for example, production of certain items (e.g., aircraft carriers) is essentially discrete. But it is important to realize the modeling possibilities of (0,1) variables: they can be used to represent either-or situations, such as whether to build a new factory, invest in a new product, etc., or to model such nonconvexities as set-up costs and minimum batch sizes.

There are also inherently combinatorial problems that can be represented in the form (IP). Consider for example the notorious ***traveling salesman problem*** [LLKS85], which arises in routing and sequencing applications. Here it is desired to visit each of n cities exactly once, starting and finishing at the same city, at minimum cost. By introducing variables x_{ij}, $1 \leq i < j \leq n$, equal to 1 if the salesman goes directly from city i to city j or vice versa, we can model the problem as minimizing a linear function over a finite (but large!) set of (0,1)-vectors of length $n(n-1)/2$. It is not hard to see that this can be written in the form (IP) by introducing appropriate constraints. Indeed, this can be done in several ways.

39.4.1 OPTIMALITY CONDITIONS AND DUALITY

Solving integer programming problems is NP-hard in general, although it is polynomial when the dimension is fixed and for certain special problems (see [Sch86, GLS88]). One reason for the difficulty is that it is far from trivial to check whether a given feasible solution is optimal. Very often, a heuristic method or routine within an algorithm will produce a very good or optimal solution quickly, but proving that it is (near-)optimal is very time-consuming.

The main tool for establishing the quality of a feasible solution for (IP) is linear programming. Note that the optimal value of (IP) is bounded above by that of its ***linear programming relaxation***

$$\begin{array}{lrcll} \max & & c^T x & & \\ & & Ax & \leq & b, \\ & 0 \leq & x & \leq & e. \end{array} \qquad \text{(LR)}$$

There are some problems (mostly network-flow related) for which the linear programming relaxation has only integer vertices. Thus solving (LR) by, say, the simplex method will solve (IP). If this is not the case, then the optimal value of (LR) (or of its linear programming dual) provides information about the quality of

a given feasible solution of (IP), and its optimal solution may help to locate a good integer solution. In any case, it is clear that a "tight" formulation (so that (LR) is "closer" to (IP), and the *integrality gap*—the gap between their optimal values—is smaller) will be very helpful in (approximately) solving (IP).

There are also specialized duals for integer programming involving subadditive functions, but they do not seem to be as useful computationally.

39.4.2 ALGORITHMS

Certain integer or combinatorial optimization problems (for example, network flow problems, see [NKT89]) can be solved very efficiently. This can be done either using the simplex method (as indicated above, for some such problems it suffices to solve the linear programming relaxation, and in some cases a polynomial bound on the number of iterations has been proved) or specialized combinatorial methods (e.g., for certain graph, network, and matroid problems), but these usually have little geometric content. For harder problems two general approaches are possible.

BRANCH-AND-BOUND

The first is an implicit enumeration scheme. Suppose we solve the linear programming relaxation (LR). If the solution is integer, we are done; otherwise, we obtain a bound on the optimal value, and by choosing a variable whose optimal value is fractional, we construct two other problems, in one of which the variable is restricted to be 0 and in the other, 1. If we continued branching in this way, we would eventually reach a tree with 2^n leaves, with each corresponding to a particular (0,1) assignment for all the variables. Two things may allow us to construct only a very much smaller enumeration tree. First, if the optimal solution for the relaxation at some node (corresponding to a partial assignment of the variables) is integer, we need not perform any more branching from this node. If this solution is the best seen so far, we record it. Second, if the linear programming relaxation at a node is either infeasible or has optimal value below that of some known integer solution (obtained by a heuristic or from examination of other parts of the tree), the node need not be considered further. If neither of these cases holds, we choose a fractional variable and branch as above, thus creating two new nodes. Since we keep track of the best integer solution found, when all nodes have been considered, the current best solution solves (IP), or, if there is none, (IP) is infeasible. Note that the efficiency of this technique depends on the tightness of the formulation, since this helps both to generate integer solutions to linear programming relaxations and to give good bounds allowing nodes to be rejected as above.

CUTTING-PLANE METHODS

The other approach tries to generate ever-tighter linear programming relaxations. Note that if we could optimize the objective function over the convex hull C of the finitely many feasible solutions of (IP), we would obtain the optimal solution directly. Since this convex hull is a polytope (such a polytope is called a $(0,1)$-*polytope*), it can be expressed as the solution set to a set of linear inequalities, so that

there is *some* linear programming relaxation that allows the integer programming problem to be solved using linear programming methods. The problem is that we do not know all of these inequalities, and even if we could describe them completely (as we can, say, for the matching problem in a nonbipartite graph) there might be exponentially many of them. Thus we would like to generate them "on the fly."

One algorithm that does not require all the inequalities to be available explicitly is the ellipsoid method. Indeed, there is a precise sense in which, if we can determine in polynomial time whether a given point is within the convex hull C, and, if not, produce a separating hyperplane, then we can optimize over C in polynomial time using the ellipsoid method, and vice versa; see [GLS88]. Of course, this is not a practical method, but it has significant theoretical consequences in combinatorial optimization (showing that some problems are polynomially solvable, and conversely, that others are NP-hard), and gives a strong indication that similar constraint-generation methods using efficient linear programming methods can be practically useful for many problems. We need to be able to re-optimize a linear programming problem easily after slight modifications (addition of constraints, fixing of variables), and for this simplex methods seem preferable at the present time to interior-point algorithms. We also need a way to generate a "good" set of constraints to add when we discover that our current linear programming relaxation is not tight enough. These constraints should be valid for all feasible solutions to (IP), but violated by the optimal solution of the current relaxation, so they are called *cutting planes.*

This approach to the study of combinatorial problems is called

POLYHEDRAL COMBINATORICS

Given a collection of subsets of some finite ground set (e.g., each subset could be the set of edges of a Hamiltonian circuit of a given graph), one can consider the corresponding collection of their (0,1) incidence vectors, and the convex hull C of these vectors. This is a (0,1)-polytope. Questions about the combinatorial system can then be reduced to questions about the resulting polytope; in particular, optimizing over the set of subsets often becomes a linear programming problem over C. It is therefore of interest to obtain complete or partial descriptions of the linear inequalities defining C. These can then be used in developing special algorithms or within the context of the cutting-plane methods above. Thus much recent research has been devoted to finding deep valid inequalities or, if possible, facets of C for practically interesting problems, and also to developing separation techniques to identify a member of a class of facets that is violated by a given point. A very interesting recent result testifies to the complexity of the convex hull of feasible solutions for at least one class of hard problems: Billera and Sarangarajan [BS96] show that *any* (0,1)-polytope is a traveling salesman polytope. Another indication of the complexity of such facetal descriptions is the following. For the case of 8 cities, the traveling salesman polytope has been completely described [CJR91]—it has 194,187 facets! No such description is known for the case of 9 cities. Recall that (0,1)-polytopes have small diameter—they satisfy the Hirsch conjecture. Unfortunately this is of little comfort when we cannot give a complete facetal description of them; moreover, they are often very degenerate, so the simplex method might take many pivot steps going nowhere. Further discussion and references on polyhedral combinatorics can be found in Chapter 7.

REMARKS

The two techniques above can be combined, so that cutting planes are added as long as one can be found, and then enumeration resorted to if necessary. If possible, the inequalities found at one node of the tree should also be valid for other nodes in the tree, so we always have tight relaxations. Using these ideas, very large combinatorial and integer optimization problems have been solved. We note that recently a traveling salesman problem with 7397 cities was solved to optimality.

39.5 SPECIAL CONVEX PROGRAMMING PROBLEMS

In Section 39.2, we described the ellipsoid method and its relatives, but noted that they are not practical for large (or even medium-sized) convex programming problems. On the other hand, Section 39.3 gave methods that are efficient even for very large linear programming problems. Here we address the possibility of solving efficiently large convex optimization problems falling into nice classes, using methods with global complexity estimates.

The first such class is that of (convex) quadratic programming problems. As we noted in Section 39.1, these arise as subproblems in methods for general smooth nonlinear programming; they are also important in their own right, with applications in portfolio analysis and constrained data-fitting. The optimality conditions for such problems are very similar to those for linear programming: another linear term is added to one of the sets of linear equations in (OC). Thus it is not too surprising that extensions of linear programming methods, of both the simplex and interior-point persuasions, have been devised for quadratic programming. The complexity of the latter kinds of algorithm is the same as for linear programming. There are also direct methods for quadratic programming, described in [Fle87, GMW81].

39.5.1 INTERIOR-POINT METHODS FOR NONLINEAR PROGRAMMING

There has recently been great interest in extending interior-point methods to certain classes of convex programming problems, maintaining a polynomial complexity bound. The most extensive work has been done by Nesterov and Nemirovskii, and appears in their book [NN94].

Any convex programming problem can be rewritten, by adding one or two variables, in the form of either

$$\min\{c^T x \mid x \in G\}$$

or

$$\min\{c^T x \mid Ax = b,\, x \in K\},$$

where G and K are closed convex sets in $\mathbb{R}^n$ with nonempty interiors, and K is a cone. The second formulation, said to be in conical form, is clearly closely related to the standard form linear programming problem (LP), and allows a nice statement of the dual as

$$\max\{b^T y \mid A^T y + s = c,\, s \in K^*\},$$

where K^* is the dual cone $\{s \in \mathbb{R}^n \mid x^T s \geq 0 \text{ for all } x \in K\}$. Weak duality is immediate, while strong duality holds if, for example, there are feasible solutions to both problems with x and s in the interiors of their respective cones.

GLOSSARY

Self-concordant barrier: Barrier function satisfying certain smoothness conditions allowing efficient interior-point methods.

Self-scaled barrier: Self-concordant barrier satisfying further conditions allowing greater freedom in efficient methods.

The view of interior-point methods given in Section 39.3 leads us to consider steepest descent steps for a strictly feasible point (in int G or $\{x \in \text{int } K \mid Ax = b\}$) with respect to the norm defined by the Hessian of a barrier function for the set G or K. Nesterov and Nemirovskii devise path-following and potential-reduction methods with polynomial complexity for the problems above as long as a barrier function F for G or K can be found satisfying certain key properties. These properties, roughly convexity and Lipschitz continuity of F and its second derivative with respect to the norm defined by the second derivative itself, define the set of *self-concordant barriers*. An attractive feature of these functions is that Newton's method performs well (in a precise sense) when applied to their minimization, not just locally but also globally. One of the Lipschitz constants, the *parameter* ν of the barrier, takes the place of the number of linear inequalities n in linear programming in complexity bounds. Thus the number of iterations of their algorithms to attain ϵ-optimality is $O(\nu \ln \frac{1}{\epsilon})$ or $O(\sqrt{\nu} \ln \frac{1}{\epsilon})$.

To develop symmetric primal-dual methods like those used in interior-point codes for LP, we need to consider the problem in conical form and require a further condition on the barrier: it should be *self-scaled* [NT97]. We will not give the precise definition here, but we note that one of its consequences is that, for every $x \in \text{int } K$ and $s \in \text{int } K^*$, there is a unique $z \in \text{int } K$ with $F''(z)x = s$. Then the generic primal-dual interior-point method is exactly as in the linear programming case, except that the last equation defining the directions becomes

$$F''(z_k)d_x + d_s = -s_k - \gamma \mu_k F'(x_k),$$

where $F''(z_k)x_k = s_k$.

Barriers of these types have strong geometric consequences. For example, the ball of radius 1 centered at a strictly feasible point and defined by the norm based on the Hessian of a self-concordant barrier at that point lies completely within G or K. Moreover, if such a barrier for G has a minimizer x^* (the *analytic center* of G), then G not only contains the ball of radius 1 centered at x^* with respect to this norm, but also is contained in the ball of radius $1 + 3\nu$, where ν is the parameter of the barrier. For self-scaled barriers, we can find even larger inscribed sets, corresponding to ℓ_∞-balls. The fact that G or K can be thus "well-fitted" by simple convex sets gives an indication of the reason that efficient algorithms can be found to optimize over them.

Because of the general theory it is obviously desirable to construct self-concordant or self-scaled barriers for convex sets and cones that can be used to model important optimization problems. Here is a list of some special cases:

1. Quadratic constraints: Let $g_i, i = 1, \ldots, m$, be convex quadratic functions (note that these include linear functions). Then

$$F(x) := -\sum_{i=1}^{m} \ln[-g_i(x)]$$

is a self-concordant barrier with parameter m for

$$G := \{x \mid g_i(x) \leq 0, i = 1, \ldots, m\}.$$

2. Second-order, or Lorentz, or "ice-cream" cone: The function

$$F(x) := -\ln(x_0^2 - \sum_{j=1}^{n} x_j^2)$$

is a self-concordant (and self-scaled) barrier with parameter 2 for the cone

$$K := \{x \in \mathbb{R}^{n+1} \mid x_0 \geq (\sum_{j=1}^{n} x_j^2)^{1/2}\}.$$

3. Symmetric positive semidefinite matrices: The function

$$F(X) := -\ln \det X$$

is a self-concordant (and self-scaled) barrier with parameter n for the cone K of symmetric positive semidefinite matrices of order n.

This last example is particularly interesting, because several important applications (including obtaining bounds in hard combinatorial optimization or control theory problems, and eigenvalue optimization) can be modeled as optimizing a linear function over the cone of symmetric positive semidefinite matrices, subject to linear constraints.

39.6 SOURCES AND RELATED MATERIAL

BOOKS AND SURVEYS

Besides the references cited above, the reader can consult the following sources for more background and further citations of the literature. In general, the chapters in [NKT89] provide state-of-the-art surveys of various fields of mathematical programming as of 1989. For nonlinear programming, see also [GMW81, Fle87, Vav91]. The book [NY83] contains almost all you want to know about localizer algorithms and the (informational) complexity of convex programming, although it predates the MIE (discussed in [NN94]); see also [BGT81]. For linear programming, the classic texts are [Dan63, Chv83, Sch86]; more recent surveys on path-following methods and potential-reduction algorithms are [Gon92, Tod95], while [Tse94] gives an overview and references in infeasible-interior-point methods. For combinatorial optimization, consult the chapters on optimization, convex polytopes, and polyhedral

combinatorics in [GGL95]. The application of interior-point methods to convex programming is discussed in [dH93, NN94].

RELATED CHAPTERS

Chapter 7: Lattice points and lattice polytopes
Chapter 13: Basic properties of convex polytopes
Chapter 17: Polytope skeletons and paths
Chapter 27: Computational convexity
Chapter 38: Linear programming in low dimensions

REFERENCES

[BGT81] R.G. Bland, D. Goldfarb, and M.J. Todd. The ellipsoid method: A survey. *Oper. Res.*, 29:1039–1091, 1981.

[Bor87] K.H. Borgwardt. *The Simplex Method, a Probabilistic Analysis.* Springer-Verlag, Berlin, 1987.

[BS96] L.J. Billera and A. Sarangarajan. All 0-1 polytopes are traveling salesman polytopes. *Combinatorica*, 16:175–188, 1996.

[Chv83] V. Chvátal. *Linear Programming.* Freeman, San Francisco, 1983.

[CJR91] T. Christof, M. Junger, and G. Reinelt. A complete description of the traveling salesman polytope on 8 nodes. *Oper. Res. Lett.*, 10:497–500, 1991.

[Dan63] G.B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, 1963.

[dH93] D. den Hertog. *Interior Point Approach to Linear, Quadratic and Convex Programming.* Kluwer, Dordrecht, 1993.

[Fle87] R. Fletcher. *Practical Methods of Optimization.* Wiley, New York, 1987.

[GGL95] R.L. Graham, M. Grötschel, and L. Lovász, editors. *Handbook of Combinatorics.* North-Holland, Amsterdam, 1995.

[GLS88] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, Berlin, 1988.

[GMW81] P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization.* Academic Press, New York, 1981.

[Gon92] C.C. Gonzaga. Path following methods for linear programming. *SIAM Rev.*, 34:167–227, 1992.

[Kal92] G. Kalai. A subexponential randomized simplex algorithm. In *Proc. 24th Annu. ACM Sympos. Theory Comput.*, pages 475–482, 1992.

[KK87] V. Klee and P. Kleinschmidt. The d-step conjecture and its relatives. *Math. Oper. Res.*, 12:718–755, 1987.

[KT93] L.G. Khachiyan and M.J. Todd. On the complexity of approximating the maximal inscribed ellipsoid for a polytope. *Math. Programming*, 61:137–159, 1993.

[LLKS85] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization.* Wiley, New York, 1985.

[Nad89] D. Naddef. The Hirsch conjecture is true for (0,1)-polytopes. *Math. Programming*, 45:109–110, 1989.

[NKT89] G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd. *Handbooks in Operations Research and Management Science, Vol. 1: Optimization.* North-Holland, Amsterdam, 1989.

[NN94] Yu.E. Nesterov and A.S. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming: Theory and Algorithms.* SIAM, Philadelphia, 1994.

[NT97] Yu.E. Nesterov and M.J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Math. Oper. Res.*, 77:1–42, 1997.

[NY83] A.S. Nemirovskii and D.B. Yudin. *Problem Complexity and Method Efficiency in Optimization.* Wiley, New York, 1983.

[ORJ94] *ORSA J. Comput.*, 6:1–34, 1994.

[Roc70] R.T. Rockafellar. *Convex Analysis.* Princeton University Press, 1970.

[Sch86] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley, Chichester, 1986.

[SW92] M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In *Proc. 9th Annu. Sympos. Theoret. Aspects Comput. Sci.*, pages 570–579, 1992.

[Tod95] M.J. Todd. Potential-reduction methods in mathematical programming. Tech. Report 1112, School of Operations Research and Industrial Engineering, Cornell Univ., Ithaca, 1995.

[Tse94] P. Tseng. Simplified analysis of an $O(nL)$-iteration infeasible predictor-corrector path-following method for monotone LCP. Tech. Report, Dept. of Mathematics, Univ. of Washington, Seattle, 1994.

[Vav91] S.A. Vavasis. *Nonlinear Optimization: Complexity Issues.* Oxford University Press, New York, 1991.

[YTM94] Y. Ye, M.J. Todd, and S. Mizuno. An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm. *Math. Oper. Res.*, 19:53–67, 1994.

40 ALGORITHMIC MOTION PLANNING

Micha Sharir

INTRODUCTION

Motion planning is a fundamental problem in robotics. It comes in a variety of forms, but the simplest version is as follows. We are given a robot system B, which may consist of several rigid objects attached to each other through various joints, hinges, and links, or moving independently, and a two-dimensional or three-dimensional environment V cluttered with obstacles. We assume that the shape and location of the obstacles and the shape of B are known to the planning system. Given an initial placement Z_1 and a final placement Z_2 of B, we wish to determine whether there exists a collision-avoiding motion of B from Z_1 to Z_2, and, if so, to plan such a motion. In this simplified and purely geometric setup, we ignore issues such as incomplete information, nonholonomic constraints, control issues related to inaccuracies in sensing and motion, nonstationary obstacles, optimality of the planned motion, and so on.

Since the early 1980's, motion planning has been an intensive area of study in robotics and computational geometry. In this chapter we will focus on *algorithmic motion planning*, emphasizing theoretical algorithmic analysis of the problem and seeking worst-case asymptotic bounds, and only mention briefly practical heuristic approaches to the problem. The majority of this chapter is devoted to the simplified version of motion planning, as stated above. Section 40.1 presents general techniques and lower bounds. Section 40.2 considers efficient solutions to a variety of specific moving systems with a small number of degrees of freedom. These efficient solutions exploit various sophisticated methods in computational and combinatorial geometry related to arrangements of curves and surfaces (Chapter 21). Section 40.3 then briefly discusses various extensions of the motion planning problem, incorporating uncertainty, moving obstacles, etc. We conclude in Section 40.4 with a brief review of Davenport-Schinzel sequences, a combinatorial structure that plays an important role in many motion planning algorithms.

40.1 GENERAL TECHNIQUES AND LOWER BOUNDS

GLOSSARY

Robot B: A mechanical system consisting of one or more rigid bodies, possibly connected by various joints and hinges.

Physical space: The two- or three-dimensional environment in which the robot moves.

Placement: The portion of physical space occupied by the robot at some instant.

Degrees of freedom k: The number of real parameters that determine the robot B's placements. Each placement can be represented as a point in $\mathbb{R}^k$.

Free placement: A placement at which the robot is disjoint from the obstacles.

Semifree placement: A placement at which the robot does not meet the interior of any obstacle (but may be in contact with some obstacles).

Configuration space $\mathcal{C}$: A portion of k-space (where k is the number of degrees of freedom of B) that represents all possible robot placements; the coordinates of any point in this space specify the corresponding placement.

Expanded obstacle / C-space obstacle / forbidden region: For an obstacle O, this is the portion O^* of configuration space consisting of placements at which the robot intersects (collides with) O.

Free configuration space $\mathcal{F}$: The subset of configuration space consisting of free placements of the robot: $\mathcal{F} = \mathcal{C} \setminus \bigcup_O O^*$. (In the literature, this usually also includes semifree placements.)

Contact surface: For an obstacle feature a (corner, edge, face, etc.) and for a feature b of the robot, this is the locus in $\mathcal{C}$ of placements at which a and b are in contact with each other. In most applications, these surfaces are semialgebraic sets of constant description complexity (see definitions below).

Collision-free motion of B: A path contained in $\mathcal{F}$. Any two placements of B that can be reached from each other via a collision-free path must lie in the same (arcwise-)connected component of $\mathcal{F}$.

Arrangement $\mathcal{A}(\Sigma)$: The decomposition of k-space into cells of various dimensions, induced by a collection Σ of surfaces in $\mathbb{R}^k$. Each cell is a maximal connected portion of the intersection of some fixed subcollection of surfaces that does not meet any other surface. See Chapter 21. Since a collision-free motion should not cross any contact surface, $\mathcal{F}$ is the union of some of the cells of $\mathcal{A}(\Sigma)$, where Σ is the collection of contact surfaces.

Semialgebraic set: A subset of $\mathbb{R}^k$ defined by a Boolean combination of polynomial equalities and inequalities in the k coordinates. See Section 29.2.

Constant description complexity: Said of a semialgebraic set if it is defined by a constant number of polynomial equalities and inequalities of constant maximum degree (where the number of variables is also assumed to be constant).

Example. Let B be a rigid polygon with k edges, moving in a planar polygonal environment V with n edges. The system has three degrees of freedom, (x, y, θ), where (x, y) are the coordinates of some reference point on B, and θ is the orientation of B. Each contact surface is the locus of placements where some vertex of B touches some edge of V, or some edge of B touches some vertex of V. There are $2kn$ contact surfaces, and if we replace θ by $\tan \frac{\theta}{2}$, then each contact surface becomes a portion of some algebraic surface of degree at most 4, bounded by a constant number of algebraic arcs, each of degree at most 2.

GLOSSARY

Cylindrical algebraic decomposition of $\mathcal{F}$: A recursive decomposition of $\mathcal{F}$ into cylindrical-like cells originally proposed by Collins. Over each cell of the decomposition, each of the polynomials involved in the definition of $\mathcal{F}$ has a fixed sign (positive, negative, or zero), implying that $\mathcal{F}$ is the union of some of the cells of this decomposition. See Section 29.5 for further details.

Connectivity graph: A graph whose nodes are the (free) cells of a decomposition of $\mathcal{F}$ and whose arcs connect pairs of adjacent cells.

Roadmap $\mathcal{R}$: A network of 1-dimensional curves within $\mathcal{F}$, having the properties that (i) it *preserves the connectivity* of $\mathcal{F}$, in the sense that the portion of $\mathcal{R}$ within each connected component of $\mathcal{F}$ is (nonempty and) connected; and (ii) it is ***reachable***, in the sense that there is a simple procedure to move from any free placement of the robot to a placement on $\mathcal{R}$; we denote the mapping resulting from this procedure by $\phi_{\mathcal{R}}$.

Retraction of $\mathcal{F}$ onto $\mathcal{R}$: A continuous mapping of $\mathcal{F}$ onto $\mathcal{R}$ that is the identity on $\mathcal{R}$. The roadmap mapping $\phi_{\mathcal{R}}$ is usually a retraction. When this is the case, we note that for any path ψ within $\mathcal{F}$, represented as a continuous mapping $\psi : [0,1] \mapsto \mathcal{F}\phi_{\mathcal{R}} \circ \psi$ is a path within $\mathcal{R}$, and, concatenating to it the motions from $\psi(0)$ and $\psi(1)$ to $\mathcal{R}$, we see that there is a collision-free motion of B between two placements Z_1, Z_2 iff there is a path within $\mathcal{R}$ between $\phi_{\mathcal{R}}(Z_1)$ and $\phi_{\mathcal{R}}(Z_2)$.

Silhouette: The set of critical points of a mapping; see Section 29.6.

CELL DECOMPOSITION

$\mathcal{F}$ is a semialgebraic set in $\mathbb{R}^k$. Applying Collins's cylindrical algebraic decomposition results in a collection of cells whose total complexity is $O((nd)^{3^k})$, where d is the maximum algebraic degree of the polynomials defining the contact surfaces; the decomposition can be constructed within a similar time bound. If the coordinate axes are generic, then we can also compute all pairs of cells of $\mathcal{F}$ that are ***adjacent*** to each other (i.e., cells whose closures (within $\mathcal{F}$) overlap), and store this information in the form of a connectivity graph. It is then easy to search for a collision-free path through this graph, if one exists, between the (cell containing the) initial robot placement and the (cell containing the) final placement. This leads to a doubly-exponential general solution for the motion planning problem:

THEOREM 40.1.1 *Cylindrical Cell Decomposition* [SS83]

Any motion planning problem, with k degrees of freedom, for which the contact surfaces are defined by a total of n polynomials of maximum degree d, can be solved by Collins's cylindrical algebraic decomposition, in randomized expected time $O((nd)^{3^k})$.

(The randomization is needed only to choose a generic direction for the coordinate axes.)

A more recent and improved solution is given in [Can87, BPR96] based on the notion of a *roadmap* $\mathcal{R}$, a network of 1-dimensional curves within (the closure of) $\mathcal{F}$, having properties defined in the glossary above. Once such a roadmap $\mathcal{R}$ has been constructed, any motion planning instance reduces to path searching within $\mathcal{R}$, which is easy to do. $\mathcal{R}$ is constructed recursively, as follows. One projects $\mathcal{F}$ onto some generic 2-plane, and computes the silhouette of $\mathcal{F}$ under this projection. Next, the critical values of the projection of the silhouette on some line are found, and a roadmap is constructed recursively within each slice of $\mathcal{F}$ at each of these critical values. The resulting "sub-roadmaps" are then merged with the silhouette, to obtain the desired $\mathcal{R}$.

The original algorithm of Canny relies heavily on the polynomials defining $\mathcal{F}$ being in general position, and on the availability of a generic plane of projection. This algorithm runs in $n^k(\log n)d^{O(k^4)}$ deterministic time, and in $n^k(\log n)d^{O(k^2)}$ expected randomized time. Recent work [BPR96] addresses and overcomes the general position issue, and produces a roadmap for any semialgebraic set; the running time of this solution is $n^{k+1}d^{O(k^2)}$.

If we ignore the dependence on the degree d, the algorithm of Canny is close to optimal in the worst case, assuming that some representation of the entire $\mathcal{F}$ has to be output, since there are easy examples where the free configuration space consists of $\Omega(n^k)$ connected components.

THEOREM 40.1.2 *Roadmap Algorithm* [Can87]

Any motion planning problem, as in the preceding theorem, can be solved by the roadmap technique in $n^k(\log n)d^{O(k^4)}$ deterministic time, and in $n^k(\log n)d^{O(k^2)}$ expected randomized time.

40.1.2 LOWER BOUNDS

Both general solutions are (at least) exponential in k (but are polynomial in the other parameters when k is fixed). This raises the problem of calibrating the complexity of the problem when k can be arbitrarily large.

THEOREM 40.1.3 *Lower Bounds*

The motion planning problem, with arbitrarily many degrees of freedom, is PSPACE-hard for the instances of: (a) coordinated motion of many rectangular boxes along a rectangular floor; (b) motion planning of a planar mechanical linkage with many links; and (c) motion planning for a multi-arm robot in a 3-dimensional polyhedral environment.

All these results appear in papers collected in [HSS87]. There are also many NP-hardness results for other systems.

Facing these findings, we can either approach the general problem with heuristic and approximate schemes, or attack specific problems with small values of k, with the goal of obtaining solutions better than those yielded by the general techniques. We will mostly survey here the latter approach, and mention towards the end what has been achieved by the first approach.

40.2 MOTION PLANNING WITH A SMALL NUMBER OF DEGREES OF FREEDOM

In this main section of the chapter, we review solutions to a variety of specific motion planning problems, most of which have 2 or 3 degrees of freedom. Exploiting the special structure of these problems leads to solutions that are more efficient than the general methods described above.

GLOSSARY

Jordan arc/curve: The image of the closed unit interval under a continuous bijective mapping into the plane. A closed Jordan curve is the image of the unit circle under a similar mapping, and an unbounded Jordan curve is an image of the open unit interval (or of the entire real line) that separates the plane.

Randomized algorithm: An algorithm that applies internal randomization ("coin-flips"). We consider here algorithms that always terminate, and produce the correct output, but whose running time is a random variable that depends on the internal coin-flips. We will state upper bounds on the expectation of the running time (the ***randomized expected time***) of such an algorithm, which hold for any input. See Chapter 34.

Minkowski sum: For two planar (or spatial) sets A and B, their Minkowski sum, or pointwise vector addition, is the set $A \oplus B = \{x + y \mid x \in A, y \in B\}$.

General position: The input to a geometric problem is said to be in general position if no nontrivial algebraic identity with integer coefficients holds among the parameters that specify the input (assuming the input is not overspecified). For example: no three input points should be collinear, no four points cocircular, no three lines concurrent, etc.

Convex distance function: A convex region B that contains the origin in its interior induces a convex distance function d_B defined by

$$d_B(p, q) = \min \{\lambda \mid q \in p \oplus \lambda B\}.$$

B-Voronoi diagram: For a set S of sites, and a convex region B as above, the B-Voronoi diagram $\mathrm{Vor}_B(S)$ of S is a decomposition of space into Voronoi cells $V(s)$, for $s \in S$, such that

$$V(s) = \{p \mid d_B(p, s) \leq d_B(p, s') \text{ for all } s' \in \mathcal{S}\}.$$

Here $d_B(p, s) = \min_{q \in s} d_B(p, q)$.

$\alpha(n)$: The extremely slowly-growing inverse Ackermann function; see Section 40.4.

Contact segment: The locus of semifree placements of a polygon B translating in the plane, at each of which either some specific vertex of B touches some specific obstacle edge, or vice-versa.

Contact curve: A generalization of "contact segment" to the locus of semifree placements of B, assuming that B has only two degrees of freedom, where some specific feature of B makes contact with some specific obstacle feature.

A TRANSLATING POLYGON IN 2D

This is a system with two degrees of freedom (translations in the x and y directions).

A CONVEX POLYGON

Suppose first the translating polygon B is a *convex* k-gon, and there are m convex polygonal obstacles, $A_1, \ldots, A_m$, with pairwise disjoint interiors, having a total of n edges. The region of configuration space where B collides with A_i is the *Minkowski sum*

$$K_i = A_i \oplus (-B) = \{x - y \mid x \in A_i,\ y \in B\}.$$

The free configuration space is the complement of $\bigcup_{i=1}^{m} K_i$. Assuming general position, one can show:

THEOREM 40.2.1 [KLPS86]

(a) Each K_i is a convex polygon, with $n_i + k$ edges, where n_i is the number of edges of A_i.

(b) For each $i \neq j$, the boundaries of K_i and K_j intersect in at most two points. (This also holds when the A_i's and B are not polygons.)

(c) Given a collection of planar regions $K_1, \ldots, K_m$, each enclosed by a closed Jordan curve, such that any pair of the bounding curves intersects at most twice, then the boundary of the union $\bigcup_{i=1}^{m} K_i$ consists of at most $6m - 12$ maximal connected portions of the boundaries of the K_i's, provided $m \geq 3$, and this bound is tight in the worst case.

These properties, combined with several algorithmic techniques, imply:

THEOREM 40.2.2

(a) The free configuration space for a translating convex polygon, as above, is a polygonal region with at most $6m-12$ convex vertices and $N = \sum_{i=1}^{m}(n_i+k) = n + km$ nonconvex vertices.

(b) $\mathcal{F}$ can be computed in deterministic time $O(N \log^2 n)$, or in randomized expected time $O(N \cdot 2^{\alpha(n)} \log n)$.

AN ARBITRARY POLYGON

Suppose next that B is an arbitrary polygonal region with k edges. Let A be the union of all obstacles, which is another polygonal region with n edges. As above, the free configuration space is the complement of the Minkowski sum

$$K = A \oplus (-B) = \{x - y \mid x \in A,\ y \in B\}.$$

K is again a polygonal region, but, in this case, its maximum possible complexity is $\Theta(k^2n^2)$, so computing it might be considerably more expensive than in the convex case.

A single face suffices. If the initial placement Z of B is given, then we do not have to compute the entire (complement of) K; it suffices to compute the connected component f of the complement of K that contains Z, because no other placement is reachable from Z via a collision-free motion.

Let Σ be the collection of all contact segments; there are $2kn$ such segments. The desired component f is the face of $\mathcal{A}(\Sigma)$ that contains Z. Using the theory of *Davenport-Schinzel sequences* (Section 40.4), one can show that the maximum possible combinatorial complexity of a single face in a two-dimensional arrangement of N segments is $\Theta(N\alpha(N))$. A more careful analysis [HCA+95] shows:

THEOREM 40.2.3

(a) The maximum combinatorial complexity of a single face in the arrangement of contact segments for the case of an arbitrary translating polygon is $\Theta(kn\alpha(k))$ (this improvement is significant only when $k \ll n$).

(b) Such a face can be computed in deterministic time $O(kn\log^2 n)$, or in randomized expected time $O(kn \cdot 2^{\alpha(n)}\log n)$.

VORONOI DIAGRAMS

Another approach to motion planning for a translating *convex* object B, is via generalized *Voronoi diagrams* (see Chapter 20), based on the convex distance function $d_B(p,q)$. This function effectively places B centered at p and expands it until it hits q. The scaling factor at this moment is the d_B-distance from p to q (if B is a unit disk, d_B is the Euclidean distance). d_B satisfies the triangle inequality, and is thus "almost" a metric, except that it is not symmetric in general; it is symmetric iff B is centrally symmetric with respect to the point of reference.

Using this distance function d_B, a *B-Voronoi diagram* $\mathrm{Vor}_B(\mathcal{S})$ of $\mathcal{S}$ may be defined for a set $\mathcal{S}$ of m pairwise disjoint obstacles.

THEOREM 40.2.4

Assuming that each of B and the obstacles in $\mathcal{S}$ has constant description complexity, and that they are in general position, the B-Voronoi diagram has $O(m)$ complexity, and can be computed in $O(m\log m)$ time (in an appropriate model of computation). If B and the obstacles are convex polygons, as above, then the complexity of $\mathrm{Vor}_B(\mathcal{S})$ is $O(N)$ and it can be computed in time $O(N\log m)$.

One can show that if Z_1 and Z_2 are two free placements of B, then there exists a collision-free motion from Z_1 to Z_2 if and only if there exists a collision-free motion of B where its center moves only along the edges of $\mathrm{Vor}_B(\mathcal{S})$, between two corresponding placements W_1, W_2, where W_i, for $i = 1, 2$, is the placement obtained by pushing B from the placement Z_i away from its d_B-nearest obstacle, until it becomes equally nearest to two or more obstacles (so that its center lies on an edge of $\mathrm{Vor}_B(\mathcal{S})$).

Thus motion planning of B reduces to a path-searching in the 1-dimensional network of edges of $\mathrm{Vor}_B(\mathcal{S})$. This technique is called the ***retraction technique***, and can be regarded as a special case of the general roadmap algorithm. The

resulting motions have "high clearance," and so are safer than arbitrary motions, because they stay equally nearest to at least two obstacles.

THEOREM 40.2.5

The motion planning problem for a convex object B translating amidst m convex and pairwise disjoint obstacles can be solved in $O(m \log m)$ time, by constructing and searching in the B-Voronoi diagram of the obstacles, assuming that B and the obstacles have constant description complexity each. If B and the obstacles are convex polygons, then the same technique yields an $O(N \log m)$ solution, where $N = n + km$ is as above.

THE GENERAL MOTION PLANNING PROBLEM WITH TWO DEGREES OF FREEDOM

If B is any system with two degrees of freedom, its configuration space is 2-dimensional, and, for simplicity, let us think of it as the plane (spaces that are topologically more complex can be decomposed into a constant number of "planar" patches). We construct a collection Σ of contact curves, which, under reasonable assumptions concerning B and the obstacles, are each an algebraic Jordan arc or curve of some fixed maximum degree b. In particular, each pair of contact curves will intersect in at most some constant number, $s \leq b^2$, of points.

As above, it suffices to compute the single face of $\mathcal{A}(\Sigma)$ that contains the initial placement of B. The theory of Davenport-Schinzel sequences implies that the complexity of such a face is $O(\lambda_{s+2}(n))$, where $\lambda_{s+2}(n)$ is the maximum length of an $(n, s+2)$-Davenport-Schinzel sequence (Section 40.4), which is slightly superlinear in n when s is fixed.

The face in question can be computed in deterministic time $O(\lambda_{s+2}(n) \log^2 n)$, using a fairly involved divide-and-conquer technique based on line-sweeping; see Section 21.5. (Some slight improvements in the running time have been obtained recently.) Using randomized incremental (or divide-and-conquer) techniques, the face can be computed in randomized expected $O(\lambda_{s+2}(n) \log n)$ time.

THEOREM 40.2.6

Under the above assumptions, the general motion planning problem for systems with two degrees of freedom can be solved in deterministic time $O(\lambda_{s+2}(n) \log^2 n)$, or in $O(\lambda_{s+2}(n) \log n)$ randomized expected time.

40.2.2 THREE DEGREES OF FREEDOM

A ROD IN A PLANAR POLYGONAL ENVIRONMENT

We next pass to systems with three degrees of freedom. Perhaps the simplest instance of such a system is the case of a line segment B ("rod," "ladder," "pipe") moving (translating and rotating) in a planar polygonal environment with n edges. The maximum combinatorial complexity of the free configuration space $\mathcal{F}$ of B is $\Theta(n^2)$ (recall that the naive bound for systems with three degrees of freedom is $O(n^3)$). A cell-decomposition representation of $\mathcal{F}$ can be constructed in (deter-

ministic) $O(n^2 \log n)$ time [LS87b]. Several alternative near-quadratic algorithms have also been developed, including one based on constructing a Voronoi diagram in $\mathcal{F}$ [OSY87].

An $\Omega(n^2)$ lower bound for this problem has been established in [KO88]. It exhibits a polygonal environment with n edges and two free placements of B that are reachable from each other. However, any free motion between them requires $\Omega(n^2)$ "elementary moves," that is, the specification of any such motion requires $\Omega(n^2)$ complexity. This is a fairly strong lower bound, since it does not rely on lower bounding the complexity of the free configuration space (or of a single connected component thereof); after all, it is not clear why a motion planning algorithm should have to produce a full description of the whole free space (or of a single component).

THEOREM 40.2.7

Motion planning for a rod moving in a polygonal environment bounded by n edges can be performed in $O(n^2 \log n)$ time. There are instances where any collision-free motion of the rod between two specified placements requires $\Omega(n^2)$ "elementary moves."

A CONVEX POLYGON IN A PLANAR POLYGONAL ENVIRONMENT

Here B is a convex k-gon, free to move (translate and rotate) in an arbitrary polygonal environment bounded by n edges. The free configuration space is 3-dimensional, and there are at most $2kn$ contact surfaces, of maximum degree 4. The naive bound on the complexity of $\mathcal{F}$ is $O((kn)^3)$ (attained if B is nonconvex), but, using Davenport-Schinzel sequences, one can show that the complexity of $\mathcal{F}$ is only $O(kn\lambda_6(kn))$. Geometrically, a vertex of $\mathcal{F}$ is a semifree placement of B at which it makes simultaneously three obstacle contacts. The above bound implies that the number of such *critical placements* is only slightly super-quadratic (and not cubic) in kn.

Computing $\mathcal{F}$ in time close to this bound has proven more difficult, and only recently has a complete solution, running in $O(kn\lambda_6(kn) \log kn)$ time and constructing the entire $\mathcal{F}$, been attained [AAAS96].

Another approach was given in [CK93]. It computes the Delaunay triangulation of the obstacles under the distance function d_B, when the orientation of B is fixed, and then traces the discrete combinatorial changes in the diagram as the orientation varies. The number of changes was shown to be $O(k^4 n\lambda_3(n))$. Using this structure, the algorithm of [CK93] produces a high-clearance motion of B between any two specified placements, in time $O(k^4 n\lambda_3(n) \log n)$.

Since all these algorithms are fairly complicated, one might consider in practice an alternative approximate scheme, proposed in [AFK+90]. This scheme discretizes the orientation of B, solves the translational motion planning for B at each of the discrete orientations, and finds those placements of B at which it can rotate (without translating) between two successive orientations. This scheme works very well in practice.

THEOREM 40.2.8

Motion planning for a k-sided convex polygon, translating and rotating in a planar

polygonal environment bounded by n edges, can be performed in $O(kn\lambda_6(kn)\log kn)$ or $O(k^4 n\lambda_3(n)\log n)$ time.

EXTREMAL PLACEMENTS

A related problem is to find the largest free placement of B in the given polygonal environment. This has applications in manufacturing, where one wants to cut out copies of B that are as large as possible from a sheet of some material.

If only translations are allowed, the B-Voronoi diagram can be used to find the largest free homothetic copy of B. If general rigid motions are allowed, the technique of [CK93] computes the largest free similar copy of B in time $O(k^4 n\lambda_3(n)\log n)$. An alternative technique is given in [AAAS96], with randomized expected running time $O(kn\lambda_6(kn)\log^4 kn)$. Both bounds are nearly quadratic in n.

Finally, we mention the special case where the polygonal environment is the interior of a convex n-gon. This is simpler to analyze. The number of free critical placements of (similar copies of) B, at which B makes simultaneously four obstacle contacts, is $O(kn^2)$ [AAAS96], and they can all be computed in $O(kn^2\log n)$ time.

THEOREM 40.2.9

The largest similar placement of a k-sided convex polygon in a planar polygonal environment bounded by n edges can be computed in randomized expected time $O(kn\lambda_6(kn)\log^4 kn)$ or in deterministic time $O(k^4 n\lambda_3(n)\log n)$. When the environment is the interior of an n-sided convex polygon, the running time improves to $O(kn^2\log n)$.

A NONCONVEX POLYGON

Next we consider the case where B is an arbitrary polygonal region (not necessarily connected), translating and rotating in a polygonal environment bounded by n edges, as above. Here one can show that the maximum complexity of $\mathcal{F}$ is $\Theta((kn)^3)$. Using standard techniques, $\mathcal{F}$ can be constructed in $\Theta((kn)^3\log kn)$ time, an algorithm which has been implemented. However, as in the purely translational case, it suffices to construct the connected component of $\mathcal{F}$ containing the initial placement of B. The general result, stated below, for systems with three degrees of freedom, implies that the complexity of such a component is only near-quadratic in kn. An algorithm that computes the component in time $O((kn)^{2+\epsilon})$ is given in [HS96].

THEOREM 40.2.10

Motion planning for an arbitrary k-sided polygon, translating and rotating in a planar polygonal environment bounded by n edges, can be performed in time $O((kn)^{2+\epsilon})$, for any $\epsilon > 0$.

A TRANSLATING POLYTOPE IN A 3-D POLYHEDRAL ENVIRONMENT

Another interesting motion planning problem with three degrees of freedom involves a polytope B, with a total of k vertices, edges, and facets, translating amidst polyhedral obstacles in $\mathbb{R}^3$, with a total of n vertices, edges, and faces. The contact surfaces in this case are planar polygons, composed of a total of $O(kn)$ triangles in 3-space.

Without additional assumptions, the complexity of $\mathcal{F}$ can be $\Theta((kn)^3)$ in the worst case. However, the complexity of a single component is only $O((kn)^2 \log kn)$. Such a component can be constructed in $O((kn)^{2+\epsilon})$ time, for any $\epsilon > 0$ [AS94].

If B is a convex polytope, and the obstacles consist of m convex polyhedra, with pairwise disjoint interiors and with a total of n faces, the complexity of the entire $\mathcal{F}$ is $O(kmn \log m)$ and it can be constructed in $O(kmn \log^2 m)$ time [AS].

THEOREM 40.2.11

Translational motion planning for an arbitrary polytope with k facets, in an arbitrary 3-dimensional polyhedral environment bounded by n facets, can be performed in time $O((kn)^{2+\epsilon})$, for any $\epsilon > 0$. If B is a convex polytope, and there are m convex pairwise disjoint obstacles with a total of n facets, then the motion planning can be performed in $O(kmn \log^2 m)$ time.

THE GENERAL MOTION PLANNING PROBLEM WITH THREE DEGREES OF FREEDOM

The last several instances were special cases of the general motion planning problem with three degrees of freedom. In abstract terms, we have a collection Σ of N contact surfaces in $\mathbb{R}^3$, where these surfaces are assumed to be (patches of) algebraic surfaces of constant maximum degree. The free configuration space consists of some cells of the arrangement $\mathcal{A}(\Sigma)$, and a single connected component of $\mathcal{F}$ is just a single cell in that arrangement.

Inspecting the preceding cases, a unifying observation is that while the maximum complexity of the entire $\mathcal{F}$ can be $\Theta(N^3)$, the complexity of a single component is invariably only near-quadratic in N. This was recently shown in [HS95a] to hold in general: the combinatorial complexity of a single cell of $\mathcal{A}(\Sigma)$ is $O(N^{2+\epsilon})$, for any $\epsilon > 0$, where the constant of proportionality depends on ϵ and on the maximum degree of the surfaces; cf. Section 21.5.

A general-purpose algorithm for computing a single cell in such an arrangement was recently given in [SS96]. It runs in randomized expected time $O(N^{2+\epsilon})$, for any $\epsilon > 0$, and is based on *vertical decompositions* in such arrangements (see Section 21.3.2).

THEOREM 40.2.12

An arbitrary motion planning problem with three degrees of freedom, involving N contact surface patches, each of constant description complexity, can be solved in time $O(N^{2+\epsilon})$, for any $\epsilon > 0$.

40.2.3 OTHER PROBLEMS WITH FEW DEGREES OF FREEDOM

COORDINATED MOTION PLANNING

Another class of motion planning problems involves coordinated motion planning of several independently moving systems. Conceptually, this situation can be handled as just another special case of the general problem: Consider all the moving objects as a single system, with $k = \sum_{i=1}^{t} k_i$ degrees of freedom, where t is the number

of moving objects, and k_i is the number of degrees of freedom of the ith object. However, k will generally be too large, and the problem then will be more difficult to tackle.

A better approach is as follows [SS91]. Let $B_1, \ldots, B_t$ be the given independent objects. For each $i = 1, \ldots, t$, construct the free configuration space $\mathcal{F}^{(i)}$ for B_i alone (ignoring the presence of all other moving objects). The actual free configuration space $\mathcal{F}$ is a subset of $\prod_{i=1}^{t} \mathcal{F}^{(i)}$. Suppose we have managed to decompose each $\mathcal{F}^{(i)}$ into subcells of constant description complexity. Then $\mathcal{F}$ is a subset of the union of Cartesian products of the form $c_1 \times c_2 \times \cdots \times c_t$, where c_i is a subcell of $\mathcal{F}^{(i)}$.

We next compute the portion of $\mathcal{F}$ within each such product. Each such subproblem can be intuitively interpreted as the coordinated motion planning of our objects, where each moves within a small portion of space, amidst only a constant number of nearby obstacles; so these subproblems are much easier to solve. Moreover, in typical cases, for most products $P = c_1 \times c_2 \times \cdots \times c_t$ the problem is trivial, because P represents situations where the moving objects are far from one another, and so cannot interact at all, meaning that $\mathcal{F} \cap P = P$. The number of subproblems that really need to be solved will be relatively small.

The connectivity graph that represents $\mathcal{F}$ is also relatively easy to construct. Its nodes are the connected components of the intersections of $\mathcal{F}$ with each of the above cell products P, and two nodes are connected to each other if they are adjacent in the overall $\mathcal{F}$. In many typical cases, determining this adjacency is easy.

As an example, one can apply this technique to the coordinated motion planning of k disks moving in a planar polygonal environment bounded by n edges, to get a solution with $O(n^k)$ running time. Since this problem has $2k$ degrees of freedom, this is a significant improvement over the bound $O(n^{2k} \log n)$ yielded by Canny's general algorithm.

TABLE 40.2.1 Summary of motion planning algorithms.

SYSTEM	MOTION	ENVIRONMENT	df	RUNNING TIME
Convex k-gon	translation	planar polygonal	2	$O(N \log m)$
Arbitrary k-gon	translation	planar polygonal	2	$O(kn \log^2 n)$
General			2	$O(\lambda_{s+2}(n) \log^2 n)$
Line segment	trans & rot	planar polygonal	3	$O(n^2 \log n)$
Convex k-gon	trans & rot	planar polygonal	3	$O(k^4 n \lambda_3(n) \log n)$
				$O(kn\lambda_6(kn) \log n)$
Arbitrary k-gon	trans & rot	planar polygonal	3	$O((kn)^{2+\epsilon})$
Convex polytope	translation	3-d polyhedral	3	$O(kmn \log^2 m)$
Arbitrary polytope	translation	3-d polyhedral	3	$O((kn)^{2+\epsilon})$
General			3	$O(N^{2+\epsilon})$

MOTION PLANNING AND ARRANGEMENTS

As can be seen from the preceding subsections, motion planning is closely related to the study of arrangements of surfaces in higher dimensions. Motion planning has motivated many problems in arrangements, such as the problem of bounding the complexity of, and designing efficient algorithms for, computing a single cell

in an arrangement of n low-degree algebraic surface patches in d dimensions. The goal is to obtain bounds close to $O(n^{d-1})$ for both combinatorial and algorithmic problems. This has been settled satisfactorily for $d = 2, 3$, as noted above, but both problems are still open in higher dimensions. See Chapter 21 for further details.

SUMMARY

Some of the above results are summarized in Table 40.2.1. For each specific system, only one or two algorithms are listed.

40.3 VARIANTS OF THE MOTION PLANNING PROBLEM

We now briefly review several variants of the basic motion planning problem, in which additional constraints are imposed on the problem. Further material on many of these problems can be found in Chapter 41.

OPTIMAL MOTION PLANNING

The preceding section described techniques for determining the existence of a collision-free motion between two given placements of some moving system. It paid no attention to the optimality of the motion, which is an important consideration in practice. There are several problems involved in optimal motion planning. First, optimality is a notion that can be defined in many ways, each of which leads to different algorithmic considerations. Second, optimal motion planning is usually much harder than motion planning per se.

SHORTEST PATHS

The simplest case is when the moving system B is a single point. In this case the cost of the motion is simply the length of the path traversed by the point (normally, we use the Euclidean distance, but other metrics have been considered as well). We thus face the problem of computing ***shortest paths*** amidst obstacles in a two- or three-dimensional environment.

The planar case. Let V be a closed planar polygonal environment bounded by n edges, and let s (the "source") be a point in V. For any other point $t \in V$, let $\pi(s,t)$ denote the (Euclidean) shortest path from s to t within V. Finding $\pi(s,t)$ for any t is facilitated by construction of the *shortest path map* $SPM(s,V)$ from s in V, a decomposition of V into regions detailed in Chapter 24. A very recent result computes $\mathrm{SPM}(s,V)$ in optimal $O(n \log n)$ time.

The same problem may be considered in other metrics. For example, it is easier to give an $O(n \log n)$ algorithm for the shortest path problem under the L_1 or L_∞ metric. See Section 24.3.

The three-dimensional case. Let V be a closed polyhedral environment bounded by a total of n faces, edges, and vertices. Again, given two points $s, t \in V$, we wish to compute the shortest path $\pi(s,t)$ within V from s to t. Here $\pi(s,t)$ is a polygonal path, bending at *edges* (sometimes also at vertices) of V. To compute $\pi(s,t)$, we

need to solve two subproblems: to find the sequence of edges (and vertices) of V visited by $\pi(s,t)$ (the *shortest-path sequence* from s to t), and to compute the actual points of contact of $\pi(s,t)$ with these edges. These points obey the rule that the incoming angle of $\pi(s,t)$ with an edge is equal to the outgoing angle. Hence, given the shortest-path sequence of length m, we need to solve a system of m quartic equations in m variables in order to find the contact points. This can be solved either approximately, using an iterative scheme, or exactly, using techniques of computational real algebraic geometry; the latter method requires exponential time. Even the first, more "combinatorial," problem of computing the shortest-path sequence is NP-hard [CR87], so the general shortest-path problem is certainly much harder in three dimensions.

Many special cases of this problem, with more efficient solutions, have been studied. See Section 24.5.

VARIOUS OPTIMAL MOTION PLANNING PROBLEMS

Suppose next that the moving system B is a rigid body free only to translate in two or three dimensions. Then the notion of optimality is still well defined—it is the total distance traversed by (any reference point attached to) B. One can then apply the same techniques as above, after replacing the obstacles by their expanded versions. For example, if B is a convex polygon in the plane, and the obstacles are m pairwise openly-disjoint convex polygons $A_1, \ldots, A_m$, then we form the Minkowski sums $K_i = A_i \oplus (-B)$, for $i = 1, \ldots, m$, and compute a shortest path in the complement of their union. Since the K_i's may overlap, we first need to compute their union, as above. A similar approach can be used in planning shortest motion of a polyhedron translating amidst polyhedra in 3-space, etc.

If B admits more complex motions, then the notion of optimality begins to be fuzzy. For example, consider the case of a line segment ("rod") translating and rotating in a planar polygonal environment. One could measure the cost of a motion by the total distance traveled by a designated endpoint (or the centerpoint) of B, or by a weighted average between such a distance and the total turning angle of B, etc. See Section 24.3.

The notion of optimality gets even more complicated when one introduces kinematic constraints on the motion of B. It is then often challenging even without obstacles; see Section 41.5.4. A version of this problem, involving obstacles, has recently been shown to be NP-hard [AKY96].

EXPLORATORY MOTION PLANNING

If the environment in which the robot moves is not known to the system a priori, but the system is equipped with sensory devices, motion planning assumes a more "exploratory" character. If only tactile (or proximity) sensing is available, then a plausible strategy might be to move along a straight line (in physical or configuration space) directly to the target position, and when an obstacle is reached, to follow its boundary until the original straight line of motion is reached again. This technique has been developed and refined for arbitrary systems with two degrees of freedom (see, e.g., [LS87]). It can be shown that this strategy provably reaches the goal, if at all possible, with a reasonable bound on the length of the motion. This technique has been implemented on several real and simulated systems, and

has applications to maze-searching problems.

One attempt to extend this technique to a system with three degrees of freedom is given in [CY91]. This technique computes within $\mathcal{F}$ a certain 1-dimensional skeleton (roadmap) $\mathcal{R}$ which captures the connectivity of $\mathcal{F}$. The twist here is that $\mathcal{F}$ is not known in advance, so the construction of $\mathcal{R}$ has to be done in an incremental, exploratory manner. This exploration can be implemented in a controlled manner that does not require too many "probing" steps, and which enables the system to recognize when the construction of $\mathcal{R}$ has been completed (if the goal has not been reached beforehand).

If vision is also available, then other possibilities need to be considered, e.g., the system can obtain partial information about its environment by viewing it from the present placement, and then "explore" it to gain progressively more information until the desired motion can be fully planned. Results of this type can be found in [GMR92] and Section 41.7.

TIME-VARYING ENVIRONMENTS

Interesting generalizations of the motion planning problem arise when some of the obstacles in the robot's environment are assumed to be moving along known trajectories. In this case the robot's goal will be to "dodge" the moving obstacles while moving to its target placement. In this "dynamic" motion planning problem, it is reasonable to assume some limit on the robot's velocity and/or acceleration. Two studies of this problem are [SM88, RS94]. They show that the problem of avoiding moving obstacles is substantially harder than the corresponding static problem. By using time-related configuration changes to encode Turing machine states, they show that the problem is PSPACE-hard even for systems with a small and fixed number of degrees of freedom. However, polynomial-time algorithms are available in a few particularly simple special cases. Another variant of this problem involves movable obstacles, which the robot B can, say, push aside to clear its passage. Again, it can be shown that the general problem of this kind is PSPACE-hard, but that polynomial-time algorithms are available in certain special cases [Wil91].

COMPLIANT MOTION PLANNING

In realistic situations, the moving system has only approximate knowledge of the geometry of the obstacles and/or of its current position and velocity, and it has an inherent amount of error in controlling its motion. The objective is to devise a strategy that will guarantee that the system reaches its goal, where such a strategy usually proceeds through a sequence of free motions (until an obstacle is hit) intermixed with *compliant motions* (sliding along surfaces of contacted obstacles) until it can be ascertained that the goal has been reached.

A standard approach to this problem is through the construction of pre-images (or back projections). See Section 41.5.3.

NONHOLONOMIC MOTION PLANNING

Another realistic constraint on the possible motions of a given system is kinematic (or *kinodynamic*). For example, the moving object B might be constrained not to

exceed certain velocity or acceleration thresholds, or has only limited steering capability. Even without any obstacles, such problems are usually quite hard, and the presence of (stationary or moving) obstacles makes them extremely complicated to solve. These so-called *nonholonomic motion planning* problems are usually handled using tools from control theory. See Section 41.5.2.

GENERAL TASK AND ASSEMBLY PLANNING

In task planning problems, the system is given a complex task to perform, such as assembling a part from several components or restructuring its workcell into a new layout, but the precise sequence of substeps needed to attain the final goal is not specified and must be inferred by the system.

Suppose we want to manufacture a product consisting of several parts. Let S be the set of parts in their final assembled form. The first question is whether the product can be disassembled by translating in some fixed direction one part after the other, so that no collision occurs. An order of the parts that satisfies this property is called a ***depth order***. It need not always exist, but when it does, the product can be assembled by translating the constituent parts one after another, in the reverse of the depth order, to their target positions. Products that can be assembled in this manner are called ***stack products*** [WL94]. The simplicity of the assembly process makes stack products attractive to manufacture. Computing a depth order in a given direction (or deciding that no such order exists) can be done in $O(m^{4/3+\epsilon})$ time, for any $\epsilon > 0$, for a set of polygons in 3-space with m vertices in total [dBOS94]. Faster algorithms are known for the special cases of axis-parallel polygons, c-oriented polygons, and "fat" objects.

Many products, however, are not stack products, that is, a single direction in which the parts must be moved is not sufficient to assemble the product. One solution is to search for an assembly sequence that allows a subcollection of parts to be moved as a rigid body in *some* direction. This can be accomplished in polynomial time, though the running time is rather high in the worst case: it may require $\Omega(m^4)$ time for a collection of m tetrahedra in 3-space. A more modest, but considerably more efficient, solution allows each disassembly step to proceed in one of a few given directions [ABHS96]. It has running time $O(m^{4/3+\epsilon})$, for any $\epsilon > 0$. See Section 41.3 for further details on assembly sequencing, and Chapter 46 for related problems.

ON-LINE MOTION PLANNING

Consider the problem of a point robot moving through a planar environment filled with polygonal obstacles, where the robot has no a priori information about the obstacles that lie ahead. One models this situation by assuming that the robot knows the location of the target position and of its own absolute position, but that it only acquires knowledge about the obstacles as it contacts them. The goal is to minimize the distance that the robot travels. See also the discussion on exploratory motion planning above.

Because the robot must make decisions without knowing what lies ahead, it is natural to use the ***competitive ratio*** to evaluate the performance of a strategy.

In particular, one would like to minimize the ratio between the distance traveled by the robot and the length of the shortest start-to-target path in that scene. The competitive ratio is the worst-case ratio achieved over all scenes having a given source-target distance. A special case of interest is when all obstacles are axis-parallel rectangles of width at least 1 located in the infinite Euclidean plane. Natural greedy strategies yield a competitive ratio of $\Theta(n)$, where n is the Euclidean source-target distance. More sophisticated algorithms obtain competitive ratios of $\Theta(\sqrt{n})$ [BRS91]. Randomized algorithms can do much better [BBF+96]. Through the use of randomization, one can translate the case of arbitrary convex obstacles [BRS91] to rectilinearly-aligned rectangles, at the cost of some increase in the competitive ratio. If the scene is not on an infinite plane but rather within some finite rectangular "warehouse," and the start location is one of the warehouse corners, then the competitive ratio drops to $\log n$ [BBFY92].

PRACTICAL APPROACHES TO MOTION PLANNING

When the number of degrees of freedom is even moderately large, exact solutions of the motion planning problem are very inefficient in practice, so one seeks heuristic but practical solutions. Several such techniques have been developed.

Potential field and probabilistic techniques. The first heuristic regards the robot as moving in a potential field induced by the obstacles and by the target placement, where the obstacles act as repulsive barriers, and the target as a strongly attracting source. By letting the robot follow the gradient of such a potential field, we obtain a motion that avoids the obstacles and that can be expected to reach the goal. An attractive feature of this technique is that planning and executing the desired motion are done in a single stage. Another important feature is the generality of the approach; it can easily be applied to systems with many degrees of freedom.

This technique, however, may lead to a motion where the robot gets stuck at a local minimum of the potential field, leaving no guarantee that the goal will be reached. To overcome this problem, several solutions have been proposed. One is to try to escape from such a "potential well" by making a few small random moves, in the hope that one of them will put the robot in a position from which the field leads it away from this well. Another approach is to use the potential field only for subproblems where the initial and final placements are close to each other, so the chance to get stuck at a local minimum is small. One then generates many random placements throughout the workspace, and applies the potential field technique to attempt to connect many pairs of them, until a path is generated from start to goal. (In this randomized technique, any convenient local planner may be used.) See [Lat91, KSLO] and Section 41.4 for more details concerning this technique.

Fat obstacles. Another technique exploits the fact that, in typical layouts, the obstacles can be expected to be "fat" (this has several definitions; intuitively, they do not have long and skinny parts). Also, the obstacles tend not to be too clustered, in the sense that each placement of the robot can interact with only a constant number of obstacles. These facts tend to make the problem easier to solve. See [SO94] for such a solution.

40.4 DAVENPORT-SCHINZEL SEQUENCES

Davenport-Schinzel sequences are interesting and powerful combinatorial structures that arise in the analysis and calculation of the lower or upper envelope of collections of functions, and therefore have applications in many geometric problems, including numerous motion planning problems, which can be reduced to the calculation of such an envelope. A recent comprehensive survey of Davenport-Schinzel sequences and their geometric applications can be found in [SA95].

An (n, s) ***Davenport-Schinzel sequence***, where n and s are positive integers, is a sequence $U = (u_1, \ldots, u_m)$ composed of n symbols with the properties:

(i) No two adjacent elements of U are equal: $u_i \neq u_{i+1}$ for $i = 1, \ldots, m-1$.

(ii) U does not contain as a subsequence any alternation of length $s+2$ between two distinct symbols: there do not exist $s+2$ indices $i_1 < i_2 < \cdots < i_{s+2}$ so that $u_{i_1} = u_{i_3} = u_{i_5} = \cdots = a$ and $u_{i_2} = u_{i_4} = u_{i_6} = \cdots = b$, for two distinct symbols a and b.

Thus, for example, an $(n, 3)$ sequence is not allowed to contain any subsequence of the form $(a \cdots b \cdots a \cdots b \cdots a)$. Let $\lambda_s(n)$ denote the maximum possible length of an (n, s) Davenport-Schinzel sequence.

The importance of Davenport-Schinzel sequences lies in their relationship to the combinatorial structure of the lower (or upper) envelope of a collection of functions (Section 21.2). Specifically, for any collection of n real-valued continuous functions $f_1, \ldots, f_n$ defined on the real line, having the property that each pair of them intersect in at most s points, one can show that the sequence of function indices i in the order in which these functions attain their lower envelope (i.e., their pointwise minimum $f = \min_i f_i$) from left to right is an (n, s) Davenport-Schinzel sequence. Conversely, any (n, s) Davenport-Schinzel sequence can be realized in this way for an appropriate collection of n continuous univariate functions, each pair of which intersect in at most s points.

The crucial and surprising property of Davenport-Schinzel sequences is that, for a fixed s, the maximal length $\lambda_s(n)$ is nearly linear in n, although for $s \geq 3$ it is slightly super-linear. Specifically, one has

$$\begin{aligned}
\lambda_1(n) &= n \\
\lambda_2(n) &= 2n - 1 \\
\lambda_3(n) &= \Theta(n\alpha(n)) \\
\lambda_4(n) &= \Theta(n \cdot 2^{\alpha(n)}) \\
\lambda_{2s}(n) &\leq n \cdot 2^{\alpha(n)^{s-1} + C_{2s}(n)} \\
\lambda_{2s+1}(n) &\leq n \cdot 2^{\alpha(n)^{s-1} \log \alpha(n) + C_{2s+1}(n)} \\
\lambda_{2s}(n) &= \Omega(n \cdot 2^{\frac{1}{(s-1)!}\alpha(n)^{s-1} + C'_{2s}(n)}),
\end{aligned}$$

where $\alpha(n)$ is the inverse of Ackermann's function, and where $C_r(n)$, $C'_r(n)$ are asymptotically smaller than the leading terms in the respective exponents. Ackermann's function $A(n)$ grows extremely quickly, with $A(4)$ an exponential "tower" of 65636 2's. Thus $\alpha(n) \leq 4$ for all practical values of n. See [SA95].

If one considers the lower envelope of n continuous, but only partially defined, functions, then the complexity of the envelope is at most $\lambda_{s+2}(n)$, where s is the maximum number of intersections between any pair of functions. Thus for a collection of n line segments (for which $s = 1$), the lower envelope consists of at most $O(n\alpha(n))$ subsegments. A surprising result is that this bound is tight in the worst case: there are collections of n segments, for arbitrarily large n, whose lower envelope does consist of $\Omega(n\alpha(n))$ subsegments. This is perhaps the most natural example of a combinatorial structure defined in terms of n simple objects, whose complexity involves the inverse Ackermann's function.

Algorithms. The lower envelope of n given total or partial continuous functions, each pair of which intersect in at most s points, can be computed by a simple divide-and-conquer technique that runs (in an appropriate model of computation) in time $O(\lambda_s(n) \log n)$ or $O(\lambda_{s+2}(n) \log n)$ (depending on whether the functions are totally or partially defined). A refined technique reduces the time for partially-defined functions to $O(\lambda_{s+1}(n) \log n)$. Thus, in the case of segments, the algorithm computes their lower envelope in optimal $O(n \log n)$ time. More complex combinatorial and algorithmic applications of Davenport-Schinzel sequences (such as the complexity and construction of a single face in a planar arrangement) are mentioned throughout this chapter.

40.5 SOURCES AND RELATED MATERIAL

SURVEYS

All results not given an explicit reference above, and additional material on motion planning and related problems may be traced in these surveys:

[Lat91]: A book devoted to robot motion planning.

[HSS87]: A collection of early papers on motion planning.

[SA95]: A book on Davenport-Schinzel sequences and their geometric applications; contains a section on motion planning.

[HS95b]: A recent review on arrangements and their applications to motion planning.

[SS88, SS90, Sha89, Sha95, AY90]: Several survey papers on algorithmic motion planning.

RELATED CHAPTERS

Chapter 20: Voronoi diagrams and Delaunay triangulations
Chapter 21: Arrangements
Chapter 24: Shortest paths and networks
Chapter 29: Computational real algebraic geometry
Chapter 41: Robotics

REFERENCES

[AAAS96] P.K. Agarwal, N. Amenta, B. Aronov, and M. Sharir. Largest placements and motion planning of a convex polygon. In *Proc. 2nd Annu. Workshop Algorithmic Found. Robot.*, Toulouse, 1996.

[ABHS96] P. Agarwal, M. de Berg, D. Halperin, and M. Sharir. Efficient generation of k-directional assembly sequences. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pages 122–131, 1996.

[AFK+90] H. Alt, R. Fleischer, M. Kaufmann, K. Mehlhorn, S. Näher, S. Schirra, and C. Uhrig. Approximate motion planning and the complexity of the boundary of the union of simple geometric figures. In *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, pages 281–289, 1990.

[AKY96] T. Asano, D. Kirkpatrick, and C.K. Yap. d_1-optimal motion for a rod. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 252–263, 1996.

[AS94] B. Aronov and M. Sharir. Castles in the air revisited. *Discrete Comput. Geom.*, 12:119–150, 1994.

[AS] B. Aronov and M. Sharir. On translational motion planning of a convex polyhedron in 3-space. *SIAM J. Comput.*, to appear.

[AY90] H. Alt and C.K. Yap. Algorithmic aspects of motion planning: A tutorial, Parts 1 and 2. *Algorithms Rev.*, 1:43–60, 61–77, 1990.

[BBFY92] E. Bar-Eli, P. Berman, A. Fiat, and P. Yan. On-line navigation in a room. In *Proc. 3rd ACM-SIAM Sympos. Discrete Algorithms*, pages 237–249, 1992.

[BBF+96] P. Berman, A. Blum, A. Fiat, H. Karloff, A. Rosen, and M. Saks. Randomized robot navigation algorithms. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pages 75–84, 1996.

[BPR96] S. Basu, R. Pollack, and M.-F. Roy. Computing roadmaps of semi-algebraic sets. In *Proc. 28th Annu. ACM Sympos. Theory Comput.*, pages 168–173, 1996.

[BRS91] A. Blum, P. Raghavan, and B. Schieber. Navigating in unfamiliar geometric terrain. In *Proc. 23rd Annu. ACM Sympos. Theory Comput.*, pages 494–504, 1991.

[Can87] J. Canny. *The Complexity of Robot Motion Planning.* MIT Press, Cambridge, 1987. See also: Computing roadmaps in general semi-algebraic sets. *Comput. J.*, 36:504–514, 1993.

[CK93] L.P. Chew and K. Kedem. A convex polygon among polygonal obstacles: placement and high-clearance motion. *Comput. Geom. Theory Appl.*, 3:59–89, 1993.

[CR87] J. Canny and J.H. Reif. New lower bound techniques for robot motion planning problems. In *Proc. 28th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 49–60, 1987.

[CY91] J. Cox and C.K. Yap. On-line motion planning: Case of a planar rod. *Ann. Math. Artif. Intell.*, 3:1–20, 1991.

[dBOS94] M. de Berg, M. Overmars, and O. Schwarzkopf. Computing and verifying depth orders. *SIAM J. Comput.*, 23:437–446, 1994.

[GMR92] L. Guibas, R. Motwani, and P. Raghavan. The robot localization problem in two dimensions. In *Proc. 3rd ACM-SIAM Sympos. Discrete Algorithms*, pages 259–268, 1992.

[HCA+95] S. Har-Peled, T.M. Chan, B. Aronov, D. Halperin, and J. Snoeyink. The complexity of a single face of a Minkowski sum. In *Proc. 7th Canad. Conf. Comput. Geom.*, Québec City, pages 91–96, 1995.

[HS95a] D. Halperin and M. Sharir. Almost tight upper bounds for the single cell and zone problems in three dimensions. *Discrete Comput. Geom.*, 14:385–410, 1995.

[HS95b] D. Halperin and M. Sharir. Arrangements and their applications in robotics: Recent developments. In K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson, editors, *The Algorithmic Foundations of Robotics*, pages 495–511. A K Peters, Boston, 1995.

[HS96] D. Halperin and M. Sharir. A near-quadratic algorithm for planning the motion of a polygon in a polygonal environment. *Discrete Comput. Geom.*, 16:121–134, 1996.

[HSS87] J.E. Hopcroft, J.T. Schwartz, and M. Sharir, editors. *Planning, Geometry, and Complexity of Robot Motion.* Ablex, Norwood, 1987.

[KLPS86] K. Kedem, R. Livne, J. Pach, and M. Sharir. On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete Comput. Geom.*, 1:59–71, 1986.

[KO88] Y. Ke and J. O'Rourke. Lower bounds on moving a ladder in two and three dimensions. *Discrete Comput. Geom.*, 3:197–217, 1988.

[KSLO] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high dimensional configuration spaces. *IEEE Trans. Robot. Autom.*, to appear. Also Tech. Rept. STANFORD-CS-TR-94-1519.

[Lat91] J.-C. Latombe. *Robot Motion Planning.* Kluwer, Boston, 1991.

[LS87a] D. Leven and M. Sharir. Planning a purely translational motion for a convex object in two-dimensional space using generalized Voronoi diagrams. *Discrete Comput. Geom.*, 2:9–31, 1987.

[LS87b] D. Leven and M. Sharir. An efficient and simple motion planning algorithm for a ladder moving in 2-dimensional space amidst polygonal barriers. *J. Algorithms*, 8:192–215, 1987.

[LS87] V.J. Lumelsky and A.A. Stepanov. Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, 2:403–430, 1987.

[OSY87] C. Ó'Dúnlaing, M. Sharir, and C.K. Yap. Generalized Voronoi diagrams for a ladder: II. Efficient construction of the diagram. *Algorithmica*, 2:27–59, 1987.

[RS94] J. Reif and M. Sharir. Motion planning in the presence of moving obstacles. *J. Assoc. Comput. Mach.*, 41:764–790, 1994.

[SA95] M. Sharir and P.K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications.* Cambridge University Press, 1995.

[Sha89] M. Sharir. Algorithmic motion planning in robotics. *Computer*, 22:9–20, 1989.

[Sha95] M. Sharir. Robot motion planning. *Comm. Pure Appl. Math.*, 48:1173–1186, 1995. Also in E. Schonberg, editor, *The Houses that Jack Built.* Courant Institute, New York, 1995, 287–300.

[SO94] A.F. van der Stappen and M.H. Overmars. Motion planning amidst fat obstacles. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 31–40, 1994.

[SM88] K. Sutner and W. Maass. Motion planning among time-dependent obstacles. *Acta Inform.*, 26:93–122, 1988.

[SS83] J.T. Schwartz and M. Sharir. On the piano movers' problem: II. General techniques for computing topological properties of real algebraic manifolds. *Adv. Appl. Math.*, 4:298–351, 1983.

[SS88] J.T. Schwartz and M. Sharir. A survey of motion planning and related geometric algorithms. *Artif. Intell.*, 37:157–169, 1988. Also in D. Kapur and J. Mundy, editors,

Geometric Reasoning, pages 157–169. MIT Press, Cambridge, 1989. And in S.S. Iyengar and A. Elfes, editors, *Autonomous Mobile Robots*, volume I, pages 365–374. IEEE Computer Society Press, Los Alamitos, 1991.

[SS90] J.T. Schwartz and M. Sharir. Algorithmic motion planning in robotics. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*, pages 391–430. Elsevier, Amsterdam, 1990.

[SS91] M. Sharir and S. Sifrony. Coordinated motion planning for two independent robots. *Ann. Math. Artif. Intell.*, 3:107–130, 1991.

[SS96] O. Schwarzkopf and M. Sharir. Vertical decomposition of a single cell in a 3-dimensional arrangement of surfaces and its applications. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 20–29, 1996.

[Wil91] G. Wilfong. Motion planning in the presence of movable obstacles. *Ann. Math. Artif. Intell.*, 3:131–150, 1991.

[WL94] R.H. Wilson and J.-C. Latombe. Geometric reasoning about mechanical assembly. *Artif. Intell.*, 71:371–396, 1994.

[Yap87] C.K. Yap. An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments. *Discrete Comput. Geom.*, 2:365–393, 1987.

41 ROBOTICS

Dan Halperin, Lydia Kavraki, and Jean-Claude Latombe

INTRODUCTION

Robotics is concerned with the generation of computer-controlled motions of physical objects in a wide variety of settings. Because physical objects define spatial distributions in 3-space, geometric representations and computations play an important role in robotics. As a result the field is a significant source of practical problems for computational geometry. There are substantial differences, however, in the ways researchers in robotics and in computational geometry address related problems. Robotics researchers are primarily interested in developing methods that work well in practice and can be combined into integrated systems. Unlike researchers in computational geometry, they often pay little attention to the underlying combinatorial and complexity issues (the focus of Chapter 40). This difference in approach will become clear in the present chapter.

In Section 41.1 we survey basic definitions and problems in robot kinematics. Part manipulation is discussed in Section 41.2 with emphasis on part grasping, fixturing, and feeding. In Section 41.3 we present algorithms for assembly sequencing. The basic path planning problem is the topic of Section 41.4. Extensions of this problem, in particular nonholonomic motion planning, are discussed in Section 41.5. We briefly survey additional topics in two sections that follow: data structures for representing moving objects in Section 41.6, and sensing and localization in Section 41.7.

GLOSSARY

Workspace W: A subset of the 2- or 3-dimensional physical space: $W \subset \mathbb{R}^k$ ($k = 2$ or 3).

Body: Rigid physical object modeled as a compact manifold with boundary $B \subset \mathbb{R}^k$ ($k = 2$ or 3). B's boundary is assumed piecewise-smooth. We will use the terms "body," "physical object," and "part" interchangeably.

Robot: A collection of bodies capable of generating their own motions.

Configuration: Any mathematical specification of the position and orientation of every body composing a robot, relative to a fixed coordinate system. The configuration of a single body is also called a ***placement*** or a ***pose***.

Configuration space $\mathcal{C}$: Set of all configurations of a robot. For almost any robot, this set is a smooth manifold. We will always denote the configuration space of a robot by $\mathcal{C}$ and its dimension by m. Given a robot A, we will let $A(\boldsymbol{q})$ denote the subset of the workspace occupied by A at configuration $\boldsymbol{q}$.

Number of degrees of freedom: The dimension m of $\mathcal{C}$. In the following we will abbreviate "degree of freedom" by ***dof***.

41.1 KINEMATICS

Many robots consist of multiple bodies connected by joints, which may be either actuated or passive. The spatial relations among these bodies and the space of their feasible motions is an important area of study in robotics. Cf. Section 48.4.1.

GLOSSARY

Linkage: A collection of bodies, called ***links***, in which some pairs of links are connected by ***joints***. The graph whose nodes (resp. edges) represent links (resp. joints) is connected.

Prismatic joint: A joint between two links that allows one link to translate along a line attached to the other.

Revolute joint: A joint between two links that allows one link to rotate about a line attached to the other.

Joint parameter: A real parameter associated with a prismatic or revolute joint whose value uniquely determines the relative position or orientation of the two links connected by that joint.

Robot arm: Serial linkage such that the first link, called the ***base***, is fixed in space. The last link is called the ***end-effector***.

There are other types of joints besides the prismatic and revolute joints considered in this chapter. Most of them can be reduced to independent prismatic and/or revolute joints. For example, a ***telescopic joint*** is equivalent to collinear prismatic joints connecting links that penetrate one another. We also note that some industrial robot arms contain closed mechanical loops. For many computational purposes, however, they can be considered as serial linkages, as we assume here.

NUMBER OF DEGREES OF FREEDOM OF A LINKAGE

Let L be an arbitrary linkage with n_{link} links and n_{joint} joints, with each joint either prismatic or revolute. The number of dofs of L, denoted by n_{dof}, is the number of joints in L that can move independently with the others complying, and is given by the Grübler formula [Rot94]:

$$n_{dof} \geq n_0(n_{link} - 1) - (n_0 - 1)n_{joint}\,,$$

where $n_0 = 3$ if the linkage is planar, and $n_0 = 6$ if the linkage is in 3-space. In general, this formula holds with equality. The strict "greater-than" is needed only for mechanisms with special proportions or alignments.

If L is a serial linkage, we have $n_{link} = n_{joint} + 1$. So $n_{dof} = n_{joint}$. If L consists of a single closed loop, we have $n_{link} = n_{joint}$. So $n_{dof} = n_{joint} - n_0$; thus, one degree of freedom requires 4 joints in 2-space and 7 joints in 3-space. If L consists of multiple loops, the Grübler formula yields $n_{dof} = n_{joint} - n_0\ell$, where ℓ is the number of independent loops.

FORWARD AND INVERSE KINEMATICS

The number of dofs of a robot arm is equal to its number of joints. The determination of the placement of the end-effector from the joint parameters is called the ***direct kinematics problem.*** In order for the last link's placement to span a 6-space, the arm must have at least 6 joints. (See Figure 48.4.1.)

The determination of the values of the arm's joint parameters from the last link's placement is called the ***inverse kinematics problem.*** For a 6-joint arm this problem has at most 16 distinct solutions (except for some singularities). In other words, at most 16 distinct legal placements of the arm's links achieve the same specified placement of the end-effector. If the arm has two prismatic joints, then the maximum drops to 8. If it has three prismatic joints, it drops to 2. Any time three consecutive revolute joints have intersecting or parallel axes, the number is at most 8 (see [Rot94]).

OPEN PROBLEM

Given a workspace W, find the optimal design of a robot arm that can reach everywhere in W without collision. Several variants of this problem are solved in [Kol95]. However the three-dimensional case is largely open. An extension of this problem also asks for a design of the layout of the workspace so that a certain task can be completed efficiently. (Additional reachability problems for planar robot arms and their solutions are presented in [O'R94, Section 8.6].)

41.2 PART MANIPULATION

Part manipulation is one of the most frequently performed operations in industrial robotics: parts are grasped from conveyor belts, they are oriented prior to feeding assembly workcells, and they are immobilized for machining operations.

GLOSSARY

Wrench: A pair $[\boldsymbol{f}, \boldsymbol{p} \times \boldsymbol{f}]$, where $\boldsymbol{p}$ denotes a point in the boundary of a body B, represented by its coordinate vector in a frame attached to B, $\boldsymbol{f}$ designates a force applied to B at $\boldsymbol{p}$, and $\times$ is the vector cross-product. If $\boldsymbol{f}$ is a unit vector, the wrench is said to be a ***unit*** wrench.

Finger: A tool that can apply a wrench.

Grasp: A set of unit wrenches $\boldsymbol{w}_i = [\boldsymbol{f}_i, \boldsymbol{p}_i \times \boldsymbol{f}_i]$, $i = 1, \ldots, p$, defined on a body B, each created by a finger in contact with the boundary, ∂B, of B. For each $\boldsymbol{w}_i$, if the contact is frictionless, $\boldsymbol{f}_i$ is normal to ∂B at $\boldsymbol{p}_i$; otherwise, it can span the friction cone defined by the Coulomb law.

Force-closure grasp: A grasp $\{\boldsymbol{w}_i\}_{i=1,\ldots,p}$ such that, for any arbitrary wrench $\boldsymbol{w}$, there exists a set of real values $\{f_1, \ldots, f_p\}$ achieving $\Sigma_{i=1}^{p} f_i \boldsymbol{w}_i = -\boldsymbol{w}$. In other words, a force-closure grasp can resist any external wrenches applied to B. If contacts are nonsticky, we require that $f_i \geq 0$ for all $i = 1, \ldots, p$, and the

grasp is called ***positive***. In this section we only consider positive grasps.

Form-closure grasp: A positive force-closure grasp in which all finger-body contacts are frictionless.

41.2.1 GRASPING

Grasp analysis and synthesis has been an active research area over the last decade and has contributed to the development of robotic hands and grasping mechanisms.

SIZE OF A FORM/FORCE CLOSURE GRASP

The following results are shown in [MNP90, MSS87]:

- Bodies with rotational symmetry (e.g., disks in 2-space, spheres and cylinders in 3-space) admit no form-closure grasps.
- All other bodies admit a form-closure grasp with at most four fingers in 2-space and twelve fingers in 3-space.
- All polyhedral bodies have a form-closure grasp with seven fingers.
- With frictional finger-body contacts, all bodies admit a force-closure grasp that consists of 3 fingers in 2-space and four fingers in 3-space.

TESTING FOR FORM/FORCE CLOSURE

A necessary and sufficient condition for force closure in 2-space (resp. 3-space) is that the finger wrenches span three (resp. six) dimensions and that a strictly positive linear combination of them be zero. Said otherwise, the null wrench (the origin) should lie in the interior of the convex hull H of the finger wrenches [MSS87]. This condition provides an effective test for deciding in constant time whether a given grasp achieves force closure. A related quantitative measure of the quality of a grasp (one among several metrics proposed) is the radius of the maximum ball centered at the origin and contained in the convex hull H [KMY92].

SYNTHESIZING FORM/FORCE CLOSURE GRASPS

Most research has concentrated on computing grasps with two to four nonsticky fingers. Optimization techniques and elementary Euclidean geometry are used in [MNP90] to derive an algorithm computing a single force-closure grasp of a polygonal or polyhedral part. This algorithm is linear in the part complexity. Other linear-time techniques using results from combinatorial geometry (Steinitz's theorem) are presented in [MSS87, Mis95]. Optimal force-closure grasps are synthesized in [FC92] by maximizing the set of external wrenches that can be balanced by the contact wrenches.

Finding the maximal regions on a body where fingers can be positioned independently while achieving force closure makes it possible to accommodate errors in finger placement. Geometric algorithms for constructing such regions are pro-

posed in [Ngu88] for grasping polygons with two fingers (with friction) and four fingers (without friction), and for grasping polyhedra with three fingers (with frictional contact capable of generating torques) and seven fingers (without friction). Curved obstacles have also been studied [PSS+95]. The latter paper contains a good overview of work on the effect of curvature at contact points on grasp planning.

DEXTROUS GRASPING

Reorienting a part by moving fingers on the part's surface is often considered to lie in the broader realm of grasping. Finger gait algorithms and nonholonomic rolling contacts (Section 41.5.2) for fingertips have been explored.

41.2.2 FIXTURING

Most manufacturing operations require fixtures to hold parts. To avoid the custom design of fixtures for each part, modular reconfigurable fixtures are often used. A typical modular fixture consists of a workholding surface, usually a plane, that has a lattice of holes where locators, clamps, and edge fixtures can be placed. Locators are simple round pins, while clamps apply pressure on the part.

Contacts between fixture elements and parts are generally assumed to be frictionless. In modular fixturing, contact locations are restricted by the lattice of holes, and form closure cannot always be achieved. In particular, when three locators and one clamp are used on a workholding plane, there exist polygons of arbitrary size for which no fixture design can be achieved [ZGW94]. But if parts are restricted to be rectilinear, a fixture can always be found as long as all edges have length at least four lattice units [Mis91].

When the fixturing kit consists of a latticed workholding plane, three locators, and one clamp, the algorithm in [BG96] finds all possible placements of a given part on the workholding surface where form closure can be achieved, along with the corresponding positions of the locators and the clamp. The algorithm in [ORSW95] computes the form-closure fixtures of input polygonal parts using a kit containing one edge fixture, one locator, and one clamp.

An algorithm for fixturing an assembly of parts that are not rigidly fastened together is proposed in [Mat95]. A large number of fixturing contacts are first scattered at random on the external boundary of the assembly. Redundant contacts are then pruned until the stability of the assembly is no longer guaranteed.

41.2.3 PART FEEDING

Part feeders account for a large fraction of the cost of a robotic assembly workcell. A typical feeder must bring parts at subsecond rates with high reliability. Part feeding often relies on ***nonprehensile manipulation***. Nonprehensile manipulation exploits task mechanics to achieve a goal state without grasping and frequently allows accomplishing complex feeding tasks with few dofs. It may also enable a robot to move parts that are too large or heavy to be grasped and lifted.

Pushing is one form of nonprehensile manipulation. Work on pushing originated in [Mas82] where a simple rule is established to qualitatively determine the motion of a pushed object. This rule makes use of the position of the center of friction of the object on the supporting surface. This result has been used in several nonprehensile

manipulation algorithms:

- A planning algorithm for a robot that tilts a tray containing a planar part of known shape to orient it to a desired orientation [EM88].
- An algorithm to compute the design of a sequence of curved fences along a conveyor belt to reorient a given polygonal part [WGPB96].
- An algorithm that computes a sequence of motions of a single articulated fence on a conveyor belt that achieves a goal orientation of an object [AHLM96].

A frictionless parallel-jaw gripper was used in [Gol93] to orient polygonal parts. For any part P having an n-sided convex hull, there exists a sequence of $2n-1$ squeezes achieving a single orientation of P (up to symmetries of the convex hull). This sequence is computed in $O(n^2 \log n)$ time. This result has been generalized to planar parts having a piecewise algebraic convex hull [RG95].

New fabrication techniques, such as MEMS, can be used to build arrays of tiny actuators that can be programmed to orient mechanical parts [BDM96].

OPEN PROBLEMS

A major open practical problem is to predict feeder throughputs to evaluate alternative feeder designs, given the geometry of the parts to be manipulated. In relation to this problem, simulation algorithms have been proposed recently to predict the pose of a part dropped on a horizontal surface [MZG+96].

41.3 ASSEMBLY SEQUENCING

Most mechanical products consist of multiple parts. The goal of assembly sequencing is to compute both an order in which parts can be assembled and the corresponding required movements of the parts.

GLOSSARY

Assembly: Collection of bodies in some given relative placements.

Subassembly: Subset of the bodies composing an assembly A in their relative positions and orientations in A.

Separated subassemblies: Subassemblies that are arbitrarily far apart from one another.

Hand: A tool that can hold an arbitrary number of bodies in fixed relative placements.

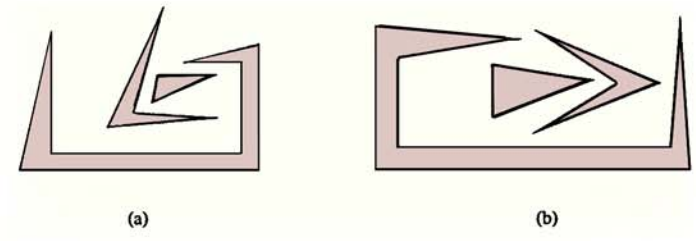
Assembly operation: A motion that merges s pairwise-separated subassemblies ($s \geq 2$) into a new subassembly; each subassembly moves as a single body. No overlapping between bodies is allowed during the operation. The parameter s is called the ***number of hands of the operation.*** We call the reverse of an assembly operation ***assembly partitioning.***

Assembly sequence: A total ordering on assembly operations that merge the

separated parts composing an assembly into this assembly. The maximum, over all the operations in the sequence, of the number of hands required by an operation is called the ***number of hands of the sequence.***

Monotone assembly sequence: A sequence in which no operation brings a body to an intermediate placement (relative to other bodies), before another operation transfers it to its final placement. See Figure 41.3.1.

FIGURE 41.3.1
Both assemblies below admit two-handed sequences with translational motions only. While (a) *accepts such a monotone sequence,* (b) *does not. To disassemble* (b) *the triangle must be translated to an intermediate position* [HW95]. *If general motions are accepted, there exists a monotone two-handed sequence for* (b). *A monotone three-handed sequence with translations only is also possible.*



NUMBER OF HANDS IN ASSEMBLY

Every assembly of convex polygons in the plane has a two-handed assembly sequence of translations. In the worst-case, s hands are necessary and sufficient for assemblies of s star-shaped polygons/polyhedra [Nat88].

There exists an assembly of six tetrahedra without a two-handed assembly sequence of translations, but with a three-handed sequence of translations. Every assembly of five or fewer convex polyhedra admits a two-handed assembly sequence of translations. There exists an assembly of thirty convex polyhedra that cannot be assembled with two hands [SS94].

COMPLEXITY OF ASSEMBLY SEQUENCING

When arbitrary sequences are allowed, assembly sequencing is PSPACE-hard. The problem remains PSPACE-hard even when the bodies are polygons, each with a constant number of vertices [Nat88].

When only two-handed monotone sequences are permitted, deciding if an assembly A can be partitioned into two subassemblies S and $A\backslash S$ such that they can be separated by an arbitrary motion is NP-complete [WKL$^+$95]. The problem remains NP-complete when both S and $A\backslash S$ are required to be connected and motions are restricted to translations [KK95].

MONOTONE TWO-HANDED ASSEMBLY SEQUENCING

A popular approach to assembly sequencing is disassembly sequencing [HS91]. A sequence that separates an assembly to its individual components is first generated

and next reversed. Most existing assembly sequencers can only generate two-handed monotone sequences. Such a sequence is computed by partitioning the assembly and, recursively, the resulting subassemblies into two separated subassemblies.

The ***nondirectional blocking graph*** (NDBG) is proposed in [WL95] to represent all the blocking relations in an assembly. It is a subdivision of the space of all allowable motions of separation into a finite number of cells such that within each cell the set of blocking relations between all pairs of parts remains fixed. Within each cell this set is represented in the form of a directed graph, called the directional blocking graph (DBG). The NDBG is the collection of the DBGs over all the cells in the subdivision.

We illustrate this approach for polyhedral assemblies when the allowable motions are infinite translations. The partitioning of an assembly consisting of polyhedral parts into two subassemblies is done as follows. For an ordered pair of parts P_i, P_j, the 3-vector $\vec{d}$ is a ***blocking direction*** if translating P_i to infinity in direction $\vec{d}$ will cause P_i to collide with P_j. For each ordered pair of parts, the set of blocking directions is constructed on the unit sphere $\mathcal{S}^2$ by drawing the boundary arcs of the union of the blocking directions (each arc is a portion of a great circle). The resulting collection of arcs partitions $\mathcal{S}^2$ into maximal regions such that the blocking relation among the parts is the same for any direction inside such a region.

Next, the blocking graph is computed for one such maximal region. The algorithm then moves to an adjacent region and updates the DBG by the blocking relations that change at the boundary between the regions, and so on. After each time the construction of a DBG is completed, this graph is checked for strong connectivity in time linear in its number of edges. The algorithm stops the first time it encounters a DBG that is not strongly connected and it outputs the two subassemblies of the partitioning. The overall sequencing algorithm continues recursively with the resulting subassemblies. If all the DBG's that are produced during a partitioning step are strongly connected, the algorithm reports that the assembly does not admit a two-handed monotone assembly sequence with infinite translations.

Polynomial-time algorithms are proposed in [WL95] to compute and exploit NDBG's for restricted families of motions. In particular, the case of partitioning a polyhedral assembly by a single translation to infinity is analyzed in detail, and it is shown that partitioning an assembly of m polyhedra with a total of v vertices takes $O(m^2v^4)$ time. Another case studied in [WL95] is where the separating motions are infinitesimal rigid motions. Then partitioning the polyhedral assembly takes $O(mc^5)$ time, where m is the number of pairs of parts in contact and c is the number of independent point-plane contact constraints. (This result is improved in [GHH+95] by using the concept of maximally covered cells; see Section 21.6.) Using these algorithms, every feasible disassembly sequence can be generated in polynomial time.

In [WL95], NDBG's are defined only for simple families of separating motions (infinitesimal rigid motions and infinite translations). An extension, called the ***interference diagram***, is proposed in [WKL+95] for more complex motions. In the worst case, however, this diagram yields a partitioning algorithm that is exponential in the number of surfaces describing the assembly. When each separating motion is restricted to be a short sequence of concatenated translations (for example, a finite translation followed by an infinite translation), rather efficient partitioning algorithms are available [HW95].

OPEN PROBLEM

The complexity of the NDBG grows exponentially with the number of parameters that control the allowable motions, making this approach highly time consuming for assembly sequencing with compound motions. For the case of infinitesimal rigid motion it has been observed that only a (relatively small) subset of the NDBG needs to be constructed [GHH+95]. Are there additional types of motion where similar gains can be made? Are there situations where the full NDBG (or a structure of comparable size) must be constructed?

41.4 PATH PLANNING

Motion planning is aimed at providing robots with the capability of deciding automatically which motions to execute in order to achieve goals specified by spatial arrangements of physical objects. It arises in a variety of forms. The simplest form—the *basic path planning problem*—requires finding a geometric collision-free path for a single robot in a known static workspace. The path is represented by an arc connecting two points in the robot's configuration space [LP83]. This arc must not intersect a forbidden region, the *C-obstacle region*, which is the image of the workspace obstacles. Other motion planning problems require dealing with moving obstacles, multiple robots, movable objects, uncertainty, etc.

In this section we consider basic path planning. In the next one we will review other motion planning problems. Most of our presentation focuses on results that can be applied to "real" robots. See Chapter 40 for basic theory on algorithmic motion planning.

GLOSSARY

Path: A continuous map $\tau : [0, 1] \to \mathcal{C}$.

Obstacle: A workspace $W \subset \mathbb{R}^k$ is often defined by a set of obstacles B_i, $i = 1, \ldots, q$, such that $W = \mathbb{R}^k \setminus \bigcup_1^q B_i$.

C-obstacle: Given an obstacle B_i, the subset $CB_i \subseteq \mathcal{C}$ such that, for any $\boldsymbol{q} \in CB_i$, $A(\boldsymbol{q})$ intersects B_i.

C-obstacle region: The union $CB = \cup_i CB_i$ plus the configurations that violate the mechanical limits of the robot's joints.

Free space: The complement of the C-obstacle region in $\mathcal{C}$, i.e., $\mathcal{C} \setminus CB$.

Free path: A path in free space.

Semifree path: A path in the closure of free space.

Basic path planning problem: Compute a free or semifree path between two input configurations.

Path planning query: Given two points in configuration space find a (semi)free path between them. The term is often used in connection with algorithms that preprocess the configuration space in preparation for many queries.

Complete algorithm: A motion planning algorithm is complete if it is guaranteed to find a (semi)free path between two given configurations whenever such a path exists, and report that there is no (semi)free path otherwise. Complete algorithms are sometimes referred to as ***exact*** algorithms. There are weaker variants of completeness, for example, *probabilistic completeness.*

COMPLETE ALGORITHMS

Basic path planning for a three-dimensional linkage made of polyhedral links is PSPACE-hard (Theorem 40.1.3c). The proof provides strong evidence that any complete algorithm will require exponential time in the number of dofs. This result remains true in more specific cases, e.g., when the robot is a planar arm in which all joints are revolute (Theorem 40.1.3b). However, it no longer holds in some very simple settings; for instance, planning the path of a planar arm within an empty circle is in P [HJW85].

Two kinds of complete planners have been proposed: general ones, which apply to virtually any robot with an arbitrary number of dofs, and specific ones, which apply to a restricted family of robots usually having a fixed small number of dofs. The general "roadmap" algorithm in [Can88] is singly-exponential in the dimension of $\mathcal{C}$ and polynomial in both the number of polynomial constraints defining the free space and their maximal degree (Theorem 40.1.2). Specific algorithms have been developed mainly for robots with 2 or 3 dofs. For a k-sided polygonal robot moving freely in a polygonal workspace, the algorithm in [HS96] takes $O((kn)^{2+\epsilon})$ time, where n is the total number of edges of the workspace (Theorem 40.2.10).

PROBABILISTIC ALGORITHMS

The complexity of path planning for robots with many dofs (more than 4 or 5) has led to the development of computational schemes that attempt to trade off completeness against time. One such scheme, ***probabilistic planning*** [BKL$^+$], avoids computing an explicit geometric representation of the free space. Instead, it uses an efficient procedure to compute distances between bodies in the workspace. It samples the configuration space by selecting a large number of configurations at random and retaining only the free configurations as *milestones.* It then checks if each pair of milestones can be connected by a collision-free straight path in configuration space. This computation yields the graph (V, E), called a *probabilistic roadmap*, where V is the set of milestones and E is the set of pairs of milestones that have been connected.

Various strategies can be applied to sample the configuration space. The strategy in [KSLO96] proceeds as sketched above. Once a roadmap has been precomputed, it is used to process an arbitrary number of path planning queries. Another sampling strategy [BL91] assumes that the initial and goal configurations are given, and incrementally builds a roadmap until these two configurations are connected.

The results reported in [BKL$^+$] bound the number of milestones generated by the algorithm given in [KSLO96], under the assumption that the configuration space satisfies some simple geometric property. One such property is ***ϵ-goodness***: a set S of volume μ is said to be ϵ-good if every point in S sees a subset of S of volume at least $\epsilon \times \mu$. Under such an assumption, the number of milestones needed

to correctly answer path planning queries with probability $1 - \alpha$ is proportional to $(1/\epsilon)(\log(1/\epsilon) + \log(1/\alpha))$.

Applications of randomized path planning include the maintenance of aircraft engines and the riveting of aircraft fuselages.

HEURISTIC ALGORITHMS

Several heuristic techniques have been proposed to speed up path planning. Some of them work well in practice, but they usually offer no performance guarantee.

Heuristic algorithms often search a regular grid defined over configuration space and generate a path as a sequence of adjacent grid points [Don87]. The search can be guided by a ***potential field***, a function over the free space that has a global minimum at the goal configuration. This function may be constructed as the sum of an attractive and a repulsive field [Kha86]. The attractive field has a single minimum at the goal and grows to infinity as the distance to the goal increases. The repulsive field is null at all configurations where the distance between the robot and the obstacles is greater than some predefined value, and grows to infinity as the robot gets closer to an obstacle. Evaluating the repulsive field requires an efficient distance computation algorithm. The search is usually done in a best-first fashion, by following the steepest descent of the potential function. Several techniques deal with local minima [BL91]. Potentials free of local minima have been proposed [RK92], but their computation is likely to be at least as expensive as path planning.

One may also construct grids at variable resolution. Hierarchical space decomposition techniques such as octrees and boxtrees have been used to that purpose [BH95]. At any decomposition level, each grid cell is labeled EMPTY, FULL, or MIXED depending on whether it lies entirely in the free space, lies in the C-obstacle region, or overlaps both. Only the MIXED cells are decomposed further, until a search algorithm finds a sequence of adjacent FREE cells connecting the initial and goal configurations.

DISTANCE COMPUTATION

The efficient computation of distances between two bodies is a crucial element of many path planners. Various algorithms have been proposed to compute distances between two convex bodies. A numerical descent technique is described in [GJK88] to compute the distance between two convex polyhedra; experience indicates that it runs in approximately linear time in the total complexity of the polyhedra. See Chapter 33 for related techniques.

In robotics applications one often needs to compute the minimum distance between two sets of bodies, one representing the robot, the other the obstacles. The cost of computing the distance between every pair of bodies can be prohibitive. Simple bounding volumes, often coupled with hierarchical decomposition techniques, have been used to reduce computation time [Qui94]. When motion is involved, incremental distance computation has been suggested for tracking the closest points on a pair of convex polyhedra [LC91]. It takes advantage of the fact that the closest features (faces, edges, vertices) change infrequently as the polyhedra move along finely discretized paths.

41.5 OTHER MOTION PLANNING PROBLEMS

There are many useful extensions of the basic path planning problem. Several are surveyed in Chapter 40, e.g., shortest paths, coordinated motion planning (multi-robot case), time-varying workspaces (moving obstacles), and exploratory motion planning. Below we focus on the following extensions: manipulation planning, nonholonomic robots, uncertainty, and optimal planning.

GLOSSARY

Movable object: Body that can be grasped and moved by a robot.

Manipulation planning: Motion planning with movable objects.

Trajectory: Path parametrized by time.

Tangent space: Given a smooth manifold M and a point $p \in M$, the vector space $T_p(M)$ spanned by the tangents at p to all smooth curves passing through p and contained in M. The tangent space has the same dimension as M.

Nonholonomic robot: Robot whose permissible velocities at every configuration $\boldsymbol{q}$ span a subset $\Omega(\boldsymbol{q})$ of the tangent space $T_{\boldsymbol{q}}(\mathcal{C})$ of lower dimension. Ω is called the ***set of controls*** of the robot.

Feasible path: A piecewise differentiable path of a nonholonomic robot whose tangent at every point belongs to the robot's set of controls, i.e., satisfies the nonholonomic velocity constraints.

Locally controllable robot: A nonholonomic robot is locally controllable if for every configuration $\boldsymbol{q}_0$ and any configuration $\boldsymbol{q}_1$ in a neighborhood U of $\boldsymbol{q}_0$, there exists a feasible path connecting $\boldsymbol{q}_0$ to $\boldsymbol{q}_1$ which is entirely contained in U.

Uncertainty in control and sensing: Distributions of control and position sensing errors over multiple executions.

Preimage: A region of configuration space from which a motion command causes the robot to reach and stop in the goal region.

Landmark: Workspace feature that the robot may reliably sense and use to precisely localize itself. The region of configuration space from which the robot can sense a landmark is called a ***landmark area.***

Kinodynamic planning: Find a minimal-time trajectory between two given configurations of a robot, given the robot's dynamic equation of motion.

41.5.1 MANIPULATION PLANNING

Many robot tasks consist of achieving arrangements of physical objects. Such objects, called movable objects, cannot move autonomously; they must be grasped by a robot. Planning with movable objects is called manipulation planning.

In [Wil91] the robot A and the movable object M are both convex polygons in a polygonal workspace. The goal is to bring A and M to specified positions. A can only translate. To grasp M, A must have one of its edges that exactly coincides with an edge of M. While A grasps M, they move together as one rigid

object. An exact cell decomposition algorithm is given that runs in $O(n^2)$ time after $O(n^3 \log^2 n)$ preprocessing, where n is the total number of edges in the workspace, the robot, and the movable object. An extension of this problem allowing an infinite set of grasps is solved by an exact cell decomposition algorithm in [ALS95].

Heuristic algorithms have also been proposed. The planner in [KL94] first plans the path of the movable object M. During that phase, it verifies only that for every configuration taken by M there exists at least one collision-free configuration of the robot where it can hold M. In the second phase, the planner determines the points along the path of M where the robot must change grasps. It then computes the paths where the robot moves alone (transit paths) to (re)grasp M. The paths of the robot when it carries M (transfer paths) are obtained through inverse kinematics. This planner is not complete, but it has solved complex tasks in practice.

41.5.2 NONHOLONOMIC ROBOTS

The trajectories of a nonholonomic robot are constrained by $p \geq 1$ nonintegrable scalar equality constraints:

$$G(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t)) = (G^1(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t)), \cdots, G^p(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))) = (0, \ldots, 0),$$

where $\dot{\boldsymbol{q}}(t) \in T_{\boldsymbol{q}(t)}(\mathcal{C})$ designates the velocity vector along the trajectory $\boldsymbol{q}(t)$. At every $\boldsymbol{q}$, the function $G_{\boldsymbol{q}} = G(\boldsymbol{q}, .)$ maps the tangent space $T_{\boldsymbol{q}}(\mathcal{C})$ into $\mathbb{R}^p$. If $G_{\boldsymbol{q}}$ is smooth and its Jacobian has full rank (two conditions that are often satisfied), the constraint $G_{\boldsymbol{q}}(\dot{\boldsymbol{q}}) = (0, \ldots, 0)$ constrains $\dot{\boldsymbol{q}}$ to be in a linear subspace of $T_{\boldsymbol{q}}(\mathcal{C})$ of dimension $m - p$. The nonholonomic robot may also be subject to scalar inequality constraints of the form $H^j(\boldsymbol{q}, \dot{\boldsymbol{q}}) > 0$. The subset of $T_{\boldsymbol{q}}(\mathcal{C})$ that satisfies all the constraints on $\dot{\boldsymbol{q}}$ is called the set $\Omega(\boldsymbol{q})$ of controls at $\boldsymbol{q}$. A feasible path is a piecewise differentiable path whose tangent lies everywhere in the control set.

A car-like robot is a classical example of a nonholonomic robot. It is constrained by one equality constraint (the linear velocity points along the car's main axis). Limits on the steering angle impose two inequality constraints. Other nonholonomic robots include tractor-trailers, airplanes, and satellites.

Given an arbitrary subset $U \subset \mathcal{C}$, the configuration $\boldsymbol{q}_1 \in U$ is said to be *U-accessible* from $\boldsymbol{q}_0 \in U$ if there exists a piecewise constant control $\dot{\boldsymbol{q}}(t)$ in the control set whose integral is a trajectory joining $\boldsymbol{q}_0$ to $\boldsymbol{q}_1$ that lies fully in U. Let $A_U(\boldsymbol{q}_0)$ be the set of configurations U-accessible from $\boldsymbol{q}_0$. The robot is said to be ***locally controllable*** at $\boldsymbol{q}_0$ iff for every neighborhood U of $\boldsymbol{q}_0$, $A_U(\boldsymbol{q}_0)$ is also a neighborhood of $\boldsymbol{q}_0$. It is locally controllable iff this is true for all $\boldsymbol{q}_0 \in \mathcal{C}$. Car-like robots and tractor-trailers that can go forward and backward are locally controllable [BL93].

Let X and Y be two smooth vector fields on $\mathcal{C}$. The Lie bracket of X and Y, denoted by $[X, Y]$, is the smooth vector field on $\mathcal{C}$ defined by $[X, Y] = dY \cdot X - dX \cdot Y$, where dX and dY, respectively, denote the $m \times m$ matrices of the partial derivatives of the components of X and Y w.r.t. the configuration coordinates in a chart placed on $\mathcal{C}$. To get a better intuition of the Lie bracket, imagine a trajectory starting at an arbitrary configuration $\boldsymbol{q}_s$ and obtained by concatenating four subtrajectories: the first is the integral curve of X during time δt; the second, third, and fourth are the integral curves of Y, $-X$, and $-Y$, respectively, each during the same δt. Let

$\boldsymbol{q}_f$ be the final configuration reached. A Taylor expansion yields:

$$\lim_{\delta t \to 0} \frac{\boldsymbol{q}_f - \boldsymbol{q}_s}{\delta t^2} = [X, Y].$$

Hence, if $[X, Y]$ is not a linear combination of X and Y, the above trajectory allows the robot to move away from $\boldsymbol{q}_s$ in a direction that is not contained in the vector subspace defined by $X(\boldsymbol{q}_s)$ and $Y(\boldsymbol{q}_s)$. But the motion along this new direction is an order of magnitude slower than along any direction $\alpha X(\boldsymbol{q}_s) + \beta Y(\boldsymbol{q}_s)$.

The ***control Lie algebra*** associated with the control set Ω, denoted by $L(\Omega)$, is the space of all linear combinations of vector fields in Ω closed by the Lie bracket operation. The following result derives from the Controllability Rank Condition Theorem [BL93]:

A robot is locally controllable if, for every $\boldsymbol{q} \in \mathcal{C}$, $\Omega(\boldsymbol{q})$ is symmetric with respect to the origin of $T_{\boldsymbol{q}}(\mathcal{C})$ and the set $\{X(\boldsymbol{q}) \mid X \in L(\Omega(\boldsymbol{q}))\}$ has dimension m.

The minimal length of the Lie brackets required to construct $L(\Omega)$, when these brackets are expressed with vectors in Ω, is called the ***degree of nonholonomy*** of the robot. The degree of nonholonomy of a car-like robot is 2. Except at some singular configurations, the degree of nonholonomy of a tractor towing a chain of s trailers is $2 + s$ [LR96]. Intuitively, the higher the degree of nonholonomy, the more complex (and the slower) the robot's maneuvers to perform some motions.

PLANNING FOR CONTROLLABLE ROBOTS

Let A be a locally controllable nonholonomic robot. A necessary and sufficient condition for the existence of a feasible free path of A between two given configurations is that they lie in the same connected component of the *open* free space. Indeed, local controllability guarantees that a possibly nonfeasible path can be decomposed into a finite number of subpaths, each short enough to be replaced by a feasible free subpath. Hence, deciding if there exists a free path for a locally controllable nonholonomic robot has the same complexity as deciding if there exists a path for the holonomic robot having the same geometry.

Transforming a nonfeasible free path τ into a feasible one can be done by recursively decomposing τ into subpaths. The recursion halts at every subpath that can be replaced by a feasible free subpath. Specific substitution rules (e.g., Reeds and Shepp curves) have been defined for car-like robots [LJTM94]. The complexity of transforming a nonfeasible free path τ into a feasible one is of the form $O(\epsilon^d)$, where ϵ is the smallest clearance between the robot and the obstacles along τ and d is the degree of nonholonomy of the robot (see [LJTM94] for the case $d = 2$).

The algorithm in [BL93] directly constructs a nonholonomic path for a car-like or a tractor-trailer robot by searching a tree obtained by concatenating short feasible paths, starting at the robot's initial configuration. The planner is ***asymptotically complete***, i.e., it is guaranteed to find a path if one exists, provided that the length of the short feasible paths are small enough. It can also find paths that minimize the number of cusps (changes of sign of the linear velocity).

PLANNING FOR NONCONTROLLABLE ROBOTS

Path planning for nonholonomic robots that are not locally controllable is much

less understood. Research has almost exclusively focused on car-like robots that can only move forward. Results include:

- No obstacles: A complete synthesis of the shortest, no-cusp path for a point moving with a lower-bounded turning radius [SL93].
- Polygonal obstacles: An algorithm to decide whether there exists such a path between two configurations; it runs in time exponential in obstacle complexity [FW88].
- Convex obstacles: The algorithm in [ART95] computes a path in polynomial time under the assumptions that all obstacles are convex and their boundaries have a curvature radius greater than the minimum turning radius of the point (***moderate*** obstacles).
- Other polynomial algorithms (e.g., [BL93]) require some sort of discretization and are only asymptotically complete.

OPEN PROBLEM

Establish a nontrivial lower bound on the complexity of planning for a nonholonomic robot that is not locally controllable.

41.5.3 UNCERTAINTY

In practice, robots deviate from planned paths due to errors in control and position sensing. A motion planning problem with uncertainty can be formulated as follows:

Input. The inputs are the initial region $I \subset \mathcal{C}$, in which the robot is known to be prior to moving, the goal region $G \subset \mathcal{C}$, in which it should terminate its motion, and the uncertainty in control and sensing. Uncertainty is specified in the form of regions. For instance, the uncertainty in position sensing is the set of actual robot configurations that are possible given the sensor readings.

Output. The output is a series of motion commands, if one exists, whose execution enables the robot to reach G from I. Each command is described by a velocity vector $\boldsymbol{v}$ and a termination condition T. The vector $\boldsymbol{v}$ specifies the desired behavior of the robot over time (with or without compliance). The condition T is a Boolean function of the sensor readings and time which causes the motion to stop as soon as it becomes true. A plan may contain conditional branchings.

This problem is NEXPTIME-hard for a point robot moving in 3-space among polyhedral obstacles [CR87].

PREIMAGE OF A GOAL

Given a goal G and a command $(\boldsymbol{v}, T)$, a preimage of G is any region $P \subset \mathcal{C}$ such that executing the command from anywhere in P makes the robot reach and stop in G [LPMT84].

One way to compute a (nonmaximal) preimage is to restrict the termination condition so that it recognizes G independently of the region from which the motion started [Erd86]. For example, one may shrink G to a subset K, called the ***kernel***

of G, such that whenever the robot is in K, all robot configurations consistent with the current sensor readings are in G. A preimage is then computed as the region from which the robot commanded along $\boldsymbol{v}$ is guaranteed to reach K. This region is called the ***backprojection*** of K for $\boldsymbol{v}$. This preimage computation approach has been well studied in a polygonal configuration space when G is a polygon [Lat91].

ONE-STEP PLANNING

In a polygonal configuration space, the kernel of a polygonal goal is either independent of the selected $\boldsymbol{v}$ or changes at a number of critical orientations of $\boldsymbol{v}$ that is linear in the workspace complexity [Lat91]. Moreover, the backprojection of a polygonal region K, when the orientation of $\boldsymbol{v}$ varies, changes topology only at a quadratic number of critical directions. Its intersection with a polygonal initial region I of constant complexity also changes qualitatively at few directions of $\boldsymbol{v}$. Checking the containment of I by the backprojection at each such direction yields a one-step motion plan, if one exists, in amortized time $O(n^2 \log n)$, where n is the number of edges in $\mathcal{C}$ [Bri95]. In [dBG+95] the computational complexity of solving certain one-step planning problems is expressed also in terms of the size of the control error.

MULTI-STEP PLANNING

For multi-step planning, algebraic approaches that check the satisfiability of a first-order semi-algebraic formula have been proposed. In [Can89] it is assumed that all possible trajectories have an algebraic description. The approach there is based on a two-player game interpretation of planning, where the robot is one player and nature the other. Each step of a plan contributes three quantifiers: one existential quantifier applies to the direction of motion, and corresponds to choosing this direction; another existential quantifier applies to time, and corresponds to choosing when to terminate the motion; one universal quantifier applies to the sensor readings and represents the unknown action of nature. The formula representing an r-step plan thus contains r quantifier alternations; checking its satisfiability takes doubly-exponential time in r, which is itself polynomial in the total complexity of the robot and the workspace.

LANDMARK-BASED PLANNING

Often a workspace contains features that can be reliably sensed and used to precisely localize the robot. Each such landmark feature induces a region in configuration space called the *landmark area* from which the robot can sense the corresponding feature.

The planner described in [LL95] considers a point robot among n circular obstacles and $O(n)$ circular landmark areas. It assumes perfect position sensing and motion control in landmark areas. Outside these areas, it assumes that the robot has no position sensing whatsoever and that directional errors in control are bounded by the angle θ. Given circular initial and goal regions I and G (with G intersecting at least one landmark area), the planner constructs a motion plan that enables the robot to move from landmark area to landmark area until it reaches the goal. It proceeds backward by computing the preimages of the landmark regions intersecting G, the preimages of the landmark regions intersected by these preimages, and so on, until a preimage contains I. The planner runs in $O(n^4 \log n)$ time; it is

complete and generates plans that minimize the number of steps to be executed in the worst case.

41.5.4 OPTIMAL PLANNING

There has been considerable research on finding shortest paths (see Chapter 24), but minimal Euclidean length may not be the most suitable criterion in practice. One is often more interested in minimizing execution time, which requires dealing with the robot's dynamics.

OPTIMAL-TIME CONTROL PLANNING

The input is a (geometric) free path τ parameterized by $s \in [0, L]$, the distance traveled from the starting configuration. The problem is to find the time parametrization $s(t)$ that minimizes travel time along τ, while satisfying actuator limits.

The equation of motion of a robot arm with m dofs can be written as $M(\boldsymbol{q})\ddot{\boldsymbol{q}} + V(\dot{\boldsymbol{q}}, \boldsymbol{q}) + G(\boldsymbol{q}) = \Gamma$, where $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, and $\ddot{\boldsymbol{q}}$ respectively denote the robot's configuration, velocity, and acceleration [Cra86]. M is the $m \times m$ inertia matrix of the robot, V the m-vector (quadratic in $\dot{\boldsymbol{q}}$) of centrifugal and Coriolis forces, and G the m-vector of gravity forces. Γ is the m-vector of the torques applied by the joint actuators.

Using the fact that the robot follows τ, this equation can be rewritten in the form: $\boldsymbol{m}\ddot{s} + \boldsymbol{v}\dot{s}^2 + \boldsymbol{g} = \Gamma$, where $\boldsymbol{m}$, $\boldsymbol{v}$, and $\boldsymbol{g}$ are derived from M, V, and G, respectively. Minimum-time control planning becomes a two-point boundary value problem: Find $s(t)$ that minimizes $t_f = \int_0^L ds/\dot{s}$, subject to $\Gamma = \boldsymbol{m}\ddot{s} + \boldsymbol{v}\dot{s}^2 + \boldsymbol{g}$, $\Gamma_{min} \leq \Gamma \leq \Gamma_{max}$, $s(0) = 0$, $s(t_f) = L$, and $\dot{s}(0) = \dot{s}(L) = 0$. Numerical techniques solve this problem by finely discretizing the path τ [BDG85].

MINIMAL-TIME TRAJECTORY PLANNING

Finding a minimal-time trajectory, called *kinodynamic motion planning*, is much harder. One approach is to first plan a geometric free path and then iteratively deform this path to reduce travel time [SD91]. Each iteration requires checking the new path for collision and recomputing the optimal-time control. No bound has been established on the running time of this approach or the goodness of its outcome. Kinodynamic planning is NP-hard for a point robot under Newtonian mechanics in 3-space [DX95]. The approximation algorithm in [DXCR93] computes a trajectory ϵ-close to optimal in time polynomial in both $1/\epsilon$ and the workspace complexity.

41.6 DATA STRUCTURES FOR MOVING OBJECTS

Robotics requires efficient algorithms to compute motions and/or to update properties of bodies as they move (e.g., distances to obstacles). Several data structures have been specifically proposed to represent moving bodies.

NONDIRECTIONAL DATA STRUCTURES

These data structures partition the space of possible motions into an arrangement of

cells such that a given property remains satisfied over each cell. They are typically computed in a preprocessing step to speed up the treatment of subsequent queries.

For example, in the context of assembly sequencing (Section 41.3), a property of interest is how the parts in an assembly block one another for a certain family of motions. It yields the concepts of a nondirectional blocking graph and an interference diagram. In motion planning with uncertainty (Section 41.5.3), a similar concept is the nondirectional backprojection/preimage of a goal [Bri95, LL95]. As the direction of motion varies, the topology of a preimage changes only at critical values which define an arrangement of cells in the motion space. This arrangement, along with a preimage computed in each cell, forms the ***nondirectional preimage.***

A related concept is used in [Gol93] to construct the possible orientations of a polygonal body after it has been squeezed by a parallel-jaw gripper (Section 41.2.3).

DYNAMIC MAINTENANCE OF KINEMATIC STRUCTURES

Several prototypes of highly flexible robots have been designed and constructed in recent years. Since the number of dofs in these new designs is far larger than in more traditional robots, they raise new algorithmic issues. Similar issues arise in computer simulation of large kinematic structures outside robotics, e.g., in molecular biology and in graphic animation of digital actors.

A basic problem in this domain can be phrased as follows. Given a linkage with many dofs, how can we efficiently maintain a data structure that allows us to quickly answer intersection (or range) queries as the bodies move. Several models for dynamic maintenance of such linkages are proposed in [HLM97], together with efficient maintenance algorithms. Tight results are given on the worst-case, amortized, and randomized complexity of this data structure problem. For the offline version of the problem, NP-hardness is established and efficient approximation algorithms are provided.

41.7 SENSING

Sensing allows a robot to acquire information about its workspace and to localize itself. A wide variety of sensors are available and provide raw data of different types, such as time of flight, light intensity, color, or force. Preprocessing these data yield more directly usable information, e.g., geometric information, which can then be exploited to perform such tasks as model construction, object identification, and robot localization. Vision sensors are the most widely used sensors. Many recent textbooks focus on the role of geometry in computer vision, e.g., [Gri90]. Touch and force sensors are important to detect and characterize contacts among objects, for instance in manipulation tasks. Sensing is a wide domain of research with many subareas and challenging problems. Here we mention only a few selected topics.

MODEL BUILDING

Consider a mobile robot in an unknown workspace W. A first task for this robot is likely to be the construction of a geometric model (also called a map) of W. This

requires the robot to perform a series of sensing operations at different locations. Each operation yields a partial model. The robot must patch together the successively obtained partial models to eventually form a complete map of the workspace. This problem is complicated by the fact that the robot has imperfect control and cannot accurately keep track of its position in a fixed coordinate system. See, e.g., [ZF96].

ROBOT LOCALIZATION

A robot often has to localize itself relative to its workspace W. A model of W is given and localization is done by matching sensory inputs against this model to infer the transform that defines the robot configuration. This problem usually arises for mobile robots. Other types of robots, such as robot arms, often have absolute references (e.g., mechanical stops) and internal sensors (e.g., joint encoders) that provide configurations more directly. Mobile robots have wheel encoders allowing dead-reckoning, but the absence of absolute reference on the one hand and slipping on the ground on the other hand usually necessitate sensor-based localization. GPS (Global Positioning System) has recently become a more widely available alternative, but it does not work in all environments.

Two kinds of sensor-based localization problems can be distinguished, *static* and *dynamic*. In the static problem, the robot is placed at an arbitrary unknown configuration and the problem is to compute this configuration. In the dynamic problem, the robot moves continuously and must regularly update its configuration. The second problem consists of refining an available estimate of the current configuration; here the computation must be done in real time. The static problem is usually more complex, but computation time is less restricted. A preprocessing approach to the static localization problem for a point robot equipped with a 360° range sensor is discussed in Section 40.3. Practical techniques for localization are also available, e.g., [TA96].

ADDITIONAL ISSUES IN SENSING

Sensor placement is the problem of computing the set of placements from which a sensor can monitor a region within a given workspace [Bri95]. Another problem is to choose a minimal set of sensors and their placement so as to completely cover a given region. This induces a family of art-gallery type problems (see Section 25.1) that vary according to the type of data that the sensor provides.

There has been considerable interest in reconstructing shapes of objects using simple sensors called *probes*; see Chapter 26. *Matching* and *aspect graphs* (Section 25.6.3) are two related topics that have been well studied, mainly in computer vision.

41.8 SOURCES AND RELATED MATERIAL

Craig's book [Cra86] provides an introduction to robot arm kinematics, dynamics, and control. For advanced kinematics see the book by Bottema and Roth [BR79].

Robot motion planning and its variants are discussed in Latombe's book [Lat91]. This book takes an algorithmic approach to a variety of advanced issues in robotics (not restricted to robot arms).

The proceedings series of the *International Symposium on Robotics Research* gives state-of-the-art presentations of robotics in general [GH96], whereas the recent collection [GHLW95] emphasizes algorithmic issues in robotics.

Several computational geometry books contain sections on robotics or motion planning [O'R94, SA95].

RELATED CHAPTERS

Chapter 21: Arrangements
Chapter 25: Visibility
Chapter 26: Geometric reconstruction problems
Chapter 29: Computational real algebraic geometry
Chapter 40: Algorithmic motion planning
Chapter 48: Geometric applications of the Grassmann-Cayley algebra

REFERENCES

[AHLM96] S. Akella, W. Huang, K. Lynch, and M. Mason. Planar manipulation on a conveyor with a one joint robot. In [GH96], pages 265–276.

[ALS95] R. Alami, J.P. Laumond, and T. Siméon. Two manipulation planning algorithms. In [GHLW95], pages 109–125.

[ART95] P.K. Agarwal, P. Raghavan, and H. Tamaki. Motion planning for a steering-constrained robot through moderate obstacles. In *Proc. 28th Annu. ACM Sympos. Theory Comput.*, pages 343–352, 1995.

[BDG85] J.E. Bobrow, S. Dubowsky, and J.S. Gibson. Time-optimal control of robotic manipulators along specified paths. *Internat. J. Robot. Res.*, 4:3–17, 1985.

[BDM96] K.F. Böhringer, B.R. Donald, and N.C. MacDonald. What programmable vector fields can (and cannot) do: Force field algorithms for MEMS and vibratory parts feeders. In *Proc. 13th IEEE Internat. Conf. Robot. Autom.*, pages 822–930, 1996.

[BG96] R.C. Brost and K.Y. Goldberg. A complete algorithm for designing planar fixtures using modular components. *IEEE Trans. Syst. Man Cybern.*, 12:31–46, 1996.

[BH95] M. Barbehenn and S. Hutchinson. Efficient search and hierarchical motion planning by dynamically maintaining single-source shortest paths trees. *IEEE Trans. Robot. Autom.*, 11:198–214, 1995.

[BKL+] J. Barraquand, L.E. Kavraki, J.C. Latombe, T.-Y. Li, R. Motwani, and P. Raghavan. A random sampling framework for path planning in large-dimensional configuration spaces. *Internat. J. Robot. Res.*, to appear.

[BL91] J. Barraquand and J.C. Latombe. Robot motion planning: A distributed representation approach. *Internat. J. Robot. Res.*, 10:628–649, 1991.

[BL93] J. Barraquand and J.C. Latombe. Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles. *Algorithmica*, 10:121–155, 1993.

[BR79] O. Bottema and B. Roth. *Theoretical Kinematics.* North Holland, Amsterdam, 1979.

[Bri95] A.J. Briggs. Efficient geometric algorithms for robot sensing and control. Report 95-1480, Dept. of Computer Science, Cornell Univ., 1995.

[Can88] J.F. Canny. *The Complexity of Robot Motion Planning.* MIT Press, Cambridge, 1988.

[Can89] J.F. Canny. On computability of fine motion plans. In *Proc. 6th IEEE Internat. Conf. Robot. Autom.*, pages 177–182, 1989.

[CR87] J.F. Canny and J. Reif. New lower bound techniques for robot motion planning problems. In *Proc. 28th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 49–60, 1987.

[Cra86] J.J. Craig. *Introduction to Robotics. Mechanics and Control.* Addison-Wesley, Reading, 1986.

[dBG+95] M. de Berg, L. Guibas, D. Halperin, M. Overmars, O. Schwarzkopf, M. Sharir, and M. Teillaud. Reaching a goal with directional uncertainty. *Theoret. Comput. Sci.*, 140:301–317, 1995.

[Don87] B.R. Donald. A search algorithm for motion planning with six degrees of freedom. *Artif. Intell.*, 31:295–353, 1987.

[DX95] B.R. Donald and P. Xavier. Provably good approximation algorithms for optimal kinodynamic planning: Robots with decoupled dynamics bounds. *Algorithmica*, 14:443–479, 1995.

[DXCR93] B.R. Donald, P. Xavier, J.F. Canny, and J. Reif. Kinodynamic motion planning. *J. Assoc. Comput. Mach.*, 40:1048–1066, 1993.

[EM88] M. Erdmann and M. Mason. An exploration of sensorless manipulation. *IEEE Trans. Robot. Autom.*, 4:369–379, 1988.

[Erd86] M. Erdmann. Using backprojections for fine motion planning with uncertainty. *Internat. J. Robot. Res.*, 5:19–45, 1986.

[FC92] C. Ferrari and J.F. Canny. Planning optimal grasps. In *Proc. 9th IEEE Internat. Conf. Robot. Autom.*, pages 2290–2295, 1992.

[FW88] S.J. Fortune and G.T. Wilfong. Planning constrained motions. In *Proc. 21st Annu. ACM Sympos. Theory Comput.*, pages 445–459, 1988.

[GH96] G. Giralt and G. Hirzinger, editors. *Robotics Research.* Springer, New York, 1996.

[GHH+95] L. Guibas, D. Halperin, H. Hirukawa, J.C. Latombe, and R.H. Wilson. A simple and efficient procedure for polyhedral assembly partitioning under infinitesimal motions. In *Proc. 12th IEEE Internat. Conf. Robot. Autom.*, pages 2553–2560, 1995.

[GHLW95] K.Y. Goldberg, D. Halperin, J.C. Latombe, and R.H. Wilson, editors. *Algorithmic Foundations of Robotics.* A K Peters, Wellesley, 1995.

[GJK88] E.G. Gilbert, D.W. Johnson, and S.S. Keerthi. A fast procedure for computing distance between complex objects in three-dimensional space. *IEEE Trans. Robot. Autom.*, 4:193–203, 1988.

[Gol93] K.Y. Goldberg. Orienting polygonal parts without sensors. *Algorithmica*, 10:201–225, 1993.

[Gri90] W.E.L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints.* MIT Press, Cambridge, 1990.

[HJW85] J.E. Hopcroft, D.A. Joseph, and S.H Whitesides. On the movement of robot arms in 2-dimensional bounded regions. *SIAM J. Computing*, 14:315–333, 1985.

[HLM97] D. Halperin, J.C. Latombe, and R. Motwani. Dynamic maintenance of kinematic structures. In J.P. Laumond and M. Overmars, editors, *Algorithms for Robotic Motion and Manipulation*, A K Peters, Wellesley, pages 155–170, 1997.

[HS91] L.S. Homem de Mello and A.C. Sanderson. A correct and complete algorithm for the generation of mechanical assembly sequences. *IEEE Trans. Robot. Autom.*, 7:228–240, 1991.

[HS96] D. Halperin and M. Sharir. A near-quadratic algorithm for planning the motion of a polygon in a polygonal environment. *Discrete Comput. Geom.*, 16:121–134, 1996.

[HW95] D. Halperin and R.H. Wilson. Assembly partitioning along simple paths: The case of multiple translations. In *Proc. 12th IEEE Internat. Conf. Robot. Autom.*, pages 1585–1593, 1995.

[Kha86] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Internat. J. Robot. Res.*, 5:90–98, 1986.

[KK95] L.E. Kavraki and M.N. Kolountzakis. Partitioning a planar assembly into two connected parts is NP-complete. *Inform. Process. Lett.*, 55:159–165, 1995.

[KL94] Y. Koga and J.C. Latombe. On multi-arm manipulation planning. In *Proc. 11th IEEE Internat. Conf. Robot. Autom.*, pages 945–952, 1994.

[KMY92] D. Kirkpatrick, B. Mishra, and C. Yap. Quantitative Steinitz's theorem with applications to multifingered grasping. *Discrete Comput. Geom.*, 7:295–318, 1992.

[Kol95] K. Kolarov. Algorithms for optimal design of robots in complex environment. In [GHLW95], pages 347–369.

[KSLO96] L.E. Kavraki, P. Svestka, J.C. Latombe, and M. Overmars. Probabilistic roadmaps for fast path planning in high dimensional configuration spaces. *IEEE Trans. Robot. Autom.*, 12:566–580, 1996.

[Lat91] J.C. Latombe. *Robot Motion Planning.* Kluwer, Boston, 1991.

[LC91] M.C. Lin and J.F. Canny. A fast algorithm for incremental distance computation. In *Proc. 8th IEEE Internat. Conf. Robot. Autom.*, pages 1008–1014, 1991.

[LJTM94] J.P. Laumond, P.E. Jacobs, M. Taix, and R.M. Murray. A motion planner for nonholonomic mobile robots. *IEEE Trans. Robot. Autom.*, 10:577–593, 1994.

[LL95] A. Lazanas and J.C. Latombe. Landmark-based robot navigation. *Algorithmica*, 13:472–501, 1995.

[LP83] T. Lozano-Pérez. Spatial planning: A configuration space approach. *IEEE Trans. Comput.*, 32:108–120, 1983.

[LPMT84] T. Lozano-Pérez, M.T. Mason, and R.H. Taylor. Automatic synthesis of fine-motion strategies for robots. *Internat. J. Robot. Res.*, 3:3–24, 1984.

[LR96] J.P. Laumond and J.J. Risler. Nonholonomic systems: controllability and complexity. *Theoret. Comput. Sci.*, 157:101–114, 1996.

[Mas82] M. Mason. *Manipulation by grasping and pushing operations.* PhD thesis, MIT, Artif. Intell. Lab., 1982.

[Mat95] R. Matikalli. *Mechanics Based Assembly Planning.* PhD thesis, Carnegie Mellon Univ., 1995.

[Mis91] B. Mishra. Workholding: Analysis and planning. In *Proc. IEEE/SRJ Internat. Conf. Intell. Robots Syst.*, pages 53–57, 1991.

[Mis95] B. Mishra. Grasp metrics: Optimality and complexity. In [GHLW95], pages 137–165.

[MNP90] X. Markenscoff, L. Ni, and C.H. Papadimitriou. The geometry of grasping. *Internat. J. Robot. Res.*, 9:61–74, 1990.

[MSS87] B. Mishra, J.T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2:541–558, 1987.

[MZG+96] B. Mirtich, Y. Zhuang, K. Goldberg, J. Craig, R. Zanutta, B. Carlisle, and J. Canny. Estimating pose statistics for robotic part feeders. In *Proc. 13th IEEE Internat. Conf. Robot. Autom.*, pages 1140–1146, 1996.

[Nat88] B.K. Natarajan. On planning assemblies. In *Proc. 4th Annu. ACM Sympos. Comput. Geom.*, pages 299–308, 1988.

[Ngu88] V.D. Nguyen. Constructing force-closure grasps. *Internat. J. Robot. Res.*, 7:3–16, 1988.

[O'R94] J. O'Rourke. *Computational Geometry in C.* Cambridge University Press, 1994.

[ORSW95] M. Overmars, A. Rao, O. Schwarzkopf, and C. Wentink. Immobilizing polygons against a wall. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 29–38, 1995.

[PSS+95] J. Ponce, A. Sudsang, S. Sullivan, B. Faverjon, J.D. Boissonnat, and J.P. Merlet. Algorithms for computing force-closure grasps of polyhedral objects. In [GHLW95], pages 167–184.

[Qui94] S. Quinlan. Efficient distance computation between non-convex objects. In *Proc. 11th IEEE Internat. Conf. Robot. Autom.*, pages 3324–3329, 1994.

[RG95] A.S. Rao and K.Y. Goldberg. Manipulating algebraic parts in the plane. *IEEE Trans. Robot. Autom.*, 11:598–602, 1995.

[RK92] E. Rimon and D. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Trans. Robot. Autom.*, 8:501–518, 1992.

[Rot94] B. Roth. Connections between robotic and classical mechanisms. In T. Kanade and R. Paul, editors, *Robotics Research 6*, pages 225–236. The International Foundation for Robotics Research, 1994.

[SA95] M. Sharir and P.K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications.* Cambridge University Press, 1995.

[SD91] Z. Shiller and S. Dubowsky. On computing time-optimal motions of robotic manipulators in the presence of obstacles. *IEEE Trans. Robot. Autom.*, 7:785–797, 1991.

[SL93] P. Souères and J.P. Laumond. Shortest path synthesis for a car-like robot. In *Proc. 2nd European Control Conf.*, 1993.

[SS94] J. Snoeyink and J. Stolfi. Objects that cannot be taken apart with two hands. *Discrete Comput. Geom.*, 12:367–384, 1994.

[TA96] R. Talluri and J.K. Aggarwal. Mobile robot self-location using model-image feature correspondence. *IEEE Trans. Robot. Autom.*, 12:63–77, 1996.

[WGPB96] J. Wiegley, K. Goldberg, M. Peshkin, and M. Brokowski. A complete algorithm for designing passive fences to orient parts. In *Proc. 13th IEEE Internat. Conf. Robot. Autom.*, pages 1133–1139, 1996.

[Wil91] G. Wilfong. Motion planning in the presence of movable objects. *Ann. Math. Artif. Intell.*, 3:131–150, 1991.

[WKL+95] R.H. Wilson, L.E. Kavraki, J.C. Latombe, and T. Lozano-Pérez. Two-handed assembly sequencing. *Internat. J. Robot. Res.*, 14:335–350, 1995.

[WL95] R.H. Wilson and J.C. Latombe. Geometric reasoning about mechanical assembly. *Artif. Intell.*, 71:371–396, 1995.

[ZF96] Z. Zhang and O. Faugeras. A 3d world model builder with a mobile robot. *Internat. J. Robot. Res.*, 11:269–285, 1996.

[ZGW94] Y. Zhuang, K.Y. Goldberg, and Y. Wong. On the existence of modular fixtures. In *Proc. 11th IEEE Internat. Conf. Robot. Autom.*, pages 543–549, 1994.

42 COMPUTER GRAPHICS

David Dobkin and Seth Teller

INTRODUCTION

Computer graphics is often cited as a prime application area for the techniques of computational geometry. The histories of the two fields have a great deal of overlap, with similar methods (e.g., sweep-line and area subdivision algorithms) arising independently in each. Both fields have often focused on similar problems, although with different computational models. For example, hidden surface removal (visible surface identification) is a fundamental problem of computer graphics. This problem has also motivated many researchers in computational geometry. At the same time, as the fields have matured, they have brought different requirements to similar problems. Here, we aim to highlight both similarities and differences between the fields.

Computational geometry is fundamentally concerned with the efficient quantitative representation and manipulation of ideal geometric entities to produce exact results. Computer graphics shares these goals, in part. However, graphics practitioners also model the interaction of objects with light and with each other, and the media through which these effects propagate. Moreover, graphics researchers and practitioners: (1) typically use finite precision (rather than exact) representations for geometry; (2) rarely find closed-form solutions to problems, instead employing sampling strategies and numerical methods; (3) often design into their algorithms explicit tradeoffs between running time and solution quality; and (4) implement most algorithms they propose.

GLOSSARY

Radiometry: The quantitative study of electromagnetic radiation.

Simulation: The representation of a natural process by a computation.

Psychophysics: The study of the human visual system's response to electromagnetic stimuli.

42.1 RELATIONSHIP TO COMPUTATIONAL GEOMETRY

GEOMETRY VS. RADIOMETRY AND PSYCHOPHYSICS

Computer graphics can be formulated as a radiometrically "weighted" counterpart to computational geometry. A fundamental computational process in graphics is ***rendering***: the synthesis of realistic images of physical objects. This is done through the application of a simulation process to quantitative models of light and

materials to predict (i.e., *synthesize*) appearance. Of course, this process must account for the shapes of and spatial relationships between objects, as must computational geometric algorithms. In graphics, however, objects are imbued further with material properties, such as *reflectance* (in its simplest form, color), *refractive index*, *opacity*, and (for light sources) *emissivity*. Moreover, physically justifiable graphics algorithms must model ***radiometry***: quantitative representations of light sources and the electromagnetic radiation they emit (with associated attributes of intensity, wavelength, phase, etc.). All graphics algorithms that produce output intended for viewing must also explicitly or implicitly involve psychophysics.

CONTINUOUS IDEAL VS. DISCRETE REPRESENTATIONS

Computational geometry is largely concerned with ideal objects (points, lines, circles, spheres, hyperplanes, polyhedra), continuous representations (effectively infinite precision arithmetic), and exact combinatorial and algebraic results. Graphics algorithms (and their implementations) model such objects as well, but do so in a discrete, finite-precision computational model. For example, most graphics algorithms use a floating-point or fixed-point coordinate representation. Thus, one can think of most computer graphics computations as occurring on a (2D or 3D) grid. However, a practical difficulty is that the grid spacing is not constant, causing certain geometric predicates (e.g., sidedness) to change under simple transformations such as scaling or translation (see Chapter 35).

An analogy can be made between this distinct choice of coordinates, and the way in which geometric objects—infinite collections of points—are represented by geometers and graphics researchers. Both might represent a sphere similarly—say, by a center and radius. However, an algorithm to render the sphere must select a finite set of "sample" points on its surface. These sample points typically arise from the position of a synthetic camera and from the locations of display elements on a two-dimensional display device, for example pixels on a computer monitor or ink dots on a page in a computer printer. The colors computed at these (zero-area) sample points, through some radiometric computation, then serve as an assignment to the discrete value of each (finite-area) display element.

CLOSED-FORM VS. NUMERICAL SOLUTION METHODS

Rarely does a problem in graphics demand a closed-form solution. Instead, graphicists typically rely on numerical algorithms to estimate solution values in an iterative fashion. Numerical algorithms are chosen by reason of efficiency, or of simplicity. Often, these are antagonistic goals. Aside from the usual dangers of finite-precision arithmetic (Chapter 35), other types of error may arise from numerical algorithms. First, using a point-sampled value to represent a finite-area function's value leads to discretization errors—differences between the reconstructed (interpolated) function, which may be piecewise-constant, piecewise-linear, piecewise-polynomial etc., and the piecewise-continuous (but unknown) true function. These errors may be exacerbated by a poor choice of sampling points, by a poor piecewise function representation or basis, or by neglect of boundaries along which the true function or its derivative have strong discontinuities. Also, numerical algorithms may suffer bias and converge to incorrect solutions (e.g., due to the misweighting, or omission, of significant contributions).

TRADING SOLUTION QUALITY FOR COMPUTATION TIME

The most successful graphics algorithms recognize sources of error and seek to bound them by various means. Moreover, for efficiency's sake an algorithm might deliberately introduce error. For example, during rendering, objects might be crudely approximated to speed the geometric computations involved. Alternatively, in a more general illumination computation, many instances of combinatorial interactions (e.g., reflections) between scene elements might be ignored except when they have a significant effect on the computed image or radiometric values. Graphics practitioners have long sought to exploit this intuitive tradeoff between solution quality and computation time.

THEORY VS. PRACTICE

Graphics algorithms, while often designed with theoretical concerns in mind, are typically intended to be of practical use. Thus, while computational geometers and computer graphicists have a substantial overlap of interest in geometry, graphicists develop computational strategies that can feasibly be implemented on modern machines. Also, while computational geometric algorithms often assume "generic" inputs, in practice geometric degeneracies do occur, and inputs to graphics algorithms are at times highly degenerate (for example, comprised entirely of isothetic rectangles).

Thus, algorithmic strategies are shaped not only by challenging inputs which arise in practice, but also by the technologies available at the time the algorithm is proposed. The relative bandwidths of CPU, bus, memory, network connections, and tertiary storage have major implications for graphics algorithms involving interaction or large amounts of simulation data, or both. For example, in the 1980's the decreasing cost of memory, and the need for robust processing of general datasets, brought about a fundamental shift in most practitioners' choice of computational techniques for resolving visibility (from combinatorial, object-space algorithms to brute force, screen-space algorithms). The increasing power of general-purpose processors, the emergence of sophisticated, robust visibility algorithms, and the wide availability of dedicated low-level graphics hardware may bring about yet another fundamental shift.

TOWARD A MORE FRUITFUL OVERLAP

Given such substantial overlap, there is potential for fruitful collaboration between geometers and graphicists (see [CAA+96] for a recent call for just this kind of collaboration). One mechanism for spurring such collaboration is the careful posing of models and open problems to both communities. To that end, these are interspersed throughout the remainder of this chapter.

42.2 GRAPHICS AS A COMPUTATIONAL PROCESS

This section gives an overview of three fundamental graphics operations: *acquisition* of simulation data; *representation* of such data and associated attributes and

energy sources; and *simulation* of these to predict behavior or appearance. One fundamental simulation process, *rendering*, is partitioned into the subcomponents *visibility* and *shading*, which are treated in separate sections below.

GLOSSARY

Rendering problem: Given quantitative descriptions of surfaces and their properties, light sources, and the media in which all these are embedded, rendering is the application of a computational model to predict appearance; that is, rendering is the synthesis of images from simulation data. Rendering typically involves for each surface a *visibility* computation followed by a *shading* computation.

Visibility computation: The determination of whether some set of surfaces, or sample points, is visible to a synthetic observer.

Shading computation: The determination of radiometric values on the surface (eventually interpreted as colors) as viewed by the observer.

MODELING, ACQUISITION, AND SIMPLIFICATION

In practice, algorithms require input. Realistic scene generation demands extremely complex geometric and radiometric models—for example, of scene geometry and reflectance properties, respectively. These inputs must arise from some source; this ***model acquisition*** problem is a core problem in graphics. Models may be generated by a human designer (for example, using Computer-Aided Design packages), generated procedurally (for example, by applying recursive rules), or constructed by machine-aided manipulation of image data (for example, generating 3D topographical maps of terrestrial or extra-terrestrial terrain from multiple photographs), or other machine-sensing methods (e.g., [CL96]). Methods for completely automatic (i.e., not human-assisted) generation of large-scale geometric models are still in their infancy. When datasets become extremely large, some kind of hierarchical, persistent spatial database is required for efficient storage and access to the data [FKST96], and simplification algorithms are necessary to store and display complex objects with varying fidelity (see, e.g., [CVM+96, HDD+92]).

REPRESENTATION: GEOMETRY, LIGHT, AND FORCES

Once a geometric model is acquired, it must be represented, and perhaps indexed spatially, for efficient manipulation. A broad variety of intrinsic (winged-edge, quad-edge, facet-edge, etc.) and extrinsic (quadtree, octree, k-d tree, BSP tree, B-rep, CSG, etc.) data structures have been developed to represent geometric data. Continuous, implicit functions have been used to model shape, as have discretized volumetric representations, in which data types or densities are associated with spatial "voxels." A subfield of modeling, Solid Modeling (Chapter 47), represents shape, mass, material, connectivity, etc. properties of objects, so that, for example, complex object assemblies may be defined for use in Computer-Aided Machining environments (Chapter 46). Some of these data structures can be adaptively subdivided, and made persistent (that is, made to exist in memory and in nonvolatile storage; see Section 42.6), so that models with wide-scale variations, or simply an

enormous data size, may be organized. None of these data structures is universal; each has been brought to bear in specific circumstances, depending on the nature of the data (manifold vs. nonmanifold, polyhedral vs. curved, etc.) and the problem at hand. We forego here a detailed discussion of representational issues; see Chapters 45 and 47.

The data structures alluded to above represent "macroscopic" properties of scene geometry—shape, gross structure, etc. Representing material properties, including reflectance over each surface, and possibly surface microstructure (such as roughness) and substructure (as with layers of skin or other tissue), is another fundamental concern of graphics. For each surface, computer graphics researchers craft and employ quantitative descriptions of the interaction of radiant energy and/or physical forces with objects having these properties. Examples include human-made objects such as machine parts, furniture, and buildings; organic objects such as flora and fauna; naturally occurring objects such as molecules, terrains, and galaxies; and wholly synthetic objects and materials. Analogously, suitable representations of radiant energy and physical forces also must be crafted in order that the simulation process can model such effects as erosion [DPH96].

SIMULATION

Graphics brings to bear a wide variety of simulation processes to predict behavior. For example, one might detect collisions to simulate a pair of tumbling dice, or simulate frictional forces in order to provide haptic (touch) feedback through a mechanical device to a researcher manipulating a virtual object [LMC94]. Increasingly, graphics researchers are incorporating spatialized sounds into simulations as well. These physically-based simulations are integral to many graphics applications. However, the generation of synthetic imagery is the most fundamental operation in graphics. The next section describes this "rendering" problem.

RENDERING

For static scenes, and with more difficulty when conditions change with time, rendering can be factored into geometrically and radiometrically view-independent tasks (such as spatial partitioning for intervisibility, and the computation of diffuse illumination) and their view-dependent counterparts (culling and specular illumination, respectively). The view-independent tasks can be cast as precomputations, while at least some view-dependent tasks cannot occur until the instantaneous viewpoint is known. Recently, these distinctions have been somewhat blurred by the development of data structures that organize lazily-computed, view-dependent information for use in interactive settings [TBD96].

We discuss the two major components of rendering, visibility (Section 42.3) and shading (Section 42.4), and review the formulation of shading as a spatially distributed, recursive, radiometrically weighted visibility computation. In the discrete domain, we discuss sampling (Section 42.5). We then pose several challenges for the future, comprising problems of current or future interest in computer graphics on which computational geometry may have a substantial impact (Section 42.6). Finally, we list several further references (Section 42.7).

42.3 VISIBILITY

GLOSSARY

Pixel: Picture element, for example on a raster display.

Viewport: A 2D array of pixels, typically comprising a rectangular region on a computer display.

View frustum: A truncated rectangular pyramid, representing the synthetic observer's field of view, with the synthetic eyepoint at the apex of the pyramid. The truncation is typically accomplished using *near* and *far* clipping planes, analogous to the "left, right, top, and bottom" planes that define the rectangular field of view. (If the synthetic eyepoint is placed at infinity, the frustum becomes a rectangular parallelpiped.) Only those portions of the scene geometry that fall inside the view frustum are rendered.

Rasterization: The transformation of a continuous scene description, through discretization and sampling, into a discrete set of pixels on a display device.

Ray casting: A hidden-surface algorithm in which, for each pixel of an image, a ray is cast from the synthetic eyepoint through the center of the pixel [App68]. The ray is parametrized by a variable t such that $t = 0$ is the eyepoint, and $t > 0$ indexes points along the ray increasingly distant from the eye. The first intersection found with a surface in the scene (i.e., the intersection with minimum positive t) locates the visible surface along the ray. The corresponding pixel is assigned the intrinsic color of the surface, or some computed value.

Depth-buffering (also ***z-buffering***)***:*** An algorithm which resolves visibility by storing a discrete depth (initialized to some large value) at each pixel [Cat74]. Only when a rendered surface fragment's depth is less than that stored at the pixel can the fragment's color replace that currently stored at the pixel.

LOCAL VISIBILITY COMPUTATIONS

Given a scene composed of modeling primitives (e.g., polygons, or spheres), and a viewing frustum defining an eyepoint, a view direction, and field of view, the visibility operation determines which scene points or fragments are ***visible***—connected to the eyepoint by a line segment that meets the closure of no other primitive. The visibility computation is global in nature, in the sense that the determination of visibility along a single ray may involve all primitives in the scene. Typically, however, visibility computations can be organized to involve coherent subsets of the model geometry.

In practice, algorithms for visible surface identification operate under severe constraints. First, available memory is limited. Second, the computation time allowed may be a fraction of a second—short enough to achieve interactive refresh rates under changes in viewing parameters (for example, the location or viewing direction of the observer). Third, visibility algorithms must be simple enough to be practical, but robust enough to apply to highly degenerate scenes that arise in practice.

The advent of machine rendering techniques brought about a cascade of screen-space and object-space combinatorial hidden-surface algorithms, famously surveyed and synthesized in [SSS74]. However, a memory-intensive screen-space technique—*depth-buffering*—soon won out due to its brutal simplicity and the decreasing cost of memory. In depth-buffering, specialized hardware performs visible surface determination independently at each pixel. Each polygon to be rendered is rasterized, producing a collection of pixel coordinates and an associated depth for each. A polygon fragment is allowed to "write" its color into a pixel only if the depth of the fragment at hand is less than the depth stored at the pixel (all pixel depths are initialized to some large value). Thus, in a complex scene each pixel might be written many times to produce the final image, wasting computation and memory bandwidth. This is known as the ***overdraw*** problem.

Two decades of spectacular improvement in graphics hardware have ensued, and high-end graphics workstations now contain hundreds of increasingly complex processors that clip, illuminate, rasterize, and texture millions of polygons per second. This capability increase has naturally led users to produce ever more complex geometric models, which suffer from increasing overdraw. Object simplification algorithms, which represent complex geometric assemblages with simpler shapes, do little to reduce overdraw. Thus, visible-surface identification (hidden-surface elimination) algorithms have again come to the fore (Section 25.6.1). In recent years, several hybrid object-space/screen-space visibility algorithms have emerged (e.g., [GKM93]). As general purpose processors continue to become faster, such hybrid algorithms will become more widely used. In certain situations, these algorithms will operate entirely in object space, without relying on special-purpose graphics hardware [CT96]. That is, sufficiently fast processors and efficient algorithms will augment, and may supplant, the depth-buffer in graphics architectures of the coming decade.

GLOBAL VISIBILITY COMPUTATIONS

Real-time systems perform visibility computations from an instantaneous synthetic viewpoint along rays associated with one or more samples at each pixel of some viewport. However, visibility computations also arise in the context of global illumination algorithms, which attempt to identify *all* significant light transport among point and area emitters and reflectors, in order to simulate realistic visual effects such as diffuse and specular interreflection and refraction. A class of *global* visibility algorithms has arisen for these problems. For example, in radiosity computations, a fundamental operation is determining ***area-area*** visibility in the presence of *blockers*; that is, the identification of those (area) surface elements visible to a given element, and for those partially visible, all tertiary elements causing (or potentially causing) occlusion [HW91, HSA91].

CONSERVATIVE ALGORITHMS

Graphics algorithms often employ *quadrature* techniques in their innermost loops—for example, estimating the energy arriving at one surface from another by casting multiple rays and determining an energy contribution along each. Thus, any efficiency gains in this frequent process (e.g., omission of energy sources known not to contribute energy at the receiver, or omission of objects known not to be blockers)

will significantly improve overall system performance. Similarly, occlusion culling algorithms (omission of objects known not to contribute pixels to the rendered image) can significantly reduce overdraw. Both techniques are examples of ***conservative*** algorithms, which overestimate some geometric set by combinatorial means, then perform a final sampling-based operation that produces a (discrete) solution or quadrature. Of course, the success of conservative algorithms in practice depends on two assumptions: first, that through a relatively simple computation, a usefully tight bound can be attained on whatever set would have been computed by a more sophisticated (e.g., exact) algorithm; and second, that the aggregate time of the conservative algorithm and the sampling pass is less than that of an exact algorithm for input sizes encountered in practice.

This idea can be illustrated as follows. Suppose the task is to render a scene of n polygons. The choice to render the visible fragments *exactly* leads to an algorithm that expends at least kn^2 time, since since n polygons (e.g., two slightly misaligned combs, each with $n/2$ teeth) can cause $O(n^2)$ visible fragments to arise. But a conservative algorithm might simply render all n polygons, incurring some overdraw (to be resolved by a depth-buffer) at each pixel, but expending only time linear in the size of the input.

This highlights an important difference between computational geometry and computer graphics. Standard computational geometry cost measures would show that the $O(n^2)$ algorithm is optimal in an output-sensitive model (Section 25.6.1). In computer graphics, hardware considerations motivate a fundamentally different approach: rendering a (judiciously chosen) superset of those polygons that will contribute to the final image. A major open problem is to unify these approaches by finding a cost function that effectively models such considerations (see below).

HARDWARE TRENDS

Special-purpose graphics hardware may wane in influence, except for the *crossbar switch* [MCEF94], or geometric transformation operation, that takes rendered entities from their initial locations (fragments of object-space primitives) to their final locations in screen space (rendered pixels). Likewise, modern graphics architectures must perform visibility determination, and all contain some form of depth-buffer. However, given sufficiently fast ray casting capability, the depth-buffer might no longer be necessary. One could perform "analytic visibility" at each pixel, obviating the standard graphics technique of fragmenting all objects into large collections of polygons, then rasterizing them. Another alternative is "image-based" rendering (see below).

OPEN PROBLEMS

Each of the problems below assumes a geometric model consisting of n polygons.

1. The set of visible fragments can have complexity $\Omega(n^2)$ in the worst case. However, the complexity is lower for many scenes. If k is the number of edge incidences (vertices) in the projected visible scene, the set of visible fragments can be computed in optimal output-sensitive $O(nk^{1/2} \log\ n)$ time [SO92]. Although specialized results have been obtained, optimality has not be reached in many cases. See Table 25.6.1.

2. Give a spatial partitioning and ray casting algorithm that runs in amortized nearly-constant time (that is, has only a weak asymptotic dependence on total scene complexity). Identify a useful "density" parameter of the scene (e.g., the largest number of simultaneously visible polygons), and express the amortized cost of a ray cast in terms of this parameter.

3. Give an output-sensitive algorithm which, for specified viewing parameters, determines the set of "contributing" polygons—i.e., those which contribute their color to at least one viewport pixel.

4. Give an output-sensitive algorithm which, for specified viewing parameters, approximates the visible set to within ϵ. That is, produce a superset of the visible polygons of size (alternatively, total projected area) at most $(1 + \epsilon)$ times the size (resp., projected area) of the true set. Is the lower bound for this problem asymptotically smaller than that for the exact visibility problem?

5. For machine-dependent parameters A and B describing the transform (per-vertex) and fill (per-pixel) costs of some rendering architecture, give an algorithm to compute a superset S of the visible polygon set minimizing the rendering cost on the specified architecture.

6. In a local illumination computation, identify those polygons (or a superset) visible from the synthetic observer, and construct, for each visible polygon P, an efficient function $V(p)$ that returns 1 iff point $p \in P$ is visible from the viewpoint.

7. In a global illumination computation, identify all pairs (or a superset) of intervisible polygons, and for each such pair P, Q, construct an efficient function $V(p, q)$ that returns 1 iff point $p \in P$ is visible from point $q \in Q$.

8. ***Image-based rendering*** [MB95]: Given a 3D model, generate a minimal set of images of the model such that for all subsequent query viewpoints, the correct image can be recovered by combination of the sample images.

42.4 SHADING

Through sampling and visibility operations, a visible surface point or fragment is identified. This point or fragment is then ***shaded*** according to a *local* or *global* illumination algorithm. Given scene light sources and material reflection and transmission properties, and the propagative media comprising and surrounding the scene objects, the shading operation determines the color and intensity of the incident and exiting radiation at the point to be shaded. Shading computations can be characterized further as ***view-independent*** (modeling only purely diffuse interactions, or directional interactions with no dependence on the instantaneous eye position) or ***view-dependent***.

Most graphics workstations perform a local shading operation in hardware, which, given a point light source, a surface point, and an eye position, evaluates the energy reaching the eye via a single reflection from the surface. This local operation is implemented in the software and hardware offered by most workstations. However, this simple model cannot produce realistic lighting cues such as shadows,

reflection, and refraction. These require more extensive, global computations as described below.

GLOSSARY

Irradiance: Total power per unit area impinging on a surface element. Units: POWER PER RECEIVER AREA.

BRDF: The Bidirectional Reflectance Distribution Function, which maps incident radiation (at general position and angle of incidence) to reflected exiting radiation (at general position and angle of exiting). Unitless, in $[0, 1]$.

BTDF: The Bidirectional Transmission Distribution Function, which maps incident radiation (at general position and angle of incidence) to transmitted exiting radiation (at general position and angle of exiting). Analogous to the BRDF.

Radiance: The fundamental quantity in image synthesis, which is conserved along a ray traveling through a nondispersive medium, and is therefore "the quantity that should be associated with a ray in ray tracing" [CW93]. Units: POWER PER SOURCE AREA PER RECEIVER STERADIAN.

Radiosity: A global illumination algorithm for ideal diffuse environments. Radiosity algorithms compute shading estimates that depend only on the surface normal and the size and position of all other surfaces and light sources, and are independent of view direction. Also a physical quantity with units POWER PER SOURCE AREA.

Ray tracing: An image synthesis algorithm in which ray casting is followed, at each surface, by a recursive shading operation involving a hemispherical integral of irradiance at each surface point. Ray tracing algorithms are best suited to scenes with small light sources and specular surfaces.

Hybrid: Global illumination algorithms that model both diffuse and directional interactions (e.g., [SP89]).

SHADING AS RECURSIVE WEIGHTED INTEGRATION

Most generally, the shading operation computes the energy leaving a differential surface element in a specified differential direction. This energy depends on the surface's emittance and on the product of the surface's reflectance with the total energy incident from all other surfaces. This relation is known as the *Rendering Equation* [Kaj86], which states intuitively that each surface fragment's appearance, as viewed from a given direction, depends on any light it emits, plus any light (gathered from other objects in the scene) that it reflects in the direction of the observer. Thus, shading can be cast as a recursive integration; to shade a surface fragment F, shade all fragments visible to F, then sum those fragments' illumination upon F (appropriately weighted by the BRDF or BTDF) with any direct illumination of F. Effects such as diffuse illumination, motion blur, Fresnel effects, etc., can be simulated by supersampling in space, time, and wavelength, respectively, and then averaging [CPC84].

Of course, a base case for the recursion must be defined. Classical ray tracers truncate the integration when a certain recursion depth is reached. If this maximum depth is set to 1, ray casting (the determination of visibility for eye rays only)

results. More common is to use a small constant greater than one, which leads to "Whitted" or "classical" ray tracing [Whi80]. For efficiency, practitioners also employ a thresholding technique: when multiple reflections cause the weight with which a particular contribution will contribute to the shading at the root to drop below a specified threshold, the recursion ceases. These termination conditions can, under some conditions, cause important energy-bearing paths to be overlooked. For example, an extremely bright light source (such as the sun) filtering through many parts of a house to reach an interior space may be incorrectly discarded by this condition.

ALIASING

From a purely physical standpoint, the amount of energy leaving a surface in a particular direction is the product of the spherical integral of incoming energy and the bidirectional reflectance (and transmittance, as appropriate) in the exiting direction. From a psychophysical standpoint, the perceived color is an inner product of the energy distribution incident on the retina with the retina's spectral response function. We do not consider psychophysical considerations further here.

Global illumination algorithms perform an integration of irradiance at each point to be shaded. Ray tracing and radiosity are examples of global illumination algorithms. Since no closed-form solutions for global illumination are known for general scenes, practitioners employ sampling strategies. Graphics algorithms typically attempt "reconstruction" of some illumination function (e.g., irradiance, or radiance), given some set of samples of the function's values and possibly other information, for example about light source positions, etc. However, such reconstruction is subject to error for two reasons.

First, the well-known phenomenon of ***aliasing*** occurs when insufficient samples are taken to find all high-frequency terms in a sampled signal. In image processing, samples arise from measurements, and reconstruction error arises from samples that are too widely spaced. However, in graphics, the sample values arise from a simulation process, for example, the evaluation of a local illumination equation, or the numerical integration of irradiance. Thus, reconstruction error can arise from simulation errors in generating the samples. This second type of error is called ***biasing***.

For example, classical ray tracers [Whi80] may suffer from biasing in three ways. First, at each shaded point, they compute irradiance only: from direct illumination by point lights; along the reflected direction; and along the refracted direction. Significant "indirect" illumination that occurs along any direction other than these is not accounted for. Thus, indirect reflection and focusing effects are missed. Classical ray tracers also suffer biasing by truncating the depth of the recursive ray tree at some finite depth d; thus, they cannot find significant paths of energy from light source to eye of length greater than d. Third, classical ray tracers truncate ray trees when their weight falls below some threshold. This can fail to account for large radiance contributions due to bright sources illuminating surfaces of low reflectance.

OPEN PROBLEMS

An enormous literature of adaptive, backward, forward, distribution, etc. ray trac-

ers has grown up to address sampling and bias errors. However, the fundamental issues can be stated simply as:

1. Given a geometric model M, a collection of light sources L, a synthetic viewpoint E, and a threshold ϵ, identify all optical paths to E bearing radiance greater than ϵ.

2. Given a geometric model M, a collection of light sources L, and a threshold ϵ, identify *all* optical paths bearing radiance greater than ϵ.

A related *inverse* problem arises in machine vision, now being adopted by computer graphics practitioners as a method for acquiring large-scale geometric models from imagery:

3. An observation of a real object comprises the *product* of irradiance and reflection (BRDF). How can one deduce the BRDF from such observations?

42.5 SAMPLING

Sampling patterns can arise from a regular grid (e.g., pixels in a viewport) but these lead to a stair-stepping kind of aliasing. One solution is to ***supersample*** (i.e., take multiple samples per pixel) and average the results. However, one must take care to supersample in a way that does not align with the scene geometry or some underlying attribute (e.g., texture) in a periodic, spatially varying fashion; otherwise aliasing (including Moiré patterns) will result.

DISCREPANCY

The quality of sampling patterns can be evaluated with a measure known as ***discrepancy***. For example, if we are sampling in a pixel, features interacting with the pixel can be modeled by line segments (representing parts of edges of features) crossing the pixel. These segments divide the pixel into two regions. A good sampling strategy will ensure that the proportion of sample points in each region approximates the proportion of pixel area in that region. The difference between these quantities is the discrepancy of the point set with respect to the line segment. We define the discrepancy of a set of samples (in this case) as the maximum discrepancy with respect to all line segments. Other measures of discrepancy are possible, as described below. See also Chapter 10, and in particular Section 10.12.

Sampling patterns are used to solve integral equations. The advantage of using a low discrepancy set is that the solution will be more accurately approximated, resulting in a better image. These differences are expressed in solution convergence rates as a function of the number of samples. For example, truly random sampling has a discrepancy that grows as $O(N^{-\frac{1}{2}})$ where N is the number of samples. There are other sampling patterns (e.g., the *Hammersley points*) that have discrepancies growing as $O(N^{-1}\log^{k-1}N)$. Sometimes one wishes to combine values obtained by different sampling methods [VG95]. The search for good sampling patterns, given a fixed number of samples, is often done by running an optimization process which

aims to find sets of ever-decreasing discrepancy. A crucial part of any such process is the ability to quickly compute the discrepancy of a set of samples.

COMPUTING THE DISCREPANCY

There are two common questions that arise in the study of discrepancy: first, given fixed N, how to construct a good sampling pattern in the model described above; second, how to construct a good sampling pattern in an alternative model.

For concreteness, consider the problem of finding low discrepancy patterns in the unit square. The unit square models an individual pixel. As stated above, the geometry of objects is modeled by edges that intersect the pixel dividing it into two regions, one where the object exists and one where it does not. An ideal sampling method would sample the regions in proportion to their relative areas.

We model this as a discrepancy problem as follows. Let S be a sample set of points in the unit square. For a line l (actually, a segment arising from a polygon boundary in the scene being rendered), define the two regions S^+ and S^- into which l divides S. Ideally, we want a sampling pattern that has the same fraction of samples in the region S^+ as the area of S^+. Thus, in the region S^+, the discrepancy with respect to l is

$$|\sharp(S \cap S^+)/\sharp(S) - Area(S^+)|,$$

where $\sharp(\cdot)$ denotes the cardinality operator. The discrepancy of the sample set S with respect to a line l is defined as the larger of the discrepancies in the two regions. The discrepancy of set S is then the maximum, over *all* lines l, of the discrepancy of S with respect to l.

Finding the discrepancy in this setting is an interesting computational geometry problem. First, we observe that we do not need to consider all lines. Rather, we need consider only those lines that pass through two points of S, plus a few lines derived from boundary conditions. This suggests the $O(n^3)$ algorithm of computing the discrepancy of each of the $O(n^2)$ lines separately. This can be improved to $O(n^2 \log n)$ by considering the fan of lines with a common vertex (i.e., one of the sample points) together. This can be further improved by appealing to duality. The traversal of this fan of lines is merely a walk in the arrangement of lines in dual space that are the duals of the sample points. This observation allows us to use techniques similar to those in Chapter 21 to derive an algorithm that runs asymptotically as $O(n^2)$. Full details are given in [DEM93].

There are other discrepancy models that arise naturally. A second obvious candidate is to measure the discrepancy of sample sets in the unit square with respect to axis-oriented rectangles. Here we can achieve a discrepancy of $O(n^2 \log n)$, again using geometric methods. We use a combination of techniques, appealing to the incremental construction of 2D convex hulls to solve a basic problem, then using the sweep paradigm to extend this incrementally to a solution of the more general problem. The sweep is easier in the case in which the rectangle is anchored with one vertex at the origin, yielding an algorithm with running time $O(n \log^2 n)$.

The model given above can be generalized to compute ***bichromatic discrepancy***. In this case, we have sample points that are colored either black or red. We can now define the discrepancy of a region as the difference between its number of red and black points. Alternatively, we can look for regions (of the allowable type) that are most nearly monochromatic in red while their complements are nearly monochromatic in black. This latter model has application in computational learn-

ing theory. For example, red points may represent situations in which a concept is true, black situations where it is false. The minimum discrepancy rectangle is now a classifier of the concept. This is a popular technique for computer-assisted medical diagnosis.

The relevance of these algorithms to computational geometry is that they will lead to faster algorithms for testing the "goodness" of sampling patterns, and thus eventually more efficient algorithms with bounded sampling error. Also, algorithms for computing the discrepancy relative to a particular set system are directly related to the system's VC-dimension (see Sections 31.2 and 34.4).

OPEN PROBLEMS

The problems below are open for both the unit cube and unit ball in all dimensions.

1. Given N, generate a minimum-discrepancy pattern of N samples.
2. Given a low-discrepancy pattern of K points, generate a low (or lower) discrepancy pattern of $K + 1$ points.

42.6 FURTHER CHALLENGES

We have described several core problems of computer graphics and illustrated the impact of computational geometry. We have only scratched the surface of a highly fruitful interaction; the possibilities are expanding, as we describe below. These computer graphics problems all build on the combinatorial framework of computational geometry and so have been, and continue to be, ripe candidates for application of computational geometry techniques. Numerous other problems remain whose combinatorial aspects are perhaps less obvious, but for which interaction may be equally fruitful.

We have focused this chapter on problems in which the parameters are static; that is, the geometry is unchanging, and nothing is moving (except perhaps the synthetic viewpoint). Now, we briefly describe situations where this is not the case and deeper analysis is required. In these situations it is likely that computational geometry can have a tremendous impact; we sketch some possibilities here.

Each of the static assumptions above may be relaxed, either alone or in combination. For example, objects may evolve with time; we may be interested in transient rather than steady-state solutions; material properties may change over time; object motions may have to be computed and resolved; etc. It is a challenge to determine how techniques of computational geometry can be modified to address state-of-the-art and future computer graphics tasks in dynamic environments.

Among the issues we have not addressed where these considerations are important are the following.

Collision detection and force feedback. Imagine that every object has an associated motion, and that some objects (e.g., virtual probes) are interactively controlled. Suppose further that when pairs of objects intersect, there is a reaction (due, e.g., to conservation of momentum). Here we wish to render frames and generate haptic feedback while accounting for such physical considerations. Are

there suitable data structures and algorithms within computational geometry to model and solve this problem (e.g., [LMC94, MC95])?

Model changes over time. In a realistic model, even unmoving objects change over time, for example becoming dirty or scratched. In some environments, objects rust or suffer other corrosive effects. Sophisticated geometric representations and algorithms are necessary to capture and model such phenomena [DPH96].

Inverse processes. Much of what we have described is a feed-forward process in which one specifies a model and a simulation process and computes a result. Of equal importance in design contexts is to specify a result and a simulation process, and compute a set of initial conditions that would produce the desired result. For example, one might wish to specify the appearance of a stage, and deduce the intensities of hundreds of illuminating light sources that would result in this appearance [SDSA93]. Or, one might wish to solve an inverse kinematics problem in which an object with multiple parts and numerous degrees of freedom is specified. Given initial and final states, one must compute a smooth, minimal energy path between the states, typically in an underconstrained framework. This is a common problem in robotics (see Section 41.1). However, the configurations encountered in graphics tend to have very high complexity. For example, convincingly simulating the motion of a human figure requires processing kinematic models with hundreds of degrees of freedom.

External memory algorithms. Computational geometry assumes a realm in which all data can be stored in RAM and accessed at no cost (or unit cost). Increasingly often, this is not the case in practice. For example, many large databases cannot be stored in main memory. Only a small subset of the model contributes to each generated image, and algorithms for efficiently identifying this subset, and maintaining it under small changes of the viewpoint or model, form an active research area in computer graphics. Given that motion in virtual environments is usually smooth, and that hard real-time constraints preclude the use of purely reactive, synchronous techniques, such algorithms must be *predictive* and *asynchronous* in nature [FKST96]. Achieving efficient algorithms for appropriately shuttling data between secondary (and tertiary) storage and main memory is an interesting challenge for computational geometry.

42.7 SOURCES AND RELATED MATERIAL

SURVEYS

All results not given an explicit reference above may be traced in these surveys:

[Dob92]: A survey article on computational geometry and computer graphics.

[Dor94]: A survey of object-space hidden-surface removal algorithms.

[Yao90, LP84]: Surveys of computational geometry.

RELATED CHAPTERS

Chapter 10: Geometric discrepancy theory and uniform distribution
Chapter 22: Triangulations
Chapter 23: Polygons
Chapter 25: Visibility
Chapter 32: Ray shooting and lines in space
Chapter 45: Splines and geometric modeling
Chapter 47: Solid modeling

REFERENCES

[App68] A. Appel. Some techniques for shading machine renderings of solids. In *Proc. AFIPS Spring Joint Comput. Conf.*, pages 37–45. Thompson Books, Washington, 1968.

[CAA+96] B. Chazelle, N. Amenta, T. Asano, G. Barequet, M. Bern, J.-D. Boissonnat, J. Canny, K. Clarkson, D. Dobkin, B. Donald, S. Drysdale, H. Edelsbrunner, D. Eppstein, A.R. Forrest, S. Fortune, K. Goldberg, M.T. Goodrich, L.J. Guibas, P. Hanrahan, C.M. Hoffmann, D. Huttenlocher, H. Imai, D. Kirkpatrick, D.T. Lee, K. Mehlhorn, V. Milenkovic, J. Mitchell, M. Overmars, R. Pollack, R. Seidel, M. Sharir, J. Snoeyink, G.T. Toussaint, S. Teller, H. Voelcker, E. Welzl, and C.K. Yap. Application challenges to computational geometry: CG impact task force report. Tech. Rep. TR-521-96, Dept. Comput. Sci., Princeton Univ., 1996. In B. Chazelle, J.E. Goodman, and R. Pollack, editors, *Discrete and Computational Geometry: Ten Years Later*, Amer. Math. Soc., Providence, to appear.

[Cat74] E. Catmull. *A Subdivision Algorithm for Computer Display of Curved Surfaces.* Ph.D. thesis, Univ. of Utah, Salt Lake City, 1974.

[CL96] B. Curless and M. Levoy. A volumetric method for building complex models from range images. *Comput. Graph.* (Proc. SIGGRAPH '96), 30:303–312, 1996.

[CPC84] R. Cook, T. Porter, and L. Carpenter. Distributed ray tracing. *Comput. Graph.* (Proc. SIGGRAPH '84), 18:137–145, 1984.

[CT96] S. Coorg and S. Teller. Temporally coherent conservative visibility. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 78–87, 1996.

[CVM+96] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P.K. Agarwal, F. Brooks, and W. Wright. Simplification envelopes. *Comput. Graph.* (Proc. SIGGRAPH '96), 30:119–128, 1996.

[CW93] M. Cohen and J. Wallace. *Radiosity and Realistic Image Synthesis.* Academic Press, Cambridge, 1993.

[DEM93] D. Dobkin, D. Eppstein, and D. Mitchell. Computing the discrepancy with applications to supersampling patterns. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 47–52, 1993.

[Dob92] D. Dobkin. Computational geometry and computer graphics. *Proc. IEEE*, 80:1400–1411, 1992.

[Dor94] S. Dorward. A survey of object-space hidden surface removal. *Internat. J. Comput. Geom. Appl.*, 4:325–362, 1994.

[DPH96] J. Dorsey, H. Pedersen, and P. Hanrahan. Flow and changes in appearance. *Comput. Graph.* (Proc. SIGGRAPH '96), 30:411–420, 1996.

[FKST96] T. Funkhouser, D. Khorramabadi, C. Séquin, and S. Teller. The UCB system for interactive visualization of large architectural models. *Presence*, 5:13–44, 1996.

[GKM93] N. Greene, M. Kass, and G. Miller. Hierarchical Z-buffer visibility. *Comput. Graph.* (Proc. SIGGRAPH '93), 27:231–238, 1993.

[HDD+92] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Comput. Graph.* (Proc. SIGGRAPH '92), 26:71–78, 1992.

[HSA91] P. Hanrahan, D. Salzman, and L. Aupperle. A rapid hierarchical radiosity algorithm. *Comput. Graph.* (Proc. SIGGRAPH '91), 25:197–206, 1991.

[HW91] E. Haines and J. Wallace. Shaft culling for efficient ray-traced radiosity. In *Proc. 2nd Eurographics Workshop Render.*, pages 122–138. Springer-Verlag, New York, 1991.

[Kaj86] J. Kajiya. The rendering equation. *Comput. Graph.* (Proc. SIGGRAPH '86), 20:143–150, 1986.

[LMC94] M. Lin, D. Manocha, and J. Canny. Fast contact determination in dynamic environments. In *Proc. 11th IEEE Internat. Conf. Robot. Autom.*, pages 602–609, 1994.

[LP84] D. Lee and F. Preparata. Computational geometry: A survey. *IEEE Trans. Comput.*, 33:1072–1101, 1984.

[MB95] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. *Comput. Graph.* (Proc. SIGGRAPH '95), 29:39–46, 1995.

[MC95] B. Mirtich and J. Canny. Impulse-based simulation of rigid bodies. In *Proc. ACM Interactive 3D Graph. Conf.*, pages 181–188, 1995.

[MCEF94] S. Molnar, M. Cox, D. Ellsworth, and H. Fuchs. A sorting classification of parallel rendering. *IEEE Comput. Graph. Appl.*, 14:23–32, 1994.

[SDSA93] C. Schoeneman, J. Dorsey, B. Smits, and J. Arvo. Painting with light. *Comput. Graph.* (Proc. SIGGRAPH '93), 27:143–146, 1993.

[SO92] M. Sharir and M. Overmars. A simple output-sensitive algorithm for hidden surface removal. *ACM Trans. Graph.*, 11:1–11, 1992.

[SP89] F. Sillion and C. Puech. A general two-pass method integrating specular and diffuse reflection. *Comput. Graph.* (Proc. SIGGRAPH '89), 23:335–344, 1989.

[SSS74] I. Sutherland, R. Sproull, and R. Schumacker. A characterization of ten hidden-surface algorithms. *ACM Comput. Surv.*, 6:1–55, 1974.

[TBD96] S. Teller, K. Bala, and J. Dorsey. Conservative radiance envelopes for ray tracing. In *Proc. 7th Eurographics Workshop Render.*, pages 105–114. Springer-Verlag, New York, 1996.

[VG95] E. Veach and L. Guibas. Optimally combining sampling techniques for Monte Carlo rendering. *Comput. Graph.* (Proc. SIGGRAPH '95), 29:419–428, 1995.

[Whi80] T. Whitted. An improved illumination model for shading display. *Commun. ACM*, 23:343–349, 1980.

[Yao90] F.F. Yao. Computational geometry. In R.A. Earnshaw and B. Wyvill, editors, *Algorithms in Complexity*, pages 345–490. Elsevier, Amsterdam, 1990.

43 PATTERN RECOGNITION

Joseph O'Rourke and Godfried T. Toussaint

INTRODUCTION

The two fundamental problems in a pattern recognition system are feature extraction (shape measurement) and classification. The problem of extracting a vector of shape measurements from a digital image can be further decomposed into three subproblems. The first is the image segmentation problem, i.e., the separation of objects of interest from their background. The cluster analysis methods discussed in Section 43.1 are useful here. The second subproblem is that of finding the objects in the segmented image. An example is the location of text lines in a document as illustrated in Section 43.2. The final subproblem is extracting the shape information from the objects detected. Here there are many tools available depending on the properties of the objects that are to be classified. The Hough transform (Section 43.2), polygonal approximation (Section 43.3), shape measurement (Section 43.4), and polygon decomposition (Section 43.5), are some of the favorite tools used here. Proximity graphs, discussed in Section 43.2, are used extensively for both cluster analysis and shape measurement.

The classification problem involves the design of efficient decision rules with which to classify the feature vector. The most powerful decision rules are the nonparametric rules which make no assumptions about the underlying distributions of the feature vectors. Of these the nearest-neighbor (NN) rule, treated in Section 43.6, is the most well-known. This section covers the three main issues concerning NN-rules: how to edit the data set so that little storage space is used, how to search for the nearest neighbor of a vector efficiently, and how to estimate the future performance of a rule both reliably and efficiently.

43.1 CLUSTER ANALYSIS AND CLASSIFICATION

GLOSSARY

Cluster analysis problem: Partitioning a collection of n points in some fixed-dimensional space into $m < n$ groups that are "natural" in some sense. Here m is usually much smaller than n.

Image segmentation problem: Partitioning the pixels in an image into "meaningful" regions, usually such that each region is associated with one physical object.

Dendrogram: A tree representing a hierarchy of categories or clusters.

Hierarchical clustering algorithms: Those that produce a dendrogram whose root is the whole set and whose leaves are the individual points.

Graph-theoretic clustering: Clustering based on deleting edges from a proximity graph.

K-means clustering: Tracking clusters over time by comparing new data with old means.

Classical cluster analysis requires partitioning points into natural clumps. "Natural" may mean that the clustering agrees with human perception, or it may simply optimize some natural mathematical measure of similarity or distance so that points that belong to one cluster are similar to each other and points far away from each other are assigned to different clusters. It is not surprising that such a general and fundamental tool has been applied to widely different subproblems in pattern recognition. One obvious application is to the determination of the number and description of classes in a pattern recognition problem where the classes are not known a priori, such as disease classification in particular or taxonomy in general. In this case m is not known beforehand and the cluster analysis reveals it.

A fundamental problem in pattern recognition of images is the segmentation problem: distinguishing the figure from the background. Clustering is one of the most powerful approaches to image segmentation, applicable even to complicated images such as those of outdoor scenes. In this approach each pixel in the image is treated as a complicated object by associating it with a local neighborhood. For example, we may define the 5×5 neighborhood of pixel p_{ij}, denoted by $N_5[p_{ij}]$, as $\{p_{mn} \mid i-2 \leq m \leq i+2, j-2 \leq n \leq j+2\}$. We next measure k properties of p_{ij} by making k measurements in $N_5[p_{ij}]$. Such measurements may include various moments of the intensity values (grey levels) found in $N_5[p_{ij}]$, etc. Thus each pixel is mapped into a point in k-dimensional pixel-space. Performing a cluster analysis of all the resulting $N \times N$ points in pixel-space yields the desired partitioning of the pixels into categories. Labeling each category of pixels with a different color then produces the segmentation.

See [Gor96] for a survey of clustering methods in which the objects in one class are not only required to be similar to each other but must satisfy other constraints on the distances between the objects or on the topology of the resulting dendrograms.

HIERARCHICAL CLUSTERING

In taxonomy there is no special number m that we want to discover but rather the goal is the production of a *dendrogram* (tree) that grows all the way from one cluster to n clusters and shows us at once how good a partitioning is obtained for any number of clusters between one and n. Such methods are referred to as *hierarchical* methods. They fall into two groups: ***agglomerative*** (bottom-up, merging) and ***divisive*** (top-down, splitting). Furthermore, each of these methods can be used with a variety of distance measures between subsets to determine when to merge or split. One of the most popular clustering methods is the agglomerative method in which the distance between subsets is the maximum distance between a pair of elements, one in each subset. This algorithm is known as the ***complete-linkage*** (or farthest-neighbor) algorithm. Krznaric and Levcopoulos [KL95] show that a complete-linkage hierarchy can be computed in $O(n \log^2 n)$ time and $O(n)$ space.

GRAPH-THEORETIC CLUSTERING

The most powerful methods of clustering in difficult problems, which give results having the best agreement with human performance, are the graph-theoretic methods [JT92]. The idea is extremely simple: Compute some proximity graph (such as the minimal spanning tree) of the original points. Then delete (in parallel) any edge in the graph that is much longer (according to some criterion) than its neighbors. The resulting forest is the clustering (each tree in this forest is a cluster).

K-MEANS TYPE CLUSTERING

There are many applications where we know that there are exactly K clusters, for example in character recognition. However, because of external factors such as the variations in people's hand-printing over time, or a change in the parameters of a machine due to wear or weather conditions, the clusters must be "tracked" over time. One of the most popular methods for doing this is the ***K-means algorithm***. Here, for each of the K clusters the mean is calculated and future data are classified with the closest mean decision rule. New data classified with this rule into some cluster are then labeled as members of that cluster and the means are updated. For some of the latest computational geometry results on these methods and pointers to the literature, see [Sch91].

DISTANCES BETWEEN SETS

A fundamental computational primitive of almost all clustering algorithms is the frequent computation of some distance measure between two sets (subsets) of points. This is especially so in the popular hierarchical methods discussed above. There exists a large variety of distance and more general similarity measures for this purpose. Here we mention a few. Most efficient algorithms for distance between sets apply only in $\mathbb{R}^2$ but some methods extend to higher dimensions. Let P and Q be two convex polygons of n sides each. Two distance measures should be distinguished: the minimum *element* distance, the smallest distance between a vertex or edge of P and a vertex or edge of Q, and the minimum *vertex* distance, the minimum distance between a vertex of P and a vertex of Q. The minimum element distance can be computed in $O(\log n)$ time [Ede85]. On the other hand, computation of the minimum vertex distance between P and Q has a linear lower bound. For the case of two nonintersecting convex polygons several different $O(n)$ time algorithms are available, and the same bound can be achieved for crossing convex polygons [Tou84].

Let R be a set of n red points and B a set of n blue points in the plane. Both the minimum distance and the maximum distance between R and B can be computed in $O(n \log n)$ time. For the latter problem, two algorithms are available. The first [TM82] is simple but does not appear to generalize to higher dimensions. The second [BT83] works by reducing the maximum distance problem between R and B to computing the diameter of 81 convex polygons. These are obtained by computing the convex hulls of the unions of 81 carefully selected subsets of R and B, and then reporting the maximum of these 81 diameters as the maximum distance. These

ideas can be extended to obtain efficient algorithms for all dimensions [Rob93]. Therefore, any improvement in high-dimensional diameter algorithms automatically improves maximum-distance algorithms.

43.2 EXTRACTING SHAPE FROM DOT PATTERNS

HOUGH TRANSFORMS

The ***Hough transform*** was originally proposed (and patented) as an algorithm to detect straight lines in digital images [Lea92]. The method may be used to detect any parametrizable pattern, and has been generalized to locate arbitrary shapes in images. The basic idea is to let each above-threshold pixel in the image vote for the points in the parameter space that could generate it. The votes are accumulated in a quantized version of the parameter space, with high vote tallies indicating detection.

Examples

1. *Lines.* Let the lines in the (x, y) image plane be parametrized as $y = mx + b$. Then a pixel at (x_0, y_0) is a witness to a line passing through it, that is, an (m, b) pair satisfying $y_0 = mx_0 + b$. Thus, (x_0, y_0) votes for all those (m, b) pairs: the line in parameter space dual to the pixel.

2. *Circles.* Parametrize the circles by their center and radius, (x_c, y_c, r). Then a pixel (x_0, y_0) gives evidence for all the parameter triples on the circular cone in the 3-parameter space with apex at $(x_0, y_0, 0)$ and axis parallel to the r-axis.

3. *Object location.* Suppose a known but arbitrary shape S is expected to be found in an image, and its most likely location is sought. For translation-only, the parameter space represents the location of some fixed point of S. Each pixel in the image of the right shading or color votes for all those translations that cover it appropriately.

The demands of high-dimensional vote accumulators have engendered the study of *dynamic quantization* of the parameter space, and ***geometric hashing***. This latter technique has features in the image vote for each member of a library of shapes by hashing into a table using the feature coordinates as a key. Each table entry may correspond to several shapes at several displacements, but all receive votes. Geometric hashing has been applied with some success to the *molecular docking* problem [LW88].

More recently, new variants of the Hough transform inspired by results in computational geometry have appeared. In [AK96] two such algorithms are presented and studied with respect to the tradeoff that exists between computational complexity and effectiveness of line detection. They obtain efficient implementations by using the plane-sweep paradigm.

TEXT-LINE ORIENTATION INFERENCE

In an automated document analysis system, given a block (paragraph) of text, the text-line orientation inference problem consists of determining the location and direction of the lines of text on the page. Almost always these lines are either horizontal (e.g., English) or vertical (e.g., Chinese) The fundamental geometric property that allows this problem to be solved is the fact that according to a universal typesetting convention guided by ease of reading, characters are printed closer together within textlines than between textlines.

One of the most successful, robust, skew-tolerant, simple, and elegant techniques for text-line orientation inference was proposed by Ittner [Itt93]. Assume that the given text block B consists of n black connected components (characters). The three key steps in the procedure are: (1) idealize each character by a point, thus obtaining a set S of n points in the plane; (2) construct the Euclidean minimal spanning tree MST(S) of the n points obtained in (1); and (3) determine the textline orientation by analysis of the distribution of the orientations of the edges in MST(S). Step (1) is done by computing the center of the bounding box of each character. Cheriton and Tarjan proposed a simple algorithm for computing the MST of a graph in $O(E)$ time where E is the number of edges in the graph. Fortunately there are many graphs defined on S (usually belonging to the class of proximity graphs [JT92]) that have the property that they contain the MST(S) and also have $O(n)$ edges. For these graphs the Cheriton-Tarjan algorithm runs in $O(n)$ time.

RELATIVE NEIGHBORHOOD GRAPHS

Relative neighborhood graphs (RNG's), introduced in [Tou80b], capture proximity between points by connecting nearby points with a graph edge. The many possible notions of "nearby" (in several metrics) lead to a variety of related graphs. It is easiest to view the graphs as connecting points only when certain regions of space are empty. Let V be a set of n points in $\mathbb{R}^d$.

GLOSSARY

δ(p,q): the distance between two points p and q.

L_p: The distance metric L_p defined as $\delta_p(x,y) = (\sum_{i=1}^{d} |x_i - y_i|^p)^{1/p}$. For L_1 this reduces to $\sum_{i=1}^{d} |x_i - y_i|$, and for L_∞, to $\max_{1 \le i \le d} |x_i - y_i|$.

Ball B(x,r): The open ball $B(x,r) = \{y \mid \delta(x,y) < r\}$.

***Nearest-neighbor Graph* NNG(V):** The graph with vertex set V and an edge (p,q) iff $B(p, \delta(p,q)) \cap V = \emptyset$.

Lune L(p,q): $L(p,q) = B(p, \delta(p,q)) \cap B(q, \delta(p,q))$.

***Relative neighborhood graph* RNG(V):** The graph with vertex set V and an edge (p,q) iff $L(p,q) \cap V = \emptyset$. Thus the edge is present iff

$$\delta(p,q) \le \max_{v \in V \setminus \{p,q\}} \{\delta(p,v), \delta(q,v)\}.$$

***Gabriel graph* GG(V):** The graph with vertex set V and an edge (p, q) iff

$$B\left(\frac{p+q}{2}, \frac{\delta(p,q)}{2}\right) \cap V = \emptyset\,.$$

β-lune: $L_\beta(p,q) = B\left(p(1-\frac{\beta}{2}) + q\frac{\beta}{2}, \frac{\beta}{2}\delta(p,q)\right) \cap B\left(q(1-\frac{\beta}{2}) + p\frac{\beta}{2}, \frac{\beta}{2}\delta(p,q)\right)$.

β-skeleton: The graph $G_\beta(V)$ with vertex set V and an edge (p, q) iff $L_\beta(p,q) \cap V = \emptyset$. The range $1 \leq \beta \leq 2$ is especially relevant, with $G_2(V) = \mathrm{RNG}(V)$ and $G_1(V) = \mathrm{GG}(V)$.

***Sphere-of-influence graph* SIG(V):** Let C_p be the circle centered on p with radius equal to the distance to a nearest neighbor of p. SIG(V) has node set V and an edge (p, q) iff C_p and C_q intersect in at least two points.

The *relative neighborhood graph* connects two points if the *lune* they determine is empty of points of V. The *Gabriel graph* is defined similarly, but with the *diameter sphere* required to be empty. The β-Skeletons are a continuous generalization of the Gabriel graph. These graph definitions are motivated by various applications: computer vision, texture discrimination, geographic analysis, pattern analysis, cluster analysis, and others.

Proximity graphs form a nested hierarchy, a version of which was first established in [Tou80b]:

THEOREM 43.2.1 *Hierarchy*

In any L_p metric, for a fixed set V and $1 \leq \beta \leq 2$,

$$\mathrm{NNG} \subseteq \mathrm{MST} \subseteq \mathrm{RNG} \subseteq G_\beta \subseteq \mathrm{GG} \subseteq \mathrm{DT}\,.$$

MST *is the minimum spanning tree, and* DT *the Delaunay triangulation, of V.*

Neighborhood graphs can have at most $O(n^2)$ edges, and $\Theta(n^2)$ complexity is achieved in many instances. For the L_2 metric in $\mathbb{R}^2$, however, RNG's (and their relatives) have linear size, which increases their usefulness. See Table 43.2.1.

TABLE 43.2.1 Size of relative neighborhood graphs.

DIM	METRIC	SIZE
2	$L_p, 1 < p < \infty$	$O(n)$: $\in [n-1, 3n-6]$
≥ 2	L_1, L_∞	$\Theta(n^2)$
3	L_2	$O(n^{4/3})$
$d > 4$	L_p	$\Theta(n^2)$

The many applications of neighborhood graphs have stimulated considerable effort on developing efficient algorithms for constructing them. The RNG has the most applications and has received the most attention. $O(n^3)$ time complexity is trivial to achieve. Exploiting the fact that the Delaunay triangulation is a superset leads easily to $O(n^2)$ in the plane. Further development is more difficult. Two milestones in algorithm development were Supowit's $O(n \log n)$ algorithm for L_2

in $\mathbb{R}^2$ [Sup83], and Agarwal and Matoušek's near-$O(n^{3/2})$ algorithm for general position points in $\mathbb{R}^3$ [AM92]. See Table 43.2.2.

TABLE 43.2.2 Relative neighborhood graph algorithms.

DIM	METRIC	COMPLEXITY
2	L_2	$O(n \log n)$
2	L_1, L_∞	$O(n \log n)$
3	L_2	$O(n^{3/2+\epsilon})$
3	L_1, L_∞	$O(n \log^2 n)$
d	L_2	$O(n^{2(1-\frac{1}{d+1})+\epsilon})$
d	L_1, L_∞	$O(n \log^{d-1} n)$

A related graph is the *sphere-of-influence* (SIG) graph. It has at most $18n = O(n)$ edges in $\mathbb{R}^2$ and can be computed in $\Theta(n \log n)$ time. The SIG can serve as a type of graph-theoretical "primal sketch." It, in some sense, explains the dot-pattern version of the Mueller-Lyer illusion.

See the survey [JT92] for further details on neighborhood graphs.

OPEN PROBLEMS

1. What is the maximum number of edges of an RNG in $\mathbb{R}^3$? It is known that it has at most $O(n^{4/3})$ edges, but no supra-linear lower bound is known.
2. That the SIG has at most $18n$ edges in the plane follows from a theorem of Erdős and Panwitz, but the best lower bound is $9n$. It is also known that the SIG has a linear number of edges in any fixed dimension [GPS94], but again tight results are not available.

43.3 POLYGONAL APPROXIMATION

Let $P = (p_1, p_2, \ldots, p_n)$ be a polygonal curve or chain in the plane, consisting of n points p_i joined by line segments $p_i p_{i+1}$. In general P may be closed and self-intersecting. Polygonal curves occur frequently in pattern recognition either as representations of the boundaries of figures or as functions of time representing, e.g., speech. In order to reduce the complexity of costly processing operations, it is often desirable to approximate a curve P with one that is composed of far fewer segments, yet is a close enough replica of P for the intended application. Some methods of reduction attempt smoothing as well. An important instance of the problem is to determine a new curve $Q = (q_1, q_2, \ldots, q_m)$ such that (1) m is significantly smaller than n; (2) the q_j are selected from among the p_i; and (3) any segment $q_j q_{j+1}$ that replaces the chain $q_j = p_r, \ldots, p_s = q_{j+1}$ is such that the distance between $q_j q_{j+1}$ and each p_k, $r \leq k \leq s$, is less than some predetermined

error tolerance ω. Different notions of distance, or error criteria, lead to different algorithmic issues. Moreover, for each distance definition, there are two constrained optimization problems that are of interest, Min-# and Min-ϵ.

GLOSSARY

Distance from point p to segment s: Minimum distance from p to any point of s.

Parallel-strip criterion: All the vertices $p_i, \ldots, p_t$ lie in a parallel strip of width 2ω whose center line is collinear with $q_j q_{j+1}$ [ET94].

Segment criterion: For each p_k, $r \leq k \leq s$, the distance from p_k to $q_j q_{j+1}$ is less than ω [MO88, CC96].

Min-# problem: Given the error tolerance ω, find a curve $Q = (q_1, \ldots, q_m)$ satisfying the constraint such that m is minimum.

Min-ϵ problem: Given m, find a curve $Q = (q_1, \ldots, q_m)$ satisfying the constraint such that the error tolerance is minimized.

The main results obtained are listed in Table 43.3.1.

TABLE 43.3.1 Polygonal approximation algorithm time complexities.

ERROR CRITERION	MIN-#	MIN-ϵ
Parallel strip	$O(n^2 \log n)$	$O(n^2 \log^2 n)$
Segment	$O(n^2)$	$O(n^2 \log n)$

OPEN PROBLEMS

Find algorithms for the strip criterion problems that match those for the segment problems. Perhaps quadratic performance is achievable for all four problems.

ITERATIVE ENDPOINT FITTING

There are many algorithms in both the pattern recognition and automated cartography literatures that are intended to yield approximations with few, but not necessarily the minimum number of, segments. This suffices for many applications, and often can be obtained more efficiently in both time and space complexities. One of the most popular such algorithms is ***iterative endpoint fitting***, which employs the parallel-strip criterion. Given a tolerance ω, the algorithm first attempts to use only one segment $Q = (p_1, p_n)$ to approximate P. If the error tolerance is exceeded, then that vertex of P whose distance to the line through $p_1 p_n$ is maximum is chosen to divide P into two subchains. The procedure is then recursively applied to these subchains. This procedure can be implemented to run in $O(n \log n)$ time.

43.4 SHAPE MEASUREMENT AND REPRESENTATION

MEDIAL AXIS

GLOSSARY

Medial axis: The set of points of a region P with more than one closest point among the boundary points ∂P of the region. Equivalently, it is the set of centers of maximal balls, i.e., of balls in P that are themselves not enclosed in another ball in P.

Voronoi diagram: The partition of a polygonal region P into cells each consisting of the points closer to a particular open edge or vertex than to any other. See Chapter 20.

The *medial* or *symmetric axis* was introduced by Blum [Blu67] to capture biological shape, and it has since found many other applications, for example, to geometric modeling (*offset* computations; see Section 47.2) and to mesh generation [SNTM92] (Section 22.4). It provides a central "skeleton" for an object that has found many uses. It connects to several other mathematical concepts, including the *cut locus* and most importantly, the Voronoi diagram (Chapter 20).

The medial axis of a convex polygon is a tree with each vertex a leaf. For a nonconvex polygon, the medial axis may have a parabolic arc associated with each reflex vertex (Figure 47.1.5). The basic properties of the medial axis were detailed by Lee [Lee82], who showed that the medial axis of a polygon P is just the Voronoi diagram minus the Voronoi edges incident to reflex vertices, and provided an $O(n \log n)$ algorithm for constructing it. After a long search by the community, an $O(n)$ algorithm was obtained [CSW95]. The simplest implementations are, however, quadratic [YR91]. See Table 52.4.1.

The medial axis has also found much use in image processing, where its digital computation is via *thinning algorithms*. Pioneered by Rosenfeld, these algorithms are very simple and easily parallelized [Cyc94].

The definition of medial axis extends to $\mathbb{R}^d$. Some work has explored $\mathbb{R}^3$ [SPB95], but currently the lack of reliable software hampers extensive applications.

POINT PATTERN MATCHING

Exact point pattern matching is an interesting algorithmic question related to string matching, but pattern recognition applications usually require some type of approximate matching. Two types may be distinguished [AG]: one-to-one matching, and Hausdorff matching.

GLOSSARY

One-to-one approximate matching: Let two finite sets of points A and B have the same cardinality. One-to-one matching requires finding a transformation of a given type of B such that each point of B is mapped to within a distance of

ϵ of a matched point of A. The matches are either determined by ***labels*** on the points, or the points are ***unlabeled*** and the match is to be discovered.

Decision problem: Given ϵ, is there such a matching?

Optimization problem: Find the minimum ϵ for which an approximate matching exists.

Hausdorff distance: For two finite sets A and B, perhaps of different cardinalities, the largest of the between-sets nearest-neighbor distances.

Hausdorff matching: Find a transformation of B that minimizes the Hausdorff distance from A.

Results on one-to-one matching algorithms obtained in [AMWW88] for a variety of permissible transformations are shown in Table 43.4.1.

TABLE 43.4.1 One-to-one point matching in two dimensions.

MOVEMENTS	LABELED	ϵ	COMPLEXITY
Translation	labeled	dec, opt	$O(n)$
Translation	unlabeled	decision	$O(n^6)$
Translation	unlabeled	optimization	$O(n^6 \log n)$
Rotation	labeled	decision	$O(n \log n)$
Rotation	labeled	optimization	$O(n^2)$
Trans+rot+refl	labeled	decision	$O(n^3 \log n)$
Trans+rot+refl	unlabeled	decision	$O(n^8)$

Hausdorff matching leads to analysis of envelopes of *Voronoi surfaces*. Typical results are shown in Table 43.4.2. Here we show the complexities when $|A| = |B| = n$, although the algorithms work for sets of different cardinalities.

TABLE 43.4.2 Hausdorff matching in the L_2 metric.

MOVEMENTS	DIM	COMPLEXITY
Translation	2	$O(n^3 \log n)$
Translation + rotation	2	$O(n^6 \log n)$
Translation	3	$O(n^{5+\epsilon})$

Another type of matching is *order type matching* (cf. Section 5.2). In [GP83], an $O(n^3)$ algorithm is given for finding all matchings between two planar point configurations in which their order types agree.

SYMMETRY DETECTION

Symmetry is an important feature in the analysis and synthesis of shape and form and has received considerable attention in the pattern recognition and computer

graphics literatures. In [WWV85] an $O(n \log n)$ algorithm is presented for computing the rotational symmetries of polygons and polyhedra of n vertices, but the constant in $\mathbb{R}^3$ is very large. Jiang and Bunke [JB91] give a simple and practical $O(n^2)$ time algorithm for polyhedra. One of the earliest applications of computational geometry to symmetry detection was the algorithm of Akl and Toussaint [AT79] to check for polygon similarity. Since then attention has been given to other aspects of symmetry and for objects other than polygons. Sugihara [Sug84] shows how a modification of the planar graph-isomorphism algorithm of Hopcroft and Tarjan can be used to find all symmetries of a wide class of polyhedra in $O(n \log n)$ time.

A related topic is centers of symmetry. Given a convex polygon P, associate with each point p in P the minimum area of the polygon to the left of any chord through p. The maximum over all points in P is known as ***Winternitz's measure of symmetry*** and the point p^* that achieves this maximum is called the ***center of area***. Diaz and O'Rourke [DO94] show that p^* is unique and propose an algorithm for computing p^* in time $O(n^6 \log^2 n)$. For a survey of the early work on detecting symmetry, see [Ead88].

THE ALPHA HULL

The α-shape $\mathcal{S}_\alpha$ of a set S of n points in $\mathbb{R}^3$ is a polyhedral surface whose boundary is a particular collection of triangles, edges, and vertices determined by the points of S [EM94]. It is similar in spirit to the β-skeleton of Section 43.2 in that it is a parametrized collection of shapes determined by an empty balls condition, but it emphasizes the external rather than the internal structure of the set. Let T be a subset of S of 1, 2, or 3 points. Then the convex hull of T, conv(T), is part of the boundary $\partial\mathcal{S}_\alpha$ of the α-shape iff the surface of some ball of radius α includes exactly the points of T while its interior is empty of points of S. Thus a triangle conv(T) is part of $\partial\mathcal{S}_\alpha$ iff there is an open α-ball that can "erase" all of the triangle but leave its vertices. $\mathcal{S}_\alpha$ is defined for all $0 \leq \alpha \leq \infty$, with $\mathcal{S}_0 = S$ and $\mathcal{S}_\infty = \text{conv}(S)$.

Every edge and triangle of $\mathcal{S}_\alpha$ is present in the Delaunay triangulation DT of S, and every edge and triangle in DT is present in some $\mathcal{S}_\alpha$. If α is varied continuously over its full range starting from ∞, the convex hull of S is gradually "eaten away" by smaller and smaller α-ball erasers, eventually exposing the original set of points. In between, the α-shape bounds a subcomplex of DT that represents the shape of S.

The alpha shape has been used for cluster analysis, molecular modeling, and the analysis of medical data, among other applications. High-quality code is available and updated regularly [EF96].

43.5 POLYGON DECOMPOSITION

COVERS AND PARTITIONS

A typical strategy for recognizing a shape as a particular member of a library of shapes is to decompose the shape into "primitive" parts, and then compare these

with the library entries via a similarity function. This has led to considerable effort on decomposing shapes and, in particular, polygons into simpler components.

GLOSSARY

Let P be a polygon.

Cover of P: A collection of sets $S_1, \ldots, S_k$ such that $S_1 \cup \cdots \cup S_k = P$.

Partition of P: A collection of sets $S_1, \ldots, S_k$ with pairwise disjoint interiors such that $S_1 \cup \cdots \cup S_k = P$.

Diagonal of P: A segment s between two vertices x and y of P such that $s \subseteq P$ and $s \cap P = \{x, y\}$.

Steiner point of P: A point of P that is not a vertex.

Polygon with holes: A multiply-connected region bounded by polygonal chains.

Decompositions may be classified along two primary dimensions: covers or partitions, and with or without Steiner points. A cover permits a polygon in the shape of the symbol '+' to be represented as the union of two rectangles, whereas a minimal partition requires three rectangles, a less natural decomposition. Decompositions without Steiner points use diagonals, and are in general easier to find but less parsimonious. For each of the four types of decomposition, different primitives may be considered. The ones most commonly used are rectangles, convex polygons, star-shaped polygons, spiral polygons, and trapezoids (see Section 23.1 for definitions). Restrictions on the shape of the piece being decomposed are often available; for example, orthogonal polygons for rectangle covers. Lastly, the distinction between simple polygons and polygons with holes is often relevant for algorithms.

Finding minimum covers of polygons is NP-hard in nearly every instance explored. Minimum partitions of polygons are somewhat easier, in that polynomial algorithms exist for convex pieces with Steiner points, star-shaped pieces without Steiner points, and rectangles for orthogonal polygons. See Section 23.2 for specific results.

Shape decomposition has a wide variety of applications, including character recognition (spiral and convex pieces), VLSI design (rectangles), and electron-beam lithography (trapezoids) [AAI86].

OPEN PROBLEM

Finding approximation algorithms with proven bounds with respect to optimality remains largely unexplored for most decomposition problems. There are, however, many heuristic algorithms.

SUM-DIFFERENCE DECOMPOSITIONS

Permitting set substraction as well as set union leads to natural shape decompositions. This is evident from the field of Constructive Solid Geometry, where shapes are described with CSG trees whose nodes are union or difference operators, and whose leaves are primitive shapes (Section 47.1). Batchelor developed a similar concept for shape description, the ***convex deficiency tree*** [Bat80]. For a shape P, the root of this tree is its hull $\text{conv}(P)$, the children of the root the hulls of the convex deficiencies $\text{conv}(P) \setminus P$, and so on [O'R94, p. 109].

Chazelle suggested [Cha79] representing a shape by the difference of convex sets: $A \setminus B$ where A and B are unions of convex polygons. It has been established that finding the minimum number of convex pieces in such a sum-difference decomposition of a multiply-connected polygonal region is NP-hard [Con90].

TEXT-BLOCK ISOLATION

The text-block isolation problem consists of extracting blocks of text (paragraphs) from a digitized document. By finding the enclosing rectangles around each connected component (character) and around the entire set of characters we obtain a well structured geometric object, namely, a rectangle with n rectangular "holes." This problem is ideally suited to a computational geometric treatment. Here we mention an elegant method that analyzes the empty (white) spaces in the document [BJF90]. This approach enumerates all maximal white rectangles implied by the black rectangles. A white rectangle is called maximal if it cannot be enlarged while remaining outside the black rectangles. Their enumeration algorithm takes $O(n \log n + m)$ time, where m is the number of maximal rectangles generated in the search. In the worst case $m = O(n^2)$. However, using a clever heuristic to exploit some properties of layouts they obtain $O(n)$ expected time.

43.6 NEAREST-NEIGHBOR DECISION RULES

GLOSSARY

Nearest-neighbor decision rule: Classifies a feature vector with the closest sample point in parameter space.

In the nonparametric classification problem [McL92] we have available a set of n feature (shape measurement) vectors taken from a collected data set of n objects (patterns) denoted by $\{X, \Theta\} = \{(X_1, \theta_1), (X_2, \theta_2), \ldots, (X_n, \theta_n)\}$, where X_i and θ_i denote, respectively, the feature vector on the ith object and the class label of that object. One of the most attractive decision procedures is the *nearest-neighbor rule (NN-rule)* [Dev81]. Let Y be a new object (feature vector) to be classified and let $X_k^* \in \{X_1, X_2, \ldots, X_n\}$ be the feature vector closest to Y. The nearest-neighbor decision rule classifies the unknown object Y into class θ_k^*. In the 1960's and 1970's many pattern recognition practitioners resisted using the NN-rule on the grounds of the mistaken assumptions that (1) all the data $\{X, \Theta\}$ must be stored in order to implement such a rule, and (2) to determine X_k^*, distances must be computed between the unknown vector Y and all members of $\{X_1, X_2, \ldots, X_n\}$. Computational geometric progress in the 1980's and 1990's along with faster and cheaper hardware has made the NN-rule a practical reality.

TRAINING-DATA EDITING RULES

Methods have been developed [TBP85] to edit (delete) "redundant" members of $\{X, \Theta\}$ in order to obtain a subset of $\{X, \Theta\}$ that implements exactly the same

decision boundary as would be obtained using all of $\{X, \Theta\}$. Such methods depend on the computation of Voronoi diagrams and of other proximity graphs that are subgraphs of the Delaunay triangulation, such as the Gabriel graph. Furthermore, the fraction of data discarded in such a method is a useful measure of the resulting reliability of the rule. If few vectors are discarded the feature space is relatively empty and more training data are needed.

NEAREST-NEIGHBOR SEARCHING

Another important issue in the implementation of nearest-neighbor decision rules, whether editing has or has not been performed, concerns the efficient search for the nearest neighbor of an unknown vector in order to classify it. Various methods exist for computing a nearest neighbor without computing distances to all the candidates [FBF77]. In fact, point location techniques (Chapter 30) find the nearest neighbor without computing any distances at all. The most promising algorithms for this problem, difficult when the dimension is high, use either bucketing techniques or k-d trees. For an example of the latest and most practical results concerning nearest-neighbor search in arbitrary dimensions, see [AMN96].

ESTIMATION OF MISCLASSIFICATION

A very important problem in pattern recognition is the estimation of the performance of a decision rule [McL92]. Many geometric problems occur here also, for which computational geometry offers elegant and efficient solutions. For example, a good method of estimating the performance of the NN-rule is to delete each member of $\{X, \Theta\} = \{(X_1, \theta_1), (X_2, \theta_2), \ldots, (X_n, \theta_n)\}$ in turn and classify it with the remaining set. Geometrically this problem reduces to computing for a given set of points in d-space the nearest neighbor of each (the ***all-nearest neighbors problem***). Vaidya [Vai89] gives an $O(n \log n)$ time algorithm to solve this problem.

43.7 SOURCES AND RELATED MATERIAL

SURVEYS

[Tou80a]: A survey of the overlap between pattern recognition and computational geometry.

[Tou91]: A survey of computer vision problems where computational geometry may apply. This survey references several others; the entire collection is of interest as well.

RELATED CHAPTERS

Chapter 20: Voronoi diagrams and Delaunay triangulations
Chapter 23: Polygons
Chapter 30: Point location

REFERENCES

[AAI86] Ta. Asano, Te. Asano, and H. Imai. Partitioning a polygonal region into trapezoids. *J. Assoc. Comput. Mach.*, 33:290–312, 1986.

[AG] H. Alt and L.J. Guibas. Resemblance of geometric objects. In J.-R. Sack and J. Urrutia, editors, *Handbook on Computational Geometry*. North Holland, Amsterdam, to appear.

[AK96] T. Asano and N. Katoh. Variants for the Hough transform for line detection. *Internat. J. Comput. Geom. Appl.*, 6:231–252, 1996.

[AM92] P.K. Agarwal and J. Matoušek. Relative neighborhood graphs in three dimensions. *Comput. Geom. Theory Appl.*, 2:1–14, 1992.

[AMN96] S. Arya, D. Mount, and O. Narayan. Accounting for boundary effects in nearest neighbor searching. *Discrete Comput. Geom.*, 16:155–176, 1996.

[AMWW88] H. Alt, K. Mehlhorn, H. Wagener, and E. Welzl. Congruence, similarity and symmetries of geometric objects. *Discrete Comput. Geom.*, 3:237–256, 1988.

[AT79] S.G. Akl and G.T. Toussaint. Addendum to "An improved algorithm to check for polygon similarity." *Inform. Process. Lett.*, 8:157–158, 1979.

[Bat80] B.G. Batchelor. Hierarchial shape description based upon convex hulls of concavities. *J. Cybern.*, 10:205–210, 1980.

[BJF90] H.S. Baird, S.E. Jones, and S.J. Fortune. Image segmentation by shape-directed covers. In *Proc. 10th IEEE Conf. Pattern Recogn.*, pages 820–825, 1990.

[Blu67] H. Blum. A transformation for extracting new descriptors of shape. In W. Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, 1967.

[BT83] B.K. Bhattacharya and G.T. Toussaint. Efficient algorithms for computing the maximum distance between two finite planar sets. *J. Algorithms*, 4:121–136, 1983.

[CC96] W.S. Chan and F. Chin. Approximation of polygonal curves with minimum number of line segments or minimum error. *Internat. J. Comput. Geom. Appl*, 6:59–77, 1996.

[Cha79] B. Chazelle. *Computational geometry and Convexity*. Ph.D. thesis, Dept. Comput. Sci., Yale Univ., 1979.

[Con90] H. Conn. *Some Polygon Decomposition Problems*. Ph.D. thesis, Dept. Comput. Sci., Johns Hopkins Univ., 1990.

[CSW95] F. Chin, J. Snoeyink, and C.-A. Wang. Finding the medial axis of a simple polygon in linear time. In *Proc. 6th Annu. Internat. Sympos. Algorithms Comput.*, volume 1004 of *Lecture Notes in Comput. Sci.*, pages 382–391. Springer-Verlag, New York, 1995.

[Cyc94] J.M. Cychosz. Efficient binary image thinning using neighborhood maps. In P. Heckbert, editor, *Graphics Gems IV*, pages 465–473. Academic Press, Boston, 1994.

[Dev81] L.P. Devroye. On the inequality of Cover & Hart in nearest neighbour discrimination. *IEEE Trans. Pattern Anal. Mach. Intell.*, 3:75–78, 1981.

[DO94] M. Díaz and J. O'Rourke. Algorithms for computing the center of area of a convex polygon. *Visual Comput.*, 10:432–442, 1994.

[Ead88] P. Eades. Symmetry finding algorithms. In G.T. Toussaint, editor, *Computational Morphology*, pages 41–51. North-Holland, Amsterdam, 1988.

[Ede85] H. Edelsbrunner. Computing the extreme distances between two convex polygons. *J. Algorithms*, 6:213–224, 1985.

[EF96] H. Edesbrunner and P. Fu. `http://alpha.ncsa.uiuc.edu/alpha`. Release 4.0, July, 1996.

[EM94] H. Edelsbrunner and E.P. Mücke. Three-dimensional alpha shapes. *ACM Trans. Graph.*, 13:43–72, 1994.

[ET94] D. Eu and G.T. Toussaint. On approximating polygonal curves in two and three dimensions. *CVGIP: Graph. Models Image Process.*, 56:231–246, 1994.

[FBF77] J.H. Friedman, J.L. Bentley, and R.A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, 3:209–226, 1977.

[GP83] J.E. Goodman and R. Pollack. Multidimensional sorting. *SIAM J. Comput.*, 12:484–507, 1983.

[Gor96] A.D. Gordon. A survey of constrained classification. *Comput. Stat. Data Anal.*, 21:17–29, 1996.

[GPS94] L.J. Guibas, J. Pach, and M. Sharir. Sphere-of-influence graphs in higher dimensions. In K. Böröczky and G. Fejes Tóth, editors, *Intuitive Geometry*, volume 63 of *Colloq. Math. Soc. János Bolyai*, pages 131–137. North-Holland, Amsterdam, 1994.

[Itt93] D.J. Ittner. Automatic inference of textline orientation. In *Proc. 2nd Annu. Sympos. Document Anal. Info. Retrieval*, pages 123–133, 1993.

[JB91] X.-Y. Jiang and H. Bunke. Determination of the symmetries of polyhedra and an application to object recognition. In *Computational Geometry—Methods, Algorithms and Applications*, volume 553 of *Lecture Notes in Comput. Sci.*, pages 113–121. Springer-Verlag, New York, 1991.

[JT92] J.W. Jaromczyk and G.T. Toussaint. Relative neighborhood graphs and their relatives. *Proc. IEEE*, 80:1502–1517, 1992.

[KL95] D. Krznaric and C. Levcopoulos. The first subquadratic algorithm for complete linkage clustering. In *Proc. 6th Annu. Internat. Sympos. Algorithms Comput.*, volume 1004 of *Lecture Notes in Comput. Sci.*, pages 392–401. Springer-Verlag, New York, 1995.

[Lea92] V.F. Leavers. *Shape Detection in Computer Vision using the Hough Transform.* Springer-Verlag, Berlin, 1992.

[Lee82] D.T. Lee. Medial axis transformation of a planar shape. *IEEE Trans. Pattern Anal. Mach. Intell.*, 4:363–369, 1982.

[LW88] Y. Lamdan and H.J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *2nd Internat. Conf. Comput. Vision*, pages 238–249, 1988.

[McL92] G.J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition.* Wiley, New York, 1992.

[MO88] A. Melkman and J. O'Rourke. On polygonal chain approximation. In G.T. Toussaint, editor, *Computational Morphology*, pages 87–95. North-Holland, Amsterdam, 1988.

[O'R94] J. O'Rourke. *Computational Geometry in C.* Cambridge University Press, 1994.

[Rob93] J.-M. Robert. Maximum distance between two sets of points in E^d. *Pattern Recogn. Lett.*, 14, 1993.

[Sch91] T. Schreiber. A Voronoi diagram based adaptive k-means-type clustering algorithm for multidimensional weighted data. In *Computational Geometry—Methods, Algo-*

rithms and Applications, volume 553 of *Lecture Notes in Comput. Sci.*, pages 265–275. Springer-Verlag, New York, 1991.

[SNTM92] V. Srinivasan, L.R. Nackman, J.-M. Tang, and S.N. Meshkat. Automatic mesh generation using the symmetric axis transform of polygonal domains. *Proc. IEEE*, 80:1485–1501, 1992.

[SPB95] E.C. Sherbrooke, N.M. Patrikalakis, and E. Brisson. Computation of the medial axis transform of 3-d polyhedra. In *Proc. 3rd Annu. ACM Sympos. Solid Modeling Appl.*, pages 187–199, 1995.

[Sug84] K. Sugihara. An $n \log n$ algorithm for determining the congruity of polyhedra. *J. Comput. Syst. Sci.*, 29:36–47, 1984.

[Sup83] K.J. Supowit. The relative neighborhood graph with an application to minimum spanning trees. *J. Assoc. Comput. Mach.*, 30:428–448, 1983.

[TBP85] G.T. Toussaint, B.K. Bhattacharya, and R.S. Poulsen. The application of Voronoi diagrams to nonparametric decision rules. In L. Billard, editor, *Computer Science and Statistics: Proc. 16th Sympos. Interface*, pages 97–108, North Holland, Amsterdam, 1985.

[TM82] G.T. Toussaint and M.A. McAlear. A simple $O(n \log n)$ algorithm for finding the maximum distance between two finite planar sets. *Pattern Recogn. Lett.*, 1:21–24, 1982.

[Tou80a] G.T. Toussaint. Pattern recognition and geometrical complexity. In *Proc. 5th IEEE Internat. Conf. Pattern Recogn.*, pages 1324–1347, 1980.

[Tou80b] G.T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recogn.*, 12:261–268, 1980.

[Tou84] G.T. Toussaint. An optimal algorithm for computing the minimum vertex distance between two crossing convex polygons. *Computing*, 32:357–364, 1984.

[Tou91] G.T. Toussaint. Computational geometry and computer vision. In B. Melter, A. Rosenfeld, and P. Bhattacharya, editors, *Vision Geometry*, pages 213–224. Amer. Math. Soc., Providence, 1991.

[Vai89] P.M. Vaidya. An $O(n \log n)$ algorithm for the all-nearest-neighbors problem. *Discrete Comput. Geom.*, 4:101–115, 1989.

[WWV85] J.D. Wolter, T.C. Woo, and R.A. Volz. Optimal algorithms for symmetry detection in two and three dimensions. *Visual Comput.*, 1:37–48, 1985.

[YR91] C. Yao and J.G. Rokne. A straightforward algorithm for computing the medial axis of a simple polygon. *Internat. J. Comput. Math.*, 39:51–60, 1991.

44 GRAPH DRAWING

Roberto Tamassia

INTRODUCTION

Graph drawing addresses the problem of constructing geometric representations of graphs, and has important applications to key computer technologies such as software engineering, database systems, visual interfaces, and computer-aided-design. Research on graph drawing has been conducted within several diverse areas, including discrete mathematics (topological graph theory, geometric graph theory, order theory), algorithmics (graph algorithms, data structures, computational geometry, VLSI), and human-computer interaction (visual languages, graphical user interfaces, software visualization). This chapter overviews aspects of graph drawing that are especially relevant to computational geometry. Basic definitions on drawings and their properties are given in Section 44.1. Bounds on geometric and topological properties of drawings (e.g., area and crossings) are presented in Section 44.2. Section 44.3 deals with the time complexity of fundamental graph drawing problems. An example of a drawing algorithm is given in Section 44.4. General techniques for drawing graphs are surveyed in Section 44.5. Section 44.6 covers selected topics that have recently attracted considerable research interest.

44.1 DRAWINGS AND THEIR PROPERTIES

TYPES OF GRAPHS

First, we define some terminology on graphs pertinent to graph drawing. Throughout let n and m be the number of graph vertices and edges respectively, and d the maximum vertex degree (i.e., number of incident edges).

GLOSSARY

Degree-k graph: Graph with maximum degree $d \leq k$.

Digraph: Directed graph, i.e., graph with directed edges (drawn as arrows).

Acyclic digraph: Without directed cycles.

Transitive edge: Edge (u, v) of a digraph is transitive if there is a directed path from u to v not containing edge (u, v).

Reduced digraph: Without transitive edges.

Source: Vertex of a digraph without incoming edges.

Sink: Vertex of a digraph without outgoing edges.

st-digraph: Acyclic digraph with exactly one source and one sink, which are joined by an edge (also called ***bipolar digraph***).

Connected graph: Any two vertices are joined by a path.

Biconnected graph: Any two vertices are joined by two vertex-disjoint paths.

Triconnected graph: Any two vertices are joined by three (pairwise) vertex-disjoint paths.

Tree: Connected graph without cycles.

Rooted tree: Directed tree with a distinguished vertex, the ***root***, such that each vertex lies on a directed path to the root.

Binary tree: Rooted tree where each vertex has at most two incoming edges.

Layered (di)graph: The vertices are partitioned into sets, called layers. A rooted tree can be viewed as a layered digraph where the layers are sets of vertices at the same distance from the root.

k-layered (di)graph: Layered (di)graph with k layers.

TYPES OF DRAWINGS

In a drawing of a graph, vertices are represented by points (or by geometric figures such as circles or rectangles) and edges are represented by curves such that any two edges intersect at most in a finite number of points. Except for Section 44.6, which covers three-dimensional drawings, we consider drawings in the plane.

GLOSSARY

Polyline drawing: Each edge is a polygonal chain (Figure 44.1.1(a)).

Straight-line drawing: Each edge is a straight-line segment (Figure 44.1.1(b)).

Orthogonal drawing: Each edge is a chain of horizontal and vertical segments (Figure 44.1.1(c)).

Bend: In a polyline drawing, point where two segments belonging to the same edge meet (Figure 44.1.1(a)).

Crossing: Point where two edges intersect (Figure 44.1.1(b)).

Grid drawing: Polyline drawing such that vertices, crossings, and bends have integer coordinates.

Planar drawing: No two edges cross (see Figure 44.1.1(d)).

Planar (di)graph: Admits a planar drawing.

Embedded (di)graph: Planar (di)graph with a prespecified topological embedding (i.e., set of faces), which must be preserved in the drawing.

Upward drawing: Drawing of a digraph where each edge is monotonically non-decreasing in the vertical direction (see Figure 44.1.1(d)).

Upward planar digraph: Admits an upward planar drawing.

Layered drawing: Drawing of a layered graph such that vertices in the same layer lie on the same horizontal line (also called ***hierarchical drawing***).

Face: A region of the plane defined by a planar drawing, where the unbounded region is called the ***external face***.

Convex drawing: Planar straight-line drawing such that the boundary of each face is a convex polygon.

Visibility drawing: Drawing of a graph based on a geometric visibility relation. E.g., the vertices might be drawn as horizontal segments, and the edges associated with vertically visible segments.

Proximity drawing: Drawing of a graph based on a geometric proximity relation. E.g., a tree is drawn as the Euclidean minimum spanning tree of a set of points.

Dominance drawing: Upward drawing of an acyclic digraph such that there exists a directed path from vertex u to vertex v if and only if $x(u) \leq x(v)$ and $y(u) \leq y(v)$, where $x(\cdot)$ and $y(\cdot)$ denote the coordinates of a vertex.

hv-drawing: Upward orthogonal straight-line drawing of a binary tree such that the drawings of the subtrees of each node are separated by a horizontal or vertical line.

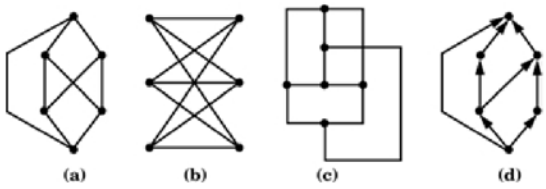


FIGURE 44.1.1
Types of drawings: (a) *polyline drawing of* $K_{3,3}$; (b) *straight-line drawing of* $K_{3,3}$; (c) *orthogonal drawing of* $K_{3,3}$; (d) *planar upward drawing of an acyclic digraph.*

Straight-line and orthogonal drawings are special cases of polyline drawings. Polyline drawings provide great flexibility since they can approximate drawings with curved edges. However, edges with more than two or three bends may be difficult to "follow" for the eye. Also, a system that supports editing of polyline drawings is more complicated than one limited to straight-line drawings. Hence, depending on the application, polyline or straight-line drawings may be preferred. If vertices are represented by points, orthogonal drawings exist only for graphs of maximum vertex degree 4.

PROPERTIES OF DRAWINGS

GLOSSARY

Crossings χ: Total number of crossings in a drawing.

Area: Area of the convex hull of the drawing.

Total edge length: Sum of the lengths of the edges.

Maximum edge length: Maximum length of an edge.

Total number of bends: Total number of bends on the edges of a polyline drawing.

Maximum number of bends: Maximum number of bends on an edge of a polyline drawing.

Angular resolution ρ: Smallest angle formed by two edges, or segments of edges, incident on the same vertex or bend, in a polyline drawing.

Aspect ratio: Ratio of the longest to the shortest side of the smallest rectangle with horizontal and vertical sides covering the drawing.

There are infinitely many drawings for a graph. In drawing a graph, we would like to take into account a variety of properties. For example, planarity and the display of symmetries are highly desirable in visualization applications. Also, it is customary to display trees and acyclic digraphs with upward drawings. In general, to avoid wasting valuable space on a page or a computer screen, it is important to keep the area of the drawing small. In this scenario, many graph drawing problems can be formalized as multi-objective optimization problems (e.g., construct a drawing with minimum area and minimum number of crossings), so that tradeoffs are inherent in solving them. Typically, it is desirable to maximize the angular resolution and to minimize the other measures.

The following examples illustrate two typical tradeoffs in graph drawing problems. Figure 44.1.2(a–b) shows two drawings of K_4, the complete graph on four vertices. The drawing of part (a) is planar, while the drawing of part (b) "maximizes symmetries." It can be shown that no drawing of K_4 is optimal with respect to both criteria, i.e., the maximum number of symmetries cannot be achieved by a planar drawing. Figure 44.1.2(c–d), shows two drawing of the same acyclic digraph G. The drawing of part (c) is upward, while the drawing of part (d) is planar. It can be shown that there is no drawing of G which is both planar and upward.

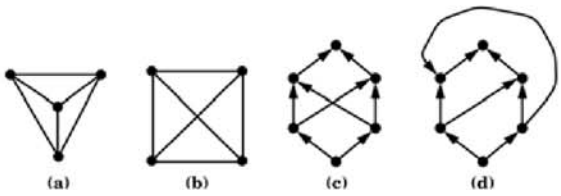


FIGURE 44.1.2
(a–b) *Tradeoff between planarity and symmetry in drawing* K_4. (c–d) *Tradeoff between planarity and upwardness in drawing an acyclic digraph* G.

44.2 BOUNDS ON DRAWING PROPERTIES

For various classes of graphs and drawing types, many universal/existential upper and lower bounds for specific drawing properties have been discovered. Such bounds typically exhibit tradeoffs between drawing properties. A universal bound applies to all the graphs of a given class. An existential bound applies to infinitely many graphs of the class.

Whenever we give bounds on the area or edge length, we assume that the drawing is constrained by some resolution rule that prevents it from being reduced by an arbitrary scaling (e.g., requiring a grid drawing, or stipulating a minimum unit distance between any two vertices).

BOUNDS ON THE AREA

Table 44.2.1 summarizes selected universal upper bounds and existential lower bounds on the area of drawings of graphs. In the table, a is an arbitrary constant $0 \leq a < 1$, and b and c are fixed constants $1 < b < c$. The abbreviations "PSL" and "PSLg" are used for "planar straight-line" "planar straight-line grid," respectively.

TABLE 44.2.1 Universal upper and existential lower bounds on area.

	CLASS OF GRAPHS	DRAWING TYPE	AREA	
1	rooted tree	upward PSLg	$\Omega(n)$	$O(n \log n)$
2	rooted tree	strictly upward PSLg	$\Omega(n \log n)$	$O(n \log n)$
3	degree-$O(n^a)$ rooted tree	upward planar polyline grid	$\Omega(n)$	$O(n)$
4	binary tree	upward planar orthog grid	$\Omega(n \log \log n)$	$O(n \log \log n)$
5	tree	PSLg	$\Omega(n)$	$O(n \log n)$
6	degree-$O(n^a)$ tree	planar polyline grid	$\Omega(n)$	$O(n)$
7	degree-4 tree	planar orthog grid	$\Omega(n)$	$O(n)$
8	planar graph	planar polyline grid	$\Omega(n^2)$	$O(n^2)$
9	planar graph	PSL	$\Omega(c^{\rho n})$	
10	planar graph	PSLg	$\Omega(n^2)$	$O(n^2)$
11	triconnected planar graph	PSL convex grid	$\Omega(n^2)$	$O(n^2)$
12	planar graph	planar orthog grid	$\Omega(n^2)$	$O(n^2)$
13	planar degree-4 graph	orthog grid	$\Omega(n \log n)$	$O(n \log^2 n)$
14	upward planar digraph	upward PSLg	$\Omega(b^n)$	$O(c^n)$
15	reduced planar *st*-digraph	upward PSLg dominance	$\Omega(n^2)$	$O(n^2)$
16	upward planar digraph	upward planar grid polyline	$\Omega(n^2)$	$O(n^2)$
17	general graph	polyline grid	$\Omega(n + \chi)$	$O((n + \chi)^2)$

In general, the effect of bends on the area requirement is dual. On one hand, bends occupy space and hence negatively affect the area. On the other hand, bends may help in routing edges without using additional space.

The following comments apply to Table 44.2.1, where specific rows of the table are indicated within parentheses. Linear or almost-linear bounds on the area can be achieved for trees (1–7). See Table 44.2.4 for tradeoffs between area and aspect ratio in drawings of trees. Planar graphs admit planar drawings with quadratic area (8–13). However, the area requirement of planar straight-line drawings may be exponential if high angular resolution is also desired (9). Almost linear area instead can be achieved in nonplanar drawings of planar graphs (13), which have applications to VLSI circuits. Upward planar drawings provide an interesting tradeoff between area and the total number of bends (14–16). Indeed, unless the digraph is reduced (15), the area can become exponential if a straight-line drawing is required (14). A quadratic area bound is achieved only at the expense of a linear number of bends (16).

BOUNDS ON THE ANGULAR RESOLUTION

Table 44.2.2 summarizes selected universal lower bounds and existential upper bounds on the angular resolution of drawings of graphs. Here c is a fixed constant with $c > 1$.

BOUNDS ON THE NUMBER OF BENDS

Table 44.2.3 summarizes selected universal upper bounds and existential lower bounds on the total and maximum number of bends in orthogonal drawings. Some

TABLE 44.2.2 Universal lower and existential upper bounds on angular resolution.

CLASS OF GRAPHS	DRAWING TYPE	ANGULAR RESOLUTION	
general graph	straight-line	$\Omega(\frac{1}{d^2})$	$O(\frac{\log d}{d^2})$
planar graph	straight-line	$\Omega(\frac{1}{d})$	$O(\frac{1}{d})$
planar graph	planar straight-line	$\Omega(\frac{1}{c^d})$	$O(\sqrt{\frac{\log d}{d^3}})$

bounds are stated for $n \geq 5$ or $n \geq 7$ because the maximum number of bends is at least 2 for K_4 and at least 3 for the skeleton graph of an octahedron, in any planar orthogonal drawing.

TABLE 44.2.3 Orthogonal drawings: universal upper and existential lower bounds on the number of bends. Notes: † $n \geq 7$; ‡ $n \geq 5$.

CLASS OF GRAPHS	DRAWING TYPE	TOTAL # BENDS		MAX # BENDS	
deg-4 †	orthog	$\geq n$	$\leq 2n+2$	≥ 2	≤ 2
planar deg-4 †	orthog planar	$\geq 2n-2$	$\leq 2n+2$	≥ 2	≤ 2
embedded deg-4	orthog planar	$\geq 2n-2$	$\leq \frac{12}{5}n+2$	≥ 3	≤ 3
biconn embedded deg-4	orthog planar	$\geq 2n-2$	$\leq 2n+2$	≥ 3	≤ 3
triconn embedded deg-4	orthog planar	$\geq \frac{4}{3}(n-1)+2$	$\leq \frac{3}{2}n+4$	≥ 2	≤ 2
embedded deg-3 ‡	orthog planar	$\geq \frac{1}{2}n+1$	$\leq \frac{1}{2}n+1$	≥ 1	≤ 1

TRADEOFF BETWEEN AREA AND ASPECT RATIO

The ability to construct area-efficient drawings is essential in practical visualization applications, where screen space is at a premium. However, achieving small area is not enough: e.g., a drawing with high aspect ratio may not be conveniently placed on a workstation screen, even if it has modest area. Hence, it is important to keep the aspect ratio small. Ideally, one would like to obtain small area for any given aspect ratio in a wide range. This would provide graphical user interfaces with the flexibility of fitting drawings into arbitrarily shaped windows.

A variety of tradeoffs for the area and aspect ratio arise even when drawing graphs with a simple structure, such as trees. Table 44.2.4 summarizes selected universal bounds that can be simultaneously achieved on the area and the aspect ratio of various types of drawings of trees. In the table, a is an arbitrary constant with $0 \leq a < 1$.

While upward planar straight-line drawings are the most natural way of visualizing rooted trees, the existing drawing techniques are unsatisfactory with respect to either the area requirement or the aspect ratio. The situation is similar for orthogonal drawings. Regarding polyline drawings, linear area can be achieved with a prescribed aspect ratio. However, experiments show that this is done at the expense of a somehow aesthetically unappealing drawing.

TABLE 44.2.4 Trees: universal upper bounds simultaneously achievable for area and aspect ratio.

CLASS OF GRAPHS	DRAWING TYPE	AREA	ASPECT RATIO
rooted tree	upward PSL layered grid	$O(n^2)$	$O(1)$
rooted tree	upward PSL grid	$O(n \log n)$	$O(n/\log n)$
rooted deg-$O(1)$ tree	upward planar polyline grid	$O(n)$	$O(n^a)$
binary tree	upward planar orthog grid	$O(n \log \log n)$	$O(n \log \log n/\log^2 n)$
deg-4 tree	orthog grid	$O(n)$	$O(1)$
deg-4 tree	orthog grid, leaves on convex hull	$O(n \log n)$	$O(1)$

For nonupward drawings of trees, linear area and optimal aspect ratio are possible for planar orthogonal drawings, and a small (logarithmic) amount of extra area is needed if the leaves are constrained to be on the convex hull of the drawing (e.g., pins on the boundary of a VLSI circuit). However, the nonupward drawing methods do not seem to yield aesthetically pleasing drawings, and are suited more for VLSI layout than for visualization applications.

TRADEOFF BETWEEN AREA AND ANGULAR RESOLUTION

Table 44.2.5 summarizes selected universal bounds that can be simultaneously achieved on the area and the angular resolution of drawings of graphs. Here b and c are fixed constants, $b > 1$ and $c > 1$.

TABLE 44.2.5 Universal upper bounds for area and lower bounds for angular resolution, simultaneously achievable.

CLASS OF GRAPHS	DRAWING TYPE	AREA	ANGULAR RESOLUTION
planar graph	straight-line	$O(d^6 n)$	$\Omega(\frac{1}{d^2})$
planar graph	straight-line	$O(d^3 n)$	$\Omega(\frac{1}{d})$
planar graph	planar straight-line grid	$O(n^2)$	$\Omega(\frac{1}{n^2})$
planar graph	planar straight-line	$O(b^n)$	$\Omega(\frac{1}{c^d})$
planar graph	planar polyline grid	$O(n^2)$	$\Omega(\frac{1}{d})$

Universal lower bounds on the angular resolution exist that depend only on the degree of the graph. Also, substantially better bounds can be achieved by drawing a planar graph with bends or in a nonplanar way.

OPEN PROBLEMS

1. Determine the area requirement of (upward) planar straight-line drawings of trees. There is currently an $O(\log n)$ gap between the known upper and lower bounds (Table 44.2.1).

2. Determine the area requirement of orthogonal (or, more generally, polyline) nonplanar drawings of planar graphs. There is currently an $O(\log n)$ gap between the known upper and lower bounds (Table 44.2.1).

3. Close the wide gap between the $\Omega(\frac{1}{d^2})$ universal lower bound and the $O(\frac{\log d}{d^2})$ existential upper bound on the angular resolution of straight-line drawings of general graphs (Table 44.2.2).

4. Close the gap between the $\Omega(\frac{1}{c^d})$ universal lower bound and the $O(\sqrt{\frac{\log d}{d^3}})$ existential upper bound on the angular resolution of planar straight-line drawings of planar graphs (Table 44.2.2).

5. Determine the best possible aspect ratio and area that can be simultaneously achieved for (upward) planar straight-line and orthogonal drawings of trees (Table 44.2.4).

44.3 COMPLEXITY OF GRAPH DRAWING PROBLEMS

Tables 44.3.1–44.3.3 summarize selected results on the time complexity of some fundamental graph drawing problems.

TABLE 44.3.1 Time complexity of some fundamental graph drawing problems: general graphs and digraphs.

CLASS OF GRAPHS	PROBLEM	TIME COMPLEXITY	
general graph	minimize crossings	NP-hard	
2-layered graph	minimize crossings in layered drawing with preassigned order on one layer	NP-hard	
general graph	maximum planar subgraph	NP-hard	
general graph	planarity testing and computing a planar embedding	$\Omega(n)$	$O(n)$
general graph	maximal planar subgraph	$\Omega(n+m)$	$O(n+m)$
general digraph	upward planarity testing	NP-hard	
embedded digraph	upward planarity testing	$\Omega(n)$	$O(n^2)$
single-source digraph	upward planarity testing	$\Omega(n)$	$O(n)$
general graph	draw as the intersection graph of a set of unit diameter disks in the plane	NP-hard	

It is interesting that apparently similar problems exhibit very different time complexities. For example, while planarity testing can be done in linear time, upward planarity testing is NP-hard. Note that, as illustrated in Figure 44.1.2(c–d), planarity and acyclicity are necessary but not sufficient conditions for upward planarity.

TABLE 44.3.2 Time complexity of some fundamental graph drawing problems: planar graphs and digraphs.

CLASS OF GRAPHS	PROBLEM	TIME COMPLEXITY	
planar graph	planar straight-line drawing with prescribed edge lengths	NP-hard	
planar graph	planar straight-line drawing with maximum angular resolution	NP-hard	
embedded graph	test the existence of a planar straight-line drawing with prescribed angles between pairs of consecutive edges incident on a vertex	NP-hard	
maximal planar graph	test the existence of a planar straight-line drawing with prescribed angles between pairs of consecutive edges incident on a vertex	$\Omega(n)$	$O(n)$
planar graph	planar straight-line grid drawing with $O(n^2)$ area and $O(1/n^2)$ angular resolution	$\Omega(n)$	$O(n)$
planar graph	planar polyline drawing with $O(n^2)$ area, $O(n)$ bends, and $O(1/d)$ angular resolutions	$\Omega(n)$	$O(n)$
triconn planar graph	planar straight-line convex grid drawing with $O(n^2)$ area and $O(1/n^2)$ angular resolution	$\Omega(n)$	$O(n)$
triconn planar graph	planar straight-line strictly convex drawing	$\Omega(n)$	$O(n)$
reduced planar *st*-digraph	upward planar grid straight-line dominance drawing with minimum area	$\Omega(n)$	$O(n)$
upward planar digraph	upward planar polyline grid drawing with $O(n^2)$ area and $O(n)$ bends	$\Omega(n)$	$O(n)$
planar deg-4 graph	planar orthogonal grid drawing with minimum number of bends	NP-hard	
planar deg-3 graph	planar orthogonal grid drawing with minimum number of bends and $O(n^2)$ area	$\Omega(n)$	$O(n^5 \log n)$
embedded deg-4 graph	planar orthogonal grid drawing with minimum number of bends and $O(n^2)$ area	$\Omega(n)$	$O(n^{7/4} \log n)$
planar deg-4 graph	planar orthogonal grid drawing with $O(n^2)$ area and $O(n)$ bends	$\Omega(n)$	$O(n)$

While many efficient algorithms exists for constructing drawings of trees and planar graphs with good universal area bounds, exact area minimization for most types of drawings is NP-hard, even for trees.

OPEN PROBLEMS

1. Reduce the time complexity of upward planarity testing for embedded digraphs (currently $O(n^2)$), or prove a superlinear lower bound (Table 44.3.1).
2. Reduce the time complexity of bend minimization for planar orthogonal drawings of embedded graphs (currently $O(n^{7/4} \log n)$), or prove a superlinear lower bound (Table 44.3.2).

TABLE 44.3.3 Time complexity of some fundamental graph drawing problems: trees.

CLASS OF GRAPHS	PROBLEM	TIME COMPLEXITY	
tree	draw as the Euclidean minimum spanning tree of a set of points in the plane	NP-hard	
degree-4 tree	minimize area in planar orthogonal grid drawing	NP-hard	
degree-4 tree	minimize total/maximum edge length in planar orthogonal grid drawing	NP-hard	
rooted tree	minimize area in a planar straight-line upward layered grid drawing that displays symmetries and isomorphisms of subtrees	NP-hard	
rooted tree	minimize area in a planar straight-line upward layered drawing that displays symmetries and isomorphisms of subtrees	$\Omega(n)$	$O(n^k)$, $k \geq 1$
binary tree	minimize area in hv-drawing	$\Omega(n)$	$O(n\sqrt{n \log n})$
rooted tree	planar straight-line upward layered grid drawing with $O(n^2)$ area	$\Omega(n)$	$O(n)$
rooted tree	planar polyline upward grid drawing with $O(n)$ area	$\Omega(n)$	$O(n)$

44.4 EXAMPLE OF A GRAPH DRAWING ALGORITHM

In this section we outline the algorithm by the author [Tam87] for computing, for an embedded degree-4 graph G, a planar orthogonal grid drawing with minimum number of bends and using $O(n^2)$ area (see Table 44.3.2). This algorithm is the core of a practical drawing algorithm for general graphs (see Section 44.5 and Figure 44.4.1(d)).

The algorithm consists of two main phases:

1. Computation of an *orthogonal shape* for G, where only the bends and the angles of the orthogonal drawing are defined.
2. Assignment of integer lengths to the segments of the orthogonal shape.

Phase 1 uses a transformation into a network flow problem (Figure 44.4.1(a–c)), where each unit of flow is associated with a right angle in the orthogonal drawing. Hence, angles are viewed as a commodity that is produced by the vertices, transported across faces by the edges through their bends, and eventually consumed by the faces.

From the embedded graph G we construct a flow network N as follows. The nodes of network N are the vertices and faces of G. Let $\deg(f)$ denote the number of edges of the circuit bounding face f. Each vertex v supplies $\sigma(v) = 4$ units of flow, and each face f consumes $\tau(f)$ units of flow, where

$$\tau(f) = \begin{cases} 2\deg(f) - 4 & \text{if } f \text{ is an internal face} \\ 2\deg(f) + 4 & \text{if } f \text{ is the external face .} \end{cases}$$

By Euler's formula, $\sum_v \sigma(v) = \sum_f \tau(f)$, i.e., the total supply is equal to the total consumption.

Network N has two types of arcs:

- arcs of the type (v, f), where f is a face incident on vertex v; the flow in (v, f) represents the angle at vertex v in face f, and has lower bound 1, upper bound 4, and cost 0;
- arcs of the type (f, g), where face f shares an edge e with face g; the flow in (f, g) represents the number of bends along edge e with the right angle inside face f, and has lower bound 0, upper bound $+\infty$, and cost 1.

The conservation of flow at the vertices expresses the fact that the sum of the angles around a vertex is equal to 2π. The conservation of flow at the faces expresses the fact that the sum of the angles at the vertices and bends of an internal face is equal to $\pi(p-2)$, where p is the number of such angles. For the external face, the above sum is equal to $\pi(p+2)$.

It can be shown that every feasible flow ϕ in network N corresponds to an admissible orthogonal shape for graph G, whose number of bends is equal to the cost of flow ϕ. Hence, an orthogonal shape for G with the minimum number of bends can be computed from a minimum cost flow in G. This flow can be constructed in $O(n^2 \log n)$ time with standard flow-augmentation methods.

Phase 2 uses a simple compaction strategy derived from VLSI layout, where the lengths of the horizontal and vertical segments are computed independently after a preliminary refinement of the orthogonal shape that decomposes each face into rectangles. The resulting drawing is shown in Figure 44.4.1(d).

44.5 TECHNIQUES FOR DRAWING GRAPHS

In this section we outline some of the most successful techniques that have been devised for drawing general graphs.

PLANARIZATION

The planarization approach is motivated by the availability of many efficient and well-analyzed drawing algorithms for planar graphs (see Table 44.3.2). If the graph is nonplanar, it is transformed into a planar graph by means of a preliminary planarization step that replaces each crossing with a fictitious vertex. Finding the minimum number of crossings or a maximum planar subgraph are NP-hard problems. Hence, existing planarization algorithms use heuristics.

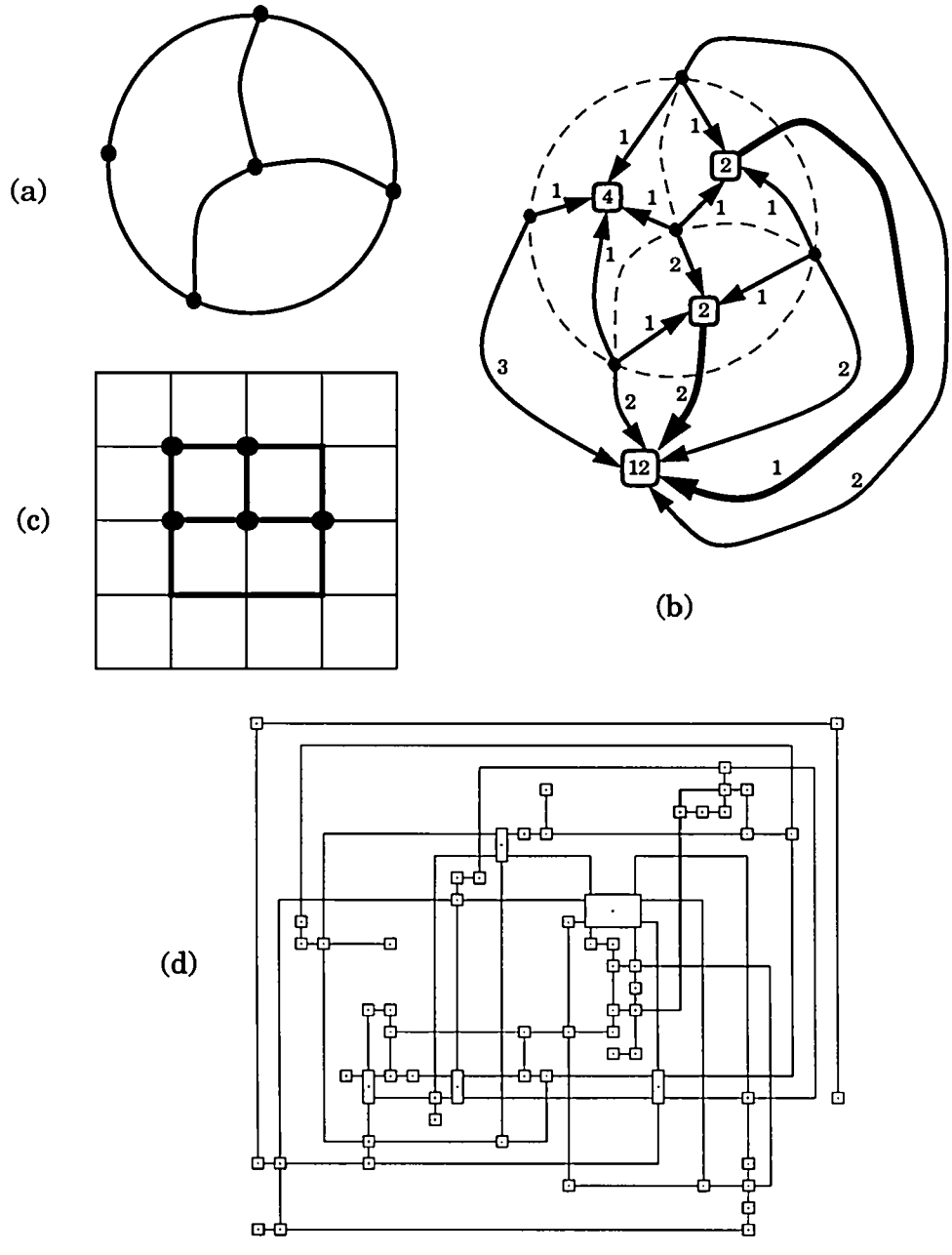
The best available heuristic for the maximum planar subgraph problem is described in [JM96]. This method has a solid theoretical foundation in polyhedral combinatorics, and achieves good results in practice.

A successful drawing algorithm based on the planarization approach and a bend-minimization method [Tam87] is described in [TDB88] (Figure 44.4.1(d) was generated by this algorithm). It has been widely used in software visualization systems.

LAYERING

The layering approach for constructing polyline drawings of directed graphs transforms the digraph into a layered digraph and then constructs a layered drawing.

FIGURE 44.4.1
(a) *Embedded graph G.* (b) *Minimum cost flow in network N: the flow is shown next to each arc; arcs with zero flow are omitted; arcs with unit cost are drawn with thick lines; a face f is represented by a box labeled with $\tau(f)$.* (c) *Planar orthogonal grid drawing of G with minimum number of bends.* (d) *Orthogonal grid drawing of a nonplanar graph produced by a drawing method for general graphs based on the algorithm of Section* 44.4.



A typical algorithm based on the layering approach consists of the following main steps:

1. Assign each vertex to a layer, with the goal of maximizing the number of edges oriented upward.
2. Insert fictitious vertices along the edges that cross layers, so that each edge in the resulting digraph connects vertices in consecutive layers. (The fictitious vertices will be displayed as bends in the final drawing.)
3. Permute the vertices on each layer with the goal of minimizing crossings.
4. Adjust the positions of the vertices in each layer with the goal of distributing the vertices uniformly and minimizing the number of bends.

Most of the subproblems involved in the various steps are NP-hard, hence heuristics must be used. The layering approach was pioneered by Sugiyama et al. [STT81]. The most notable developments of this technique are due to Gansner et al. [GNV88, GKNV93].

PHYSICAL SIMULATION

This approach uses a physical model where the vertices and edges of the graph are viewed as objects subject to various forces. Starting from an initial random configuration, the physical system evolves into a final configuration of minimum energy, which yields the drawing. Rather than solving a system of differential equations, the evolution of the system is usually simulated using numerical methods (e.g., at each step, the forces are computed and corresponding incremental displacements of the vertices are performed). Drawing algorithms based on the physical simulation approach are often able to detect and display symmetries in the graph. However, their running time is typically high.

The physical simulation approach was pioneered in [Ead84, KS80]. Recent sophisticated developments include [DH96, FR91].

44.6 RECENT RESEARCH TRENDS

In this section, we present an overview of selected areas of graph drawing that have recently attracted increasing attention.

THREE-DIMENSIONAL DRAWINGS

Recent advances in hardware and software technology for computer graphics open the possibility of displaying three-dimensional (3D) visualizations on a variety of low-cost workstations. Previous research on 3D graph drawing has focused on the development of visualization systems.

Much work needs to be done on the theoretical foundations of 3D graph drawing. Recent progress is reported in [CGT96, CELR95, LD95].

PROXIMITY AND VISIBILITY DRAWINGS

The geometric representation of graphs by means of the vertical visibility relation among horizontal segments is well understood (see, e.g., [TT86] and Section 25.2). It has applications to motion planning and VLSI layout. Recent progress on visibility drawings has been reported in [AGW96]. See also Section 25.2.

Proximity drawings include Gabriel, relative neighborhood, Delaunay, sphere of influence, and minimum spanning drawings (see Section 43.2 for definitions). Increasing attention has been devoted to the problem of characterizing the classes of graphs that admit various types of proximity drawings. A survey of this area appears in [DLL95].

DECLARATIVE METHODS

Research in graph drawing has traditionally focused on algorithmic methods, where the drawing of the graph is generated according to a prespecified set of aesthetic criteria (such as planarity or area minimization) that are embodied in an algorithm. Although the algorithmic approach is computationally efficient, it does not naturally support constraints, i.e., requirements that the user may want to impose on the drawing of a specific graph (e.g., clustering, or aligning a given set of vertices). Previous work has shown that only a rather limited constraint satisfaction capability can be added to existing drawing algorithms (see, e.g., [TDB88]).

Recently, several attempts have been made at developing languages for the specification of constraints and at devising techniques for graph drawing based on the resolution of systems of constraints (see, e.g., [Kam89]). A visual approach to graph drawing, where the layout of a graph is pictorially specified "by example," is proposed by Cruz and Garg [CG95]. Recent work by Eades and Lin [LE95] also makes an attempt at combining algorithmic and declarative methods. Brandenburg presents a comprehensive approach to graph drawing based on graph grammars [Bra95].

Related work includes the physical simulation approach (Section 44.5) and genetic algorithms [KMS94].

DYNAMIC GRAPH DRAWING

Many graphic interfaces provide graph editors that allow the user to interactively modify the drawing of a graph. After an editing session, it would be useful to have an algorithm that "beautifies" the drawing with smooth modifications that preserve the user's "mental map" of the diagram [ELMS91].

This scenario motivates the development of dynamic algorithms that make incremental changes in a drawing of a graph subject to a sequence of update operations, such as insertion and deletion of vertices and edges. Tradeoffs between running time, optimization of the drawing properties, and preservation of the mental map are typical issues to be addressed in dynamic graph drawing.

Most of the existing results on dynamic graph drawing are limited to planar graphs (see, e.g., [CDTT95]).

EXPERIMENTATION

Many graph drawing algorithms have been implemented and used in practical applications. Most papers show sample outputs, and some also provide limited experimental results on small test suites. However, in order to evaluate the practical performance of a graph drawing algorithm in visualization applications, it is essential to perform extensive experimentations with input graphs derived from the application domain.

The performance of four planar straight-line drawing algorithms on 10,000 randomly generated maximal planar graphs is compared by Jones et al. [JEM+91].

Himsolt [Him95a] presents a comparative study of twelve graph drawings algorithms based on various approaches. The experiments are conducted on 100 sample graphs with the graph drawing system *GraphEd* [Him95b]. Many examples of drawings constructed by the algorithms are shown, and various objective and subjective evaluations on the aesthetic quality of the drawings produced are given.

Di Battista et al. [DGL+95] report on an extensive experimental study comparing three orthogonal drawing algorithms based on the planarization approach. The test data are 11,582 graphs, ranging from 10 to 100 vertices, which are generated from a core set of 112 graphs used in "real-life" software engineering and database applications. The experiments are conducted with *Diagram Server* [DGST90, DLV95] (a graph drawing system based on a client-server architecture) and provide a detailed quantitative evaluation of the performance of the three algorithms.

The observed practical behavior of the three algorithms was consistent with their theoretical properties. Namely, the algorithm of [TDB88], based on the planarization approach and a bend-minimization method [Tam87] (see Section 44.4 and Figure 44.4.1(d)), clearly outperformed the other two algorithms, based on linear-time heuristic methods, for most quality measures (including crossings, area, and bends); but it was considerably slower.

44.7 SOURCES AND RELATED MATERIAL

A comprehensive bibliography on graph drawing algorithms [DETT94] cites more than 300 papers written before 1993. Most papers on graph drawing are cited in `geom.bib`, the computational geometry bibliography [O'R93] available from `ftp://cs.usask.ca/pub/geometry/` (search for keyword "graph drawing"). Surveys on various aspects of graph drawing appear in [DLL95, EL89, GT95, HK95, Riv89, Riv93, SSV95, Tam90a, Tam90b]. The author maintains a WWW page (`http://www.cs.brown.edu/people/rt/gd.html`) with pointers to graph drawing resources on the Web.

The proceedings of the annual Symposium on *Graph Drawing* are published by Springer-Verlag in the *Lecture Notes in Computer Science* series [TT95, Bra96]. Special issues dedicated to graph drawing have appeared or will appear in *Algorithmica* [DT96], *Computational Geometry: Theory and Applications* [DT], and the *Journal of Visual Languages and Computing* [CE95].

RELATED CHAPTERS

Chapter 25: Visibility

REFERENCES

[AGW96] H. Alt, M. Godau, and S. Whitesides. Universal 3-dimensional visibility representations for graphs. In F.J. Brandenburg, editor, *Graph Drawing* (Proc. GD '95), volume 1027 of *Lecture Notes in Comput. Sci.*, pages 8–19. Springer-Verlag, Berlin, 1996.

[Bra95] F.J. Brandenburg. Designing graph drawings by layout graph grammars. In R. Tamassia and I.G. Tollis, editors, *Graph Drawing* (Proc. GD '94), volume 894 of *Lecture Notes in Comput. Sci.*, pages 416–427. Springer-Verlag, Berlin, 1995.

[Bra96] F.J. Brandenburg, editor. *Graph Drawing* (Proc. GD '95), volume 1027 of *Lecture Notes in Comput. Sci.*. Springer-Verlag, Berlin, 1996.

[CDTT95] R.F. Cohen, G. Di Battista, R. Tamassia, and I.G. Tollis. Dynamic graph drawing: Trees, series-parallel digraphs, and planar *st*-digraphs. *SIAM J. Comput.*, 24:970–1001, 1995.

[CE95] I.F. Cruz and P. Eades. (Special issue on graph visualization.) *J. Visual Lang. Comput.*, 6(3), 1995.

[CELR95] R.F. Cohen, P. Eades, T. Lin, and F. Ruskey. Three-dimensional graph drawing. In R. Tamassia and I.G. Tollis, editors, *Graph Drawing* (Proc. GD '94), volume 894 of *Lecture Notes in Comput. Sci.*, pages 1–11. Springer-Verlag, Berlin, 1995.

[CG95] I.F. Cruz and A. Garg. Drawing graphs by example efficiently: Trees and planar acyclic digraphs. In R. Tamassia and I.G. Tollis, editors, *Graph Drawing* (Proc. GD '94), volume 894 of *Lecture Notes in Comput. Sci.*, pages 404–415. Springer-Verlag, Berlin, 1995.

[CGT96] M. Chrobak, M.T. Goodrich, and R. Tamassia. Convex drawings of graphs in two and three dimensions. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 319–328, 1996.

[DETT94] G. Di Battista, P. Eades, R. Tamassia, and I.G. Tollis. Algorithms for drawing graphs: an annotated bibliography. *Comput. Geom. Theory Appl.*, 4:235–282, 1994.

[DGL+95] G. Di Battista, A. Garg, G. Liotta, R. Tamassia, E. Tassinari, and F. Vargiu. An experimental comparison of three graph drawing algorithms. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 306–315, 1995.

[DGST90] G. Di Battista, A. Giammarco, G. Santucci, and R. Tamassia. The architecture of Diagram Server. In *Proc. IEEE Workshop Visual Lang.*, pages 60–65, 1990.

[DH96] R. Davidson and D. Harel. Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics*, 15:301–331, 1996.

[DLL95] G. Di Battista, W. Lenhart, and G. Liotta. Proximity drawability: a survey. In R. Tamassia and I.G. Tollis, editors, *Graph Drawing* (Proc. GD '94), volume 894 of *Lecture Notes in Comput. Sci.*, pages 328–339. Springer-Verlag, Berlin, 1995.

[DLV95] G. Di Battista, G. Liotta, and F. Vargiu. Diagram Server. *J. Visual Lang. Comput.*, 6:275–298, 1995.

[DT96] G. Di Battista and R. Tamassia. (Special issue on graph drawing.) *Algorithmica*, 16(1), 1996.

[DT] G. Di Battista and R. Tamassia. (Special issue on graph drawing.) *Comput. Geom. Theory Appl.*, to appear.

[Ead84] P. Eades. A heuristic for graph drawing. *Congr. Numer.*, 42:149–160, 1984.

[EL89] P. Eades and X. Lin. How to draw a directed graph. In *Proc. IEEE Workshop Visual Lang.*, pages 13–17, 1989.

[ELMS91] P. Eades, W. Lai, K. Misue, and K. Sugiyama. Preserving the mental map of a diagram. In *Proc. Compugraphics '91*, pages 24–33, 1991.

[FR91] T. Fruchterman and E. Reingold. Graph drawing by force-directed placement. *Softw.–Pract. Exp.*, 21:1129–1164, 1991.

[GKNV93] E.R. Gansner, E. Koutsofios, S.C. North, and K.P. Vo. A technique for drawing directed graphs. *IEEE Trans. Softw. Eng.*, 19:214–230, 1993.

[GNV88] E.R. Gansner, S.C. North, and K.P. Vo. DAG—A program that draws directed graphs. *Softw.–Pract. Exp.*, 18:1047–1062, 1988.

[GT95] A. Garg and R. Tamassia. Upward planarity testing. *Order*, 12:109–133, 1995.

[Him95a] M. Himsolt. Comparing and evaluating layout algorithms within GraphEd. *J. Visual Lang. Comput.*, 6:255–273, 1995.

[Him95b] M. Himsolt. GraphEd: A graphical platform for the implementation of graph algorithms. In R. Tamassia and I.G. Tollis, editors, *Graph Drawing* (Proc. GD '94), volume 894 of *Lecture Notes in Comput. Sci.*, pages 182–193. Springer-Verlag, Berlin, 1995.

[HK95] X. He and M.-Y. Kao. Regular edge labelings and drawings of planar graphs. In R. Tamassia and I.G. Tollis, editors, *Graph Drawing* (Proc. GD '94), volume 894 of *Lecture Notes in Comput. Sci.*, pages 96–103. Springer-Verlag, Berlin, 1995.

[JEM+91] S. Jones, P. Eades, A. Moran, N. Ward, G. Delott, and R. Tamassia. A note on planar graph drawing algorithms. Tech. Rep. 216, Dept. of Comput. Sci., Univ. Queensland, 1991.

[JM96] M. Jünger and P. Mutzel. Maximum planar subgraphs and nice embeddings: Practical layout tools. *Algorithmica*, 16:33–59, 1996.

[Kam89] T. Kamada. *Visualizing Abstract Objects and Relations.* World Scientific, Singapore, 1989.

[KMS94] C. Kosak, J. Marks, and S. Shieber. Automating the layout of network diagrams with specified visual organization. *IEEE Trans. Syst. Man Cybern.*, 24:440–454, 1994.

[KS80] J.B. Kruskal and J.B. Seery. Designing network diagrams. In *Proc. 1st Gen. Conf. Social Graph.*, pages 22–50. U.S. Dept. Census, 1980.

[LD95] G. Liotta and G. Di Battista. Computing proximity drawings of trees in the 3-dimensional space. In *Proc. 4th Workshop Algorithms Data Struct.*, volume 955 of *Lecture Notes in Comput. Sci.*, pages 239–250. Springer-Verlag, New York, 1995.

[LE95] T. Lin and P. Eades. Integration of declarative and algorithmic approaches for layout creation. In R. Tamassia and I.G. Tollis, editors, *Graph Drawing* (Proc. GD '94), volume 894 of *Lecture Notes in Comput. Sci.*, pages 376–387. Springer-Verlag, Berlin, 1995.

[O'R93] J. O'Rourke. Computational geometry column 19. *Internat. J. Comput. Geom. Appl.*, 3:221–224, 1993. Also *SIGACT News*, 24(2):15–17, 1993.

[Riv89] I. Rival. Graphical data structures for ordered sets. In I. Rival, editor, *Algorithms and Order*, pages 3–31. Kluwer, Boston, 1989.

[Riv93] I. Rival. Reading, drawing, and order. In I.G. Rosenberg and G. Sabidussi, editors, *Algebras and Orders*, pages 359–404. Kluwer, Boston, 1993.

[SSV95] F. Shahrokhi, L.A. Székely, and I. Vrt'o. Crossing numbers of graphs, lower bound techniques and algorithms: a survey. In R. Tamassia and I.G. Tollis, editors, *Graph Drawing* (Proc. GD '94), volume 894 of *Lecture Notes in Comput. Sci.*, pages 131–142. Springer-Verlag, Berlin, 1995.

[STT81] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical systems. *IEEE Trans. Syst. Man Cybern.*, 11:109–125, 1981.

[Tam87] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16:421–444, 1987.

[Tam90a] R. Tamassia. Drawing algorithms for planar st-graphs. *Australasian J. Combin.*, 2:217–235, 1990.

[Tam90b] R. Tamassia. Planar orthogonal drawings of graphs. In *Proc. IEEE Internat. Sympos. Circuits Syst.*, 1990.

[TDB88] R. Tamassia, G. Di Battista, and C. Batini. Automatic graph drawing and readability of diagrams. *IEEE Trans. Syst. Man Cybern.*, 18:61–79, 1988.

[TT86] R. Tamassia and I.G. Tollis. A unified approach to visibility representations of planar graphs. *Discrete Comput. Geom.*, 1:321–341, 1986.

[TT95] R. Tamassia and I.G. Tollis, editors. *Graph Drawing* (Proc. GD '94), volume 894 of *Lecture Notes in Comput. Sci.*. Springer-Verlag, Berlin, 1995.

45 SPLINES AND GEOMETRIC MODELING

Chandrajit L. Bajaj and Susan Evans

INTRODUCTION

Piecewise polynomials of fixed degree and continuously differentiable up to some order are known as ***splines*** or ***finite elements***. Splines are used in applications ranging from computer-aided design, computer graphics, data visualization, geometric modeling, and image processing to the solution of partial differential equations via finite element analysis. The spline-fitting problem of constructing a mesh of finite elements that interpolate or approximate multivariate data is by far the primary research problem in geometric modeling. ***Parametric splines*** are vectors of a set of multivariate polynomial (or rational) functions while ***implicit splines*** are zero contours of collections of multivariate polynomials. This chapter dwells mainly on spline surface fitting methods in real Euclidean space. We first discuss tensor product surfaces (Section 45.1), perhaps the most popular. The next sections cover generalized spline surfaces (Section 45.2), free-form surfaces (Section 45.3), and subdivision surfaces (Section 45.4). This classification is not strict, and some overlap exists. Interactive editing of surfaces is discussed in the final section (Section 45.5).

The various spline methods may be distinguished by several criteria:

- Implicit or parametric representations.
- Algebraic and geometric degree of the spline basis.
- Number of surface patches required.
- Computation (time) and memory (space) required.
- Stability of fitting algorithms.
- Local or nonlocal interpolation.
- Splitting or nonsplitting of input mesh.
- Convexity or nonconvexity of the input and solution.
- Fairness of the solution (first- and second-order variation).

These distinctions will guide the discussions throughout the chapter.

45.1 TENSOR PRODUCT SURFACES

Tensor product B-splines have emerged as the polynomial basis of choice for working with parametric surfaces. The theory of tensor product patches requires that data have a rectangular geometry and that the parametrizations of opposite boundary curves be similar. It is based on the concept of bilinear interpolation. The most general results obtained to date are summarized in Table 45.1.1, and will be discussed below.

GLOSSARY

Affine invariance: A property of a curve or surface generation scheme, implying invariance with respect to whether computation of a point on a curve or surface occurs before or after an affine map is applied to the input data.

A-spline: Collection of bivariate Bernstein-Bézier polynomials, each over a triangle and with prescribed geometric continuity, such that the zero contour of each polynomial defines a smooth and single-sheeted real algebraic curve segment. ("A" stands for "algebraic.")

A-patch: Smooth and "functional" zero contour of a Bernstein-Bézier polynomial over a tetrahedron.

Barycentric combination: A weighted average where the sum of the weights equals one.

Barycentric coordinates: A point in $\mathbb{R}^2$ may be written as a unique barycentric combination of three points. The coefficients in this combination are its barycentric coordinates. Similarly, a point in $\mathbb{R}^3$ may be written as a unique barycentric combination of four points. See Figure 28.2.1.

Basis function: Functions form linear spaces, which have bases. The elements of these bases are the basis functions.

Bernstein-Bézier form: Let $p_1,\ p_2,\ p_3,\ p_4 \in \mathbb{R}^3$ be affinely independent. Then the tetrahedron with these points as vertices is $V = [p_1 p_2 p_3 p_4]$. Any polynomial $f(p)$ of degree n can be expressed in the Bernstein-Bézier (BB) form over V as

$$f(p) = \sum_{|\lambda|=n} b_\lambda \; B_\lambda^n(\alpha),\ \lambda \in \mathcal{Z}_+^4\ , \tag{45.1.1}$$

where

$$B_\lambda^n(\alpha) = \frac{n!}{\lambda_1!\lambda_2!\lambda_3!\lambda_4!}\ \alpha_1^{\lambda_1}\alpha_2^{\lambda_2}\alpha_3^{\lambda_3}\alpha_4^{\lambda_4}$$

are Bernstein polynomials, $|\lambda| = \sum_{i=1}^4 \lambda_i$ with $\lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)^T$, the barycentric coordinates of p are $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)^T$, $b_\lambda = b_{\lambda_1\lambda_2\lambda_3\lambda_4}$ are the *control points*, and $\mathcal{Z}_+^4$ is the set of all four-dimensional vectors with nonnegative integer components.

Bernstein polynomials: The basis functions for Bézier curves and surfaces.

Bézier curve: A curve whose points are determined by the parameter u in the equation $\sum_{i=0}^n B_i^n(u)P_i$, where the $B_i^n(u)$ are basis functions, and the P_i control points.

Bilinear interpolation: A tensor product of two orthogonal linear interpolants and the "simplest" surface defined by values at four points on a rectangle.

Blending functions: The basis functions used by interpolation schemes such as Gordon surfaces.

B-spline surface: Traditionally, a tensor product of curves defined using piecewise basis polynomials (B-spline basis). Any B-spline can be written in piecewise Bézier form. ("B" stands for "basis.")

C^k continuity: Smoothness defined in terms of matching of up to kth order derivatives along patch boundaries.

Control point: The coefficients in the expansion of a Bézier curve in terms of Bernstein polynomials.

Convex hull: The smallest convex set that contains a given set.

Convex set: A set such that the straight line segment connecting any two points of the set is completely contained within the set.

G^k continuity: Geometric continuity with smoothness defined in terms of matching of up to kth order derivatives allowing for reparametrization. For example, G^1 smoothness is defined in terms of matching tangent planes along patch boundaries.

Knots: A spline curve is defined over a partition of an interval of the real line. The points that define the partition are called knots.

Mesh: A decomposition of a geometric domain into finite elements; see Section 22.4.

Ruled (lofted) surface: A surface that interpolates two given curves using linear interpolation.

Tensor product surfaces: A surface represented with basis functions that are constructed as products of univariate basis functions. A tensor product Bézier surface is given by the equation $\sum_{i=0}^{n}\sum_{j=0}^{m} B_i^n(u)B_j^m(v)P_{ij}$, where the $B_i^n(u)$ and $B_j^m(v)$ are the univariate Bernstein polynomial basis functions, and the P_{ij} are control points.

Transfinite interpolation: Interpolating entire curves as opposed to values at discrete points.

Variation diminishing: A curve or surface scheme has this property if its output "wiggles less" than the control points from which it is constructed.

PARAMETRIC BÉZIER AND B-SPLINES

Tensor product Bézier surfaces are obtained by repeated applications of bilinear interpolation. Properties of tensor product Bézier patches include affine invariance, the "convex hull property," and the variation diminishing property. The boundary curves of a patch are polynomial curves that have their Bézier polygon given by the boundary polygons of the control net of the patch. Hence the four corners of the control net lie on the patch.

Piecewise bicubic Bézier patches may be used to fit a C^1 surface through a rectangular grid of points. After the rectangular network of curves has been created, there are four coefficients left to determine the corner twists of each patch. These four corner twists cannot be specified independently and must satisfy a "compatibility constraint." Common twist estimation methods include zero twists, Adini's twist, Bessel twist, and Brunet's twist [Far93]. To obtain C^1 continuity between two patches the directions and lengths of the polyhedron edges must be matched across the common polyhedron boundary defining the common boundary curve. Piecewise bicubic Hermite patches are similar to the piecewise bicubic Bézier patches,

TABLE 45.1.1 Tensor product surfaces.

TYPE	INPUT	PROPERTIES
Piecewise Bézier and Hermite	rectangular grid of points, corner twists	C^1, initial global data survey data to determine the tangent and cross-derivative vectors at patch corners
Bicubic B-spline	rectangular grid of points	C^1
Coons patches	4 boundary curves	C^1
Gordon surfaces	rectangular network of curves	C^1, Gregory square
Biquadratic B-spline	limit of Doo-Sabin subdivision of rectangular faces	C^1
Bicubic B-spline	limit of Catmull-Clark subdivision of rectangular faces	C^1
Biquadratic splines	control points on mesh with arbitrary topology	G^1, system of linear equations for smoothness conditions around singular vertices
Biquartic splines	cubic curve mesh	C^1, interpolate second-order data at mesh points
Bisextic B-spline	rectangular network of cubic curves	C^1
Triquadratic/tricubic A-patches	rectilinear 3D grid points	C^1, local calculation of first-order cross derivatives
Triple products of B-splines	rectangular boxes	

but take points, partials, and mixed partials as input. The mixed partials affect only the interior shape of the patch, and are also called ***twist vectors***.

It is not possible to model a general closed surface or a surface with handles as a single nondegenerate B-spline. To represent free-form surfaces a significant amount of recent work has been done in the areas of geometric continuity, nontensor product patches, and generalizing B-splines. Common schemes include splitting, convex combinations of blending functions, subdivision, and local interpolation by construction.

IMPLICIT BÉZIER AND B-SPLINES

Patrikalakis and Kriezis [PK89] demonstrate how implicit algebraic surfaces can be manipulated in rectangular boxes as functions in a tensor product B-spline basis. This work, however, leaves open the problem of selecting weights or specifying knot sequences for C^1 meshes of tensor product implicit algebraic surface patches that fit given spatial data. Moore and Warren [MW91] extend the "marching cubes" scheme to compute a C^1 piecewise tensor product triquadratic approximation to scattered data using a Powell-Sabin-like split over subcubes. In [BBX95] an incremental and adaptive approach is used to construct C^1 spline functions defined over an octree subdivision that approximate a dense set of multiple volumetric scattered scalar values.

COONS PATCHES AND GORDON SURFACES

Coons patches interpolate four boundary curves. They are constructed by composing two ruled, or lofted, surfaces and one bilinear surface, and hence are called ***bilinearly blended surfaces***. A Coons patch has four blending functions $f_i(u)$, $g_i(v)$, $i = 1, 2$. There are only two restrictions on the f_i and g_i: each pair must sum to one, and we must have $f_1(0) = g_1(0) = 1$ and $f_2(1) = g_2(1) = 0$ in order to interpolate.

A network of curves may be filled in with a C^1 surface using bicubically blended Coons patches. For this the four twists at the data points and the four cross boundary derivatives must be computed. Compatibility problems may arise in computing the twists. If $\mathbf{x}(u, v)$ is twice differentiable, we have $\mathbf{x}_{uv} = \mathbf{x}_{vu}$, but this simplification does not apply here. One approach is to adjust the given data so that the incompatibilities disappear. Or if the data cannot be changed one can use a method known as ***Gregory's square*** that replaces the constant twist terms by variable twists that are computed from the cross boundary derivatives. The resulting surface does not have continuous twists at the corners and is rational parametric, which may not be acceptable geometry for certain geometric modeling systems.

Gordon surfaces are a generalization of Coons patches used to construct a surface that interpolates a rectangular network of curves. The idea is to take a univariate interpolation scheme, apply it to all curves, add the resulting surfaces, and subtract the tensor product interpolant that is defined by the univariate scheme. Polynomial interpolation or spline interpolation schemes may be used. Methods for Coons patches and Gordon surfaces can be formulated in terms of Boolean sums and projectors. This has also been generalized to create triangular Coons patches.

45.2 GENERALIZED SPLINE SURFACES

B-PATCHES

The B-patches developed by Seidel [Sei89, DMS92] are based on the study of symmetric recursive evaluation algorithms, and are defined by generalizing the deBoor algorithm for the evaluation of a B-spline segment from curves to surfaces. A polynomial surface that has a symmetric recursive evaluation algorithm is called a ***B-patch***. B-patches generalize Bézier patches over triangles, and are characterized by control points and a three-parameter family of knots. Every bivariate polynomial $F : \mathbb{R}^2 \to \mathbb{R}^d$ of degree n has a unique representation

$$F(U) = \sum_{|\vec{i}|=n} N^n_{\vec{i}}(U) P_{\vec{i}}, \qquad P_{\vec{i}} \in \mathbb{R}^d$$

as a B-patch, with parameters $\mathcal{K} = R_0, \ldots, R_{n-1}, S_0, \ldots, S_{n-1}, T_0, \ldots, T_{n-1}$ in $\mathbb{R}^2$, if the parameters (R_i, S_j, T_k) are affinely independent for $0 \leq |\vec{i}| \leq n - 1$. The

real-valued polynomials $N_i^n(U)$ are called the ***normalized B-weights*** of degree n over $\mathcal{K}$.

MULTISIDED PATCHES

Multisided patches can be generated in basically two ways. Either the polygonal domain which is to be mapped into $\mathbb{R}^3$ is subdivided in the parametric plane, or one uniform equation is used as a combination of equations. In the former case, triangular or rectangular elements are put together or recursive subdivision is applied. In the latter case, either the known control point methods are generalized, or a weighted sum of interpolants is used. With constrained domain mapping, a domain point for an n-sided patch is represented by n dependent parameters. If the remainder of the parameters can be computed when any two parameters are independently chosen, it is called a *symmetric system of parameters*. The main results from multisided patch schemes obtained to date are summarized in Table 45.2.1.

TABLE 45.2.1 Multisided schemes.

TYPE	LIMITATIONS	PROPERTIES	DOMAIN POINTS
Sabin	n=3,5	C^1	constrained domain mapping, symmetric system of parameters
Gregory/Charrot	n=3,5	C^1	barycentric coordinates
Hosaka/Kimura	$n \leq 6$	C^1	constrained domain mapping, symmetric system of parameters
Varady		VC^1	$2n$ variables constrained along polygon sides
Base points	$n = 4, 5, 6$	rational Bézeir surfaces	base points in the parametric domain map to rational curves in $\mathbb{R}^3$
S-patches	multisided	G^1 rational bi-quadratic and bicubic B-splines	embed n-sided domain polygon into simplex of dimension $n-1$
Multisided A-patches	"functional" bd curves	C^1, C^2 implicit Bezier surfaces	Hermite interpolation of boundary curves

TRIANGULAR RATIONAL PATCHES WITH BASE POINTS

Another approach to creating multisided patches is to introduce base points into rational parametric functions. Base points are parameter values for which the homogeneous coordinates (x, y, z, w) are mapped to $(0, 0, 0, 0)$ by the rational parametrization. Gregory's patch [Gre83] is defined using a special collection of rational basis functions that evaluate to $0/0$ at vertices of the parametric domain, and thus introduce base points in the resulting parametrization. Warren [War92] uses base points to create parametrizations of four-, five-, and six-sided surface patches using rational Bézier surfaces defined over triangular domains. Setting a triangle of weights to zero at one corner of the domain triangle produces a four-sided patch that is the image of the domain triangle.

S-PATCHES

Loop and DeRose [LD89, LD90] present generalizations of biquadratic and bicubic B-spline surfaces that are capable of representing surfaces of arbitrary topology by placing restrictions on the connectivity of the control mesh, relaxing C^1 continuity to G^1 (geometric) continuity, and allowing n-sided finite elements. This generalized view considers the spline surface to be a collection of possibly rational polynomial maps from independent n-sided polygonal domains, whose union possesses continuity of some number of geometric invariants, such as tangent planes. This more general view allows patches to be sewn together to describe free-form surfaces in more complex ways.

An n-sided S-patch S is constructed by embedding its n-sided domain polygon P into a simplex $\triangle$ whose dimension is one less than the number of sides of the polygon. The edges of the polygon map to edges of the simplex. A Bézier simplex $\mathbf{B}$ is then constructed using $\triangle$ as a domain. The patch representation S is obtained by restricting the Bézier simplex to the embedded domain polygon.

A-PATCHES

The A-patch technique provides simple ways to guarantee that a constructed implicit surface is single-sheeted and free of undesirable singularities. The technique uses the zero contouring surfaces of trivariate Bernstein-Bézier polynomials to construct a piecewise smooth surface. We call such iso-surfaces *A-patches*. Algorithms to fill an n-sided hole, using either a single multisided A-patch or a network of A-patches, are given in [BE95]. The blends may be C^0, C^1, or C^2 exact fits (interpolation), as well as C^1 or C^2 least squares fits (interpolation and approximation).

For degree-bounded patches, a triangular network of A-patches for the hole may be generated in two ways. First, the n-sided hole is projected onto a plane and the result of a planar triangulation is projected back onto the hole. Second, an initial multisided A-patch is created for the hole and then a coarse triangulation for the patch is generated using a rational spline approximation [BX94].

MULTIVARIATE BOX SPLINES AND SIMPLEX SPLINES

Multivariate splines are a generalization of univariate B-splines to a multivariate setting. Multivariate splines have applications in data fitting, computer-aided design, the finite element method, and image analysis. Work on splines has traditionally been for a given planar triangulation using a polynomial function basis. Box splines are multivariate generalizations of B-splines with uniform knots. Many of the basis functions used in finite element calculations on uniform triangles occur as special instances of box splines. In general a box spline is a locally supported piecewise polynomial. One can define translates of box splines that form a negative partition of unity.

In the bivariate case, box splines correspond to surfaces defined over a regular tessellation of the plane. If the tessellation is composed of triangles, it is possible to represent the surface as a collection of Bernstein-Bézier patches. The two most commonly used special tessellations arise from a rectangular grid by drawing in lines in north-easterly diagonals in each subrectangle or by drawing in both diagonals

for each subrectangle. For these special triangulations there is an elegant way to construct locally supported splines.

Multivariate splines defined as projections of simplices are called ***simplex splines***. Auerbach [AMNS91] constructs approximations with simplex splines over irregular triangles. Bivariate quadratic simplicial B-splines defined by their corresponding sets of knots derived from a (suboptimal) constrained Delaunay triangulation of the domain are employed to obtain a C^1 surface. This approach is well suited for scattered data.

Fong and Seidel [FS86, FS92] construct multivariate B-splines for quadratics and cubics by matching B-patches with simplex splines. The surface scheme is an approximation scheme based on blending functions and control points and allows the modeling of C^{k-1} continuous piecewise polynomial surfaces of degree k over arbitrary triangulations of the parameter plane. The resulting surfaces are defined as linear combinations of the blending functions, and are parametric piecewise polynomials over a triangulation of the parameter plane whose shape is determined by their control points.

45.3 FREE-FORM SURFACES

The representation of free-form surfaces is one of the major issues in geometric modeling. These surfaces are generally defined in a piecewise manner by smoothly joining several, mostly four-sided, patches. Common approaches to constructing surfaces over irregular meshes are local construction, blending polynomial pieces, and splitting.

GLOSSARY

Blending polynomial pieces: Constructing k pieces for a k-sided mesh facet such that each piece matches a part of the facet data, and a convex combination of the pieces matches the whole.

Vertex enclosure constraint: Not every mesh of polynomial curves with a well-defined tangent plane at the mesh points can be interpolated by a smooth regularly parameterized surface with one polynomial piece per facet. This constraint on the mesh is a necessary and sufficient condition to guarantee the existence of such an interpolant [Pet91]. Rational patches, singular parametrizations, and the splitting of patches are techniques to enforce the vertex enclosure constraint.

MAIN RESULTS

Blending approaches prescribe a mesh of boundary curves and their normal derivatives. For this approach, however, the existence of a well-defined tangent plane at the data points is not sufficient to guarantee the existence of a C^1 mesh interpolant, because the mixed derivatives p_{uv} and p_{vu} are given independently at

any point p. Splitting approaches, on the other hand, expect to be given at least tangent vectors at the data points, and sometimes the complete boundary. Mann et al. [MLL+92] conclude that local polynomial interpolants generally produce unsatisfactory shapes.

With splitting schemes, every triangle in the triangulation of the data points (also called a *macro-triangle*) is split into several *mini-triangles*. Split-triangle interpolants do not require derivative information of higher order than the continuity of the desired interpolant. The simplest of the split-triangle interpolants is the C^1 Clough-Tocher interpolant. Each vertex is joined to the centroid, and the macro-triangle is split into three mini-triangles. The first-order data that this interpolant requires are position and gradient value at the macro-triangle vertices, plus some cross-boundary derivative at the midpoint of each edge. There are twelve data per macro-triangle, and cubic polynomials are used over each mini-triangle. The C^1 Powell-Sabin interpolants produce C^1 piecewise quadratic interpolants to C^1 data at the vertices of a triangulated data set. Each macro-triangle is split into six or twelve mini-triangles.

PARAMETRIC PATCH SCHEMES

These patches are given in vector valued parametric form, generally mapping a rectangular or triangular parametric domain into $\mathbb{R}^3$. Parametric free-form surface patch schemes are summarized in Table 45.3.1.

TABLE 45.3.1 Free-form parametric schemes.

DEGREE	SCHEME	INPUT	PROPERTIES
Piecewise biquartic	local interpolation	cubic curve mesh	C^1, interpolate second-order data at mesh points
Piecewise biquadratic	G-edges	control points on a mesh with arbitrary topology	G^1, system of linear eqns for smoothness conditions around singular vertices
Sextic triangular pieces	approximation, no local splitting	triangular control mesh	G^1
Quadratic/cubic triangular pieces	splitting, subdivision	irregular mesh of points	C^1, refine mesh by Doo-Sabin to isolate regions of irregular points

IMPLICIT PATCH SCHEMES

While it is possible to model a general closed surface of arbitrary genus as a single implicit surface patch, the geometry of such a global surface is difficult to specify, interactively control, and polygonize. The main difficulties stem from the fact that implicit representations are iso-contours which generally have multiple real sheets, self-intersections, and several other undesirable singularities. Looking on

TABLE 45.3.2 Free-form implicit schemes.

DEGREE	SCHEME	INPUT	PROPERTIES
5, 7	local interpolation, no splitting	curve mesh from spatial triang	C^1 interpolate or approximate
2	simplicial hull construction	spatial triangulation	
3	simplicial hull construction, Clough-Tocher split	spatial triangulation	C^1
3	simplicial hull construction, Clough-Tocher split of coplanar faces	spatial triangulation	C^1 A-patches, 3 or 4 sides
5	simplicial hull construction, Clough-Tocher split of coplanar faces	spatial triangulation	C^2 A-patches

the bright side, implicit polynomial splines of the same degree have more degrees of freedom compared with parametric splines, and hence potentially are more flexible for approximating a complicated surface with fewer pieces and for achieving a higher order of smoothness. The potential of implicits remains largely latent: virtually all commercial and many research modeling systems are based on the parametric representation. An exception is SHASTRA, which allows modeling with both implicit and parametric splines [Baj93]. Implicit free-form surface schemes are summarized in Table 45.3.2.

A-SPLINES

An *A-spline* is a piecewise G^k-continuous chain of real algebraic curve segments, such that each curve segment is a smooth and single-sheeted zero contour of a bivariate Bernstein-Bézier polynomial (called a regular curve segment). A-splines are a suitable polynomial form for working with piecewise implicit polynomial curves. A characterization of A-splines defined over triangles or quadrilaterals is available [BX92, XB95].

CURVILINEAR MESH SCHEMES

Bajaj and Ihm [Baj92, BI92b] construct implicit surfaces to solve the scattered data-fitting problem. The resulting surfaces approximate or contain with C^1 continuity any collection of points and algebraic space curves with derivative information. Their Hermite interpolation algorithm solves a homogeneous linear system of equations to compute the coefficients of the polynomial defining the algebraic surface. This idea has been extended to C^k (rescaling continuity) interpolate or least squares approximate implicit or parametric curves in space [BIW93]. This problem is formulated as a constrained quadratic minimization problem, where the algebraic distance is minimized instead of the geometric distance.

In a *curvilinear-mesh-based* scheme, Bajaj and Ihm [BI92a] construct low-degree implicit polynomial spline surfaces by interpolating a mesh of curves in space using the techniques of [Baj92, BI92b, BIW93]. They consider an arbitrary spatial triangulation $\mathcal{T}$ consisting of vertices in $\mathbb{R}^3$ (or more generally, a simplicial polyhedron $\mathcal{P}$ when the triangulation is closed), with possibly normal vectors at

the vertex points. Their algorithm constructs a C^1 mesh of real implicit algebraic surface patches over $\mathcal{T}$ or $\mathcal{P}$. The scheme is local (each patch has independent free parameters) and there is no local splitting. The algorithm first converts the given triangulation or polyhedron into a curvilinear wire frame, with at most cubic parametric curves which C^1 interpolate all the vertices. The curvilinear wire frame is then fleshed to produce a single implicit surface patch of degree at most 7 for each triangular face $\mathcal{T}$ of $\mathcal{P}$. If the triangulation is convex then the degree is at most 5. Similar techniques exist for parametrics [Pet91, Far86, Sar87]; however, the geometric degrees of the solution surfaces tend to be prohibitively high.

SIMPLEX- AND BOX-BASED SCHEMES

In a ***simplex-based*** approach, one first constructs a tetrahedral mesh (called the simplicial hull) conforming to a surface triangulation $\mathcal{T}$ of a polyhedron $\mathcal{P}$. The implicit piecewise polynomial surface consists of the zero set of a Bernstein-Bézier polynomial, defined within each tetrahedron (simplex) of the simplicial hull. A simplex-based approach enforces continuity between adjacent patches by enforcing that vertex/edge/face-adjacent trivariate polynomials are continuous with one another.

Similar to the trivariate interpolation case, Powell-Sabin or Clough-Tocher splits are used to introduce degree-bounded vertices to prevent the continuity system from propagating globally. Such splitting, however, could result in a large number of patches. However, as only the zero set of the polynomial is of interest, one does not need a complete mesh covering the entire space.

Sederberg [Sed85] showed how various smooth implicit algebraic surfaces, represented in trivariate Bernstein basis form, can be manipulated as functions in Bézier control tetrahedra with finite weights. He showed that if the coefficients of the Bernstein-Bézier form of the trivariate polynomial on the lines that parallel one edge, say L, of the tetrahedron all increase (or decrease) monotonically in the same direction, then any line parallel to L will intersect the zero contour algebraic surface patch at most once.

Guo [Guo91] used cubics to create free-form geometric models and enforced monotonicity conditions on a cubic polynomial along the direction from one vertex to a point of the face opposite the vertex. A Clough-Tocher split is used to subdivide each tetrahedron of the simplicial hull. Dahmen and Thamm-Scharr [DTS93] utilize a single cubic patch per tetrahedron, except for tetrahedra on coplanar faces.

Lodha [Lod92] constructed low degree surfaces with both parametric and implicit representations and investigated their properties. A method is described for creating quadratic triangular Bézier surface patches that lie on implicit quadric surfaces. Another method is described for creating biquadratic tensor product Bézier surface patches that lie on implicit cubic surfaces. The resulting patches satisfy all the standard properties of parametric Bézier surfaces, including interpolation of the corners of the control polyhedron and the convex hull property.

Bajaj and Ihm, Guo, and Dahmen [BI92a, Guo91, Guo93, Dah89] provide heuristics based on monotonicity and least square approximation to circumvent the multiple-sheeted and singularity problems of implicit patches.

Bajaj, Chen, and Xu [BCX95] construct 3- and 4-sided A-patches that are implicit surfaces in Bernstein-Bézier (BB) form and that are smooth and single-

sheeted. They give sufficiency conditions for the BB form of a trivariate polynomial within a tetrahedron, such that the zero contour of the polynomial is a single-sheeted nonsingular surface within the tetrahedron, and its cubic-mesh complex for the polyhedron $\mathcal{P}$ is guaranteed to be both nonsingular and single-sheeted. They distinguish between convex and nonconvex facets and edges of the triangulation. A double-sided tetrahedron is built for nonconvex facets and edges, and single-sided tetrahedra are built for convex facets and edges. A generalization of Sederberg's condition is given for a three-sided j-patch where any line segment passing through the jth vertex of the tetrahedron and its opposite face intersects the patch only once. Instead of having coefficients be monotonically increasing or decreasing there is a single sign change condition. There are also free parameters for both local and global shape control.

Reconstructing surfaces and scalar fields defined over the surface from scattered data using implicit Bézier splines is described in [BBX95].

45.4 SUBDIVISION SURFACES

Subdivision techniques can be used to produce generally pleasing surfaces from arbitrary control meshes. The faces of the mesh need not be planar, nor need the vertices lie on a topologically regular mesh. Subdivision consists of splitting and averaging. Each edge or face is split, and each new vertex introduced by the splitting is positioned at a fixed affine combination of its neighbor's weights. Subdivision schemes are summarized in Table 45.4.1.

TABLE 45.4.1 Subdivision schemes.

TYPE	PROPERTIES
Doo-Sabin; Catmull-Clark	C^1, interpolate centroids of all faces at each step
Nasri	interpolate points/normals on irregular networks
Loop	C^1, split each triangle of a triangular mesh into 4 triangles
Hoppe et al.	extends Loop's method to incorporate shape edges in limit surfaces; initial vertices belong to vertex, edge, or face of limit surface
Storry and Ball	C^1 n-sided B-spline patch to fit in bicubic surface, one dof
Dyn, Levin, and Gregory	interpolatory butterfly subdivision, modify set of deterministic rules for subdivision
Bajaj, Chen, and Xu	approximation, one step subdivision to build simplicial hull, C^1 cubic and C^2 quintic A-patches
Reif	regularity conditions

MAIN ALGORITHMS

Subdivision algorithms start with a polyhedral configuration of points, edges, and faces. The control mesh will in general consist of large regular regions and isolated

singular regions. Subdivision enlarges the regular regions of the control net and shrinks the singular regions. Each application of the subdivision algorithm constructs a refined polyhedron, consisting of more points and smaller faces, tending in the limit to a smooth surface. In general the new control points are computed as linear combinations of old control points. The associated matrix is called the ***subdivision matrix***. Except for some special cases, the limiting surface does not have an explicit analytic representation. If each face of the polyhedron is a rectangle, the Doo-Sabin subdivision rules generate biquadratic tensor product B-splines, and the Catmull-Clark subdivision rules generate bicubic tensor product B-splines. Also, the subdivision technique of Loop generates three-direction box splines.

Reif [Rei92] presents a unified approach to subdivision algorithms for meshes with arbitrary topology and gives a sufficient condition for the regularity of the surface. The existence of a smooth regular parametrization for the generated surface near the point is determined from the leading eigenvalues of the subdivision matrix and an associated characteristic map.

APPROXIMATING SCHEMES

Bajaj, Chen, and Xu [BCX94] construct an "inner" simplicial hull after one step of subdivision of the input polyhedron $\mathcal{P}$. As in traditional subdivision schemes, $\mathcal{P}$ is used as a control mesh for free-form modeling, while an inner surface triangulation $\mathcal{T}$ of the hull can be considered as the second level mesh. Both a C^1 mesh with cubic A-patches and a C^2 mesh with quintic patches can be constructed to approximate the polyhedron $\mathcal{P}$.

INTERPOLATING SCHEMES

There are two key approaches to constructing interpolating subdivision surfaces. One approach is to first compute a new configuration of vertices, edges, and faces with the same topology such that the vertices of the new configuration converge to the given vertices in the limit. The subdivision technique is then applied to this new configuration The other approach is to modify the deterministic subdivision rules so that the limiting surface interpolates the vertices.

45.5 INTERACTIVE EDITING OF SURFACES

ENERGY-BASED SPLINES

A group of researchers [TF88, PB88, PTBK87, WFB87] have presented discrete models which are based extensively on the theory of elasticity and plasticity, using energy fields to define and enforce constraints. Haumann [Hau87] used the same approach but used a triangularized model and a simpler physical model based on points, springs, and hinges. Thingvold and Cohen [TC90] defined a model of elastic

and plastic B-spline surfaces which supports both animation and design operations. The basis for the physical model is a generalized point-mass/spring/hinge model that has been adapted into a simultaneous refinement of the geometric/physical model. Always having a sculptured surface representation as well as the physical hinge/spring/mesh model allows the user to intertwine physical-based operations, such as force application, with geometrical modeling. Refinement operations for spring and hinge B-spline models are compatible with the physics and mathematics of B-spline models. The models of elasticity and plasticity are written in terms of springs and hinges, and can be implemented with standard integration techniques to model realistic motions of elastic and plastic surfaces. These motions are controlled by the physical properties assigned and by kinematic constraints on various portions of the surface.

HIERARCHICAL SPLINES

Hierarchical splines are a multiresolution approach to the representation and manipulation of free-form surfaces. A hierarchical B-spline is constructed from a base surface (level 0) and a series of overlays are derived from the immediate parent in the hierarchy. Forsey and Bartels [FB88] present a refinement scheme that uses a hierarchy of rectangular B-spline overlays to produce C^2 surfaces. Overlays can be added manually to add detail to the surface, and local or global changes to the surface can be made by manipulating control points at different levels.

Forsey and Wang [FW93] create hierarchical bicubic B-spline approximations to scanned cylindrical data. The resulting hierarchical spline surface is interactively modifiable using editing capabilities of the hierarchical surface representation, allowing either local or global changes to surface shape while retaining the details of the scanned data. Oscillations occur, however, when the data have high-amplitude or high-frequency regions. Forsey and Bartels use a hierarchical wavelet-based representation for fitting tensor product parametric spline surfaces to gridded data in [FB95]. The multiresolution representation is extend to include arbitrary meshes in [EDD+95]. The method is based on approximating an arbitrary mesh by a special type of mesh and using a continuous parametrization of the arbitrary mesh over a simple domain mesh.

45.6 SOURCES AND RELATED MATERIAL

SURVEYS

All results not given an explicit reference above may be traced in these surveys.

[Alf89]: Scattered data fitting and multivariate splines.

[Baj92]: Summary of data fitting with implicit algebraic splines.

[DM83]: Scattered data fitting and multivariate splines.

[Far86, Far93]: Summary of the history of triangular Bernstein-Bézier patches.

[Hol82]: Scattered data fitting and multivariate splines.

[Sch94]: Scattered data fitting and multivariate splines.

[War94]: Subdivision techniques.

RELATED CHAPTERS

Chapter 22: Triangulations
Chapter 29: Computational real algebraic geometry
Chapter 42: Computer graphics
Chapter 47: Solid modeling

REFERENCES

[Alf89] P. Alfeld. Scattered data interpolation in three or more variables. In T. Lyche and L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design*, pages 1–34. Academic Press, New York, 1989.

[AMNS91] S. Auerbach, R.H.J. Gmelig Meyling, M. Neamtu, and H. Schaeben. Approximation and geometric modeling with simplex B-splines associated with irregular triangles. *Comput. Aided Geom. Design*, 8:67–87, 1991.

[Baj92] C. Bajaj. Surface fitting with implicit algebraic surface patches. In H. Hagen, editor, *Topics in Surface Modeling*, pages 23–52. SIAM, Philadelphia, 1992.

[Baj93] C. Bajaj. The emergence of algebraic curves and surfaces in geometric design. In R. Martin, editor, *Directions in Geometric Computing*, pages 1–29. Information Geometers, Winchester, 1993.

[BBX95] C. Bajaj, F. Bernardini, and G. Xu. Adaptive resconstruction of surfaces and surface-on-surface from dense scattered trivariate data. Tech. Rep. CS-95-028, Dept. Comput. Sci., Purdue Univ., 1995.

[BBX95] C. Bajaj, F. Bernardini, and G. Xu. Automatic reconstruction of surfaces and scalar fields from 3D scans. *Comput. Graph.* (Proc. SIGGRAPH '95), 29:109–118, 1995.

[BCX94] C. Bajaj, J. Chen, and G. Xu. Smooth low degree approximations of polyhedra. Tech. Rep. CSD-TR-94-002, Dept. Comput. Sci., Purdue Univ., 1994.

[BCX95] C. Bajaj, J. Chen, and G. Xu. Modeling with cubic A-patches. *ACM Trans. Graph.*, 14:103–133, 1995.

[BE95] C. Bajaj and S. Evans. Smooth multi-sided blends with A-patches. Presented at *4th Annu. SIAM Conf. Geom. Design*, 1995.

[BI92a] C. Bajaj and I. Ihm. C^1 smoothing of polyhedra with implicit algebraic splines. *Comput. Graph.* (Proc. SIGGRAPH '92), 26:79–88, 1992.

[BI92b] C. Bajaj and I. Ihm. Algebraic surface design with Hermite interpolation. *ACM Trans. Graph.*, 11:61–91, 1992.

[BIW93] C. Bajaj, I. Ihm, and J. Warren. Higher-order interpolation and least-squares approximation using implicit algebraic surfaces. *ACM Trans. Graph.*, 12:327–347, 1993.

[BX92] C. Bajaj and G. Xu. A-Splines: Local interpolation and approximation using C^k-continuous piecewise real algebraic curves. Tech. Rep. CAPO-92-44, Dept. Comput. Sci., Purdue Univ., 1992.

[BX94] C. Bajaj and G. Xu. Rational spline approximations of real algebraic curves and surfaces. In H.P. Dikshit and C. Michelli, editors, *Advances in Computational Mathematics*, pages 73–85. World Scientific, Singapore, 1994.

[Dah89] W. Dahmen. Smooth piecewise quadratic surfaces. In T. Lyche and L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design*, pages 181–193. Academic Press, Boston, 1989.

[DM83] W. Dahmen and C. Michelli. Recent progress in multivariate splines. In L. Shumaker, C. Chui, and J. Word, editors, *Approximation Theory IV*, pages 27–121. Academic Press, New York, 1983.

[DMS92] W. Dahmen, C. Micchelli, and H. Seidel. Blossoming begets B-spline bases built better by B-patches. *Math. Comput.*, 59:97–115, 1992.

[DTS93] W. Dahmen and T.-M. Thamm-Schaar. Cubicoids: Modeling and visualization. *Comput. Aided Geom. Design*, 10:89–108, 1993.

[EDD+95] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. *Comput. Graph.* (Proc. SIGGRAPH '95), 29:173–180, 1995.

[Far86] G. Farin. Triangular Bernstein-Bézier patches. *Comput. Aided Geom. Design*, 3:83–127, 1986.

[Far93] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide.* Academic Press, New York, 1993.

[FB88] D. Forsey and R. Bartels. Hierarchical B-spline refinement. *Comput. Graph.* (Proc. SIGGRAPH '88), 22:205–212, 1988.

[FB95] D. Forsey and R. Bartels. Surface fitting with hierarchical splines. *ACM Trans. Graph.*, 14:134–161, 1995.

[FS86] P. Fong and H.-P. Seidel. Control points for multivariate B-spline surfaces over arbitrary triangulations. *Comput. Graph. Forum*, 10:309–317, 1986.

[FS92] P. Fong and H.-P. Seidel. An implementation of multivariate B-spline surfaces over arbitrary triangulations. In *Proc. Graphics Interface*, pages 1–10, Vancouver, 1992.

[FW93] D. Forsey and L. Wang. Multi-resolution surface approximation for animation. In *Proc. Graphics Interface*, pages 192–199, Toronto, 1993.

[Gre83] J.A. Gregory. C^1 rectangular and non-rectangular surface patches. In R.E. Barnhill and W. Boehm, editors, *Computer Aided Geometric Design*, pages 25–33. North-Holland, Amsterdam, 1983.

[Guo91] B. Guo. Surface generation using implicit cubics. In N.M. Patrikalakis, editor, *Scientific Visualizaton of Physical Phenomena*, pages 485–530. Springer-Verlag, Tokyo, 1991.

[Guo93] B. Guo. Non-splitting macro patches for implicit cubic spline surfaces. *Comput. Graph. Forum*, 12:434–445, 1993.

[Hau87] D. Haumann. Modeling the physical behavior of flexible objects. ACM SIGGRAPH Course Notes #17, 1987.

[Hol82] K. Hollig. Multivariate splines. *SIAM J. Numer. Anal.*, 19:1013–1031, 1982.

[LD89] C. Loop and T. DeRose. A multisided generalization of Bézier surfaces. *ACM Trans. Graph.*, 8:205–234, 1989.

[LD90] C. Loop and T. DeRose. Generalized B-spline surfaces of arbitrary topology. *Comput. Graph.* (Proc. SIGGRAPH '90), 24:347–356, 1990.

[Lod92] S. Lodha. Surface approximation by low degree patches with multiple representations. PhD thesis, Purdue Univ., 1992.

[MLL+92] S. Mann, C. Loop, M. Lounsbery, D. Meyers, J. Painter, T. DeRose, and K. Sloan. A survey of parametric scattered data fitting using triangular interpolants. In H. Hagen, editor, *Curve and Surface Modeling*, pages 145–172. SIAM, Philadelphia, 1992.

[MW91] D. Moore and J. Warren. Approximation of dense scattered data using algebraic surfaces. In *Proc. 24th Hawaii Internat. Conf. Syst. Sci.*, pages 681–690, 1991.

[PB88] J. Platt and A. Barr. Constraint methods for flexible models. *Comput. Graph.* (Proc. SIGGRAPH '88), 22:279–288, 1988.

[Pet91] J. Peters. Smooth interpolation of a mesh of curves. *Constr. Approx.*, 7:221–246, 1991.

[PK89] N.M. Patrikalakis and G.A. Kriezis. Representation of piecewise continuous algebraic surfaces in terms of B-splines. *Visual Comput.*, 5:360–374, 1989.

[PTBK87] J. Platt, D. Terzopoulos, A. Barr, and K. Fleischer. Elastically deformable models. *Comput. Graph.* (Proc. SIGGRAPH '87), 21:205–214, 1987.

[Rei92] U. Reif. A unified approach to subdivision algorithms. Tech. Rep. 92-16, Math. Inst. A, Univ. Stuttgart, 1992.

[Sar87] R. Sarraga. G^1 interpolation of generally unrestricted cubic Bézier curves. *Comput. Aided Geom. Design*, 4:23–39, 1987.

[Sch94] L. Schumaker. Applications of multivariate splines. In *Mathematics of Computation, 1943–1993: A Half-Century of Computational Mathematics*, pages 177–201. Amer. Math. Soc., Providence, 1994.

[Sed85] T.W. Sederberg. Piecewise algebraic patches. *Comput. Aided Geom. Design*, 2:53–59, 1985.

[Sei89] H. Seidel. A new multiaffine approach to B-splines. *Comput. Aided Geom. Design*, 6:23–32, 1989.

[TC90] J. Thingvold and E. Cohen. Physical modeling with B-spline surfaces for interactive design and animation. *Comput. Graph.* (Proc. SIGGRAPH '90), 24:129–137, 1990.

[TF88] D. Terzopolos and K. Fleischer. Modeling inelastic deformation: Visoelasticity, plasticity, fracture. *Comput. Graph.* (Proc. SIGGRAPH '88), 22:269–278, 1988.

[War92] J. Warren. Creating multisided rational Bezier surfaces using base points. *ACM Trans. Graph.*, 11:127–139, 1992.

[War94] J. Warren. Subdivision methods for geometric design. Manuscript, 1994.

[WFB87] A. Witkin, K. Fleischer, and A. Barr. Energy constraints on parameterized models. *Comput. Graph.* (Proc. SIGGRAPH '87), 21:225–232, 1987.

[XB95] G. Xu and C. Bajaj. Regular algebraic curve segments (I)—Definitions and characteristics. Tech. Rep. CAPO-95-061, Dept. Comput. Sci., Purdue Univ., 1995.

46 DESIGN AND MANUFACTURING

Ravi Janardan and Tony C. Woo

INTRODUCTION

This chapter surveys some recent work on the application of techniques from computational geometry to problems in computer-aided design and manufacturing. Three representative topics are addressed: mold design, numerically controlled machining, and inspection. Within each topic, we discuss problems that have benefited from the application of geometric techniques, and mention several other problems where such techniques could be used to advantage.

46.1 MOLD DESIGN

Casting and injection molding processes are used extensively to mass-manufacture a wide variety of products. A key step here is the design of the mold from a CAD model of the part, since this affects both the speed of the process and the quality of the finished part. For instance, how the model is decomposed into pieces to make the mold halves determines the number of undercuts in the mold: the greater the number of undercuts, the slower the de-molding process. As another example, the location of venting holes on the mold and the choice of pouring direction determine the extent of air pockets created during mold filling; this ultimately affects the strength and finish of the product.

GLOSSARY

Mold: A cavity in the shape of the part to be manufactured into which molten metal is poured. It consists of two mating parts called ***mold halves***. Once the metal has hardened, the mold halves are pulled apart in opposite directions (i.e., ***de-molded***) and the part is removed.

Undercut: Any point p on a part's surface such that the outward normal at p makes an angle greater than $90°$ with the de-molding direction for the mold half containing p. Generally, a group of such points forms a recess or projection in the part that prevents easy de-molding.

Parting line: A continuous closed curve on the surface of the part that defines the two halves; thus it also defines the profile of the contact surface between the two mold halves (Figure 46.1.1).

CH(P): The convex hull of a polyhedron P.

Dent: For a polyhedron P, a connected component of $\mathrm{CH}(P) - P$.

Fillability: The ability to fill a mold from a given pouring direction without cre-

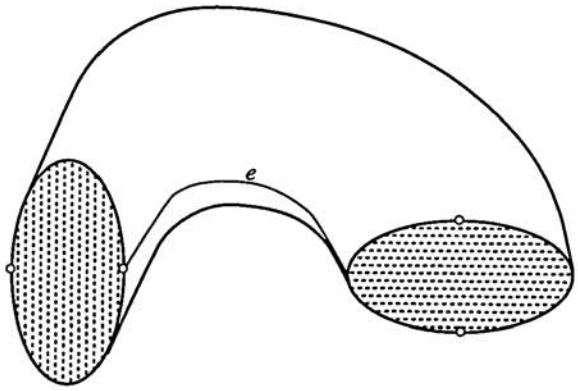


FIGURE 46.1.1
A parting line (e) for the exhaust manifold of an automobile.

ating air pockets. This is a function of the mold geometry, the pouring direction, and the location of air-venting holes.

Part decomposition: The process of dividing a part into smaller pieces and making mold halves for these that satisfy certain optimization criteria.

RESULTS

Recent geometric work on mold design falls into two broad categories: fillability and part decomposition.

Fillability addresses problems such as deciding if a mold can be filled from a given pouring direction without creating air pockets, and finding a pouring direction that eliminates air pockets using the smallest number of venting holes. In [BT95], several results are presented for the two-dimensional version of the fillability problem, including: (a) deciding in $O(n)$ time whether an n-vertex simple polygon P can be filled from a given pouring gate in a given pouring direction without creating air pockets; (b) enumerating in $O(n \log n)$ time all pouring directions that permit such a fill; (c) computing in $O(n \log n)$ time a pouring direction that minimizes the number of air pockets; and (d) characterizing various classes of polygons with respect to their fillability. The corresponding three-dimensional problems are investigated in [BvKT93]. For an n-vertex polyhedron P the decision problem is solved in $O(n)$ time and the enumeration and optimization problems in $O(n^2)$ time. For the optimization problem, a reduction from the so-called 3SUM-HARD problem [GO95] is used to establish a type of lower bound.

Part decomposition refers to the problem of "cutting" the CAD model into smaller pieces and making mold halves for these that meet certain optimization criteria. For instance, how can a 3D part P be divided into two such that the parting line is as "flat" as possible? As noted in the mold-design literature, the flatter the parting line, the more cost-effective and accurate the mold. While the notion of flatness has not been quantified in the literature, it is generally taken to mean that the parting line should lie as nearly in a plane as possible. Although a parting line that lies completely in a plane can always be produced by intersecting P with a plane, this can create undercuts, even if P is a convex polyhedron (Figure 46.1.2).

The problem of computing a flattest undercut-free parting line for an n-vertex convex polyhedron P is considered in [MGJ96]. That such a line always exists is

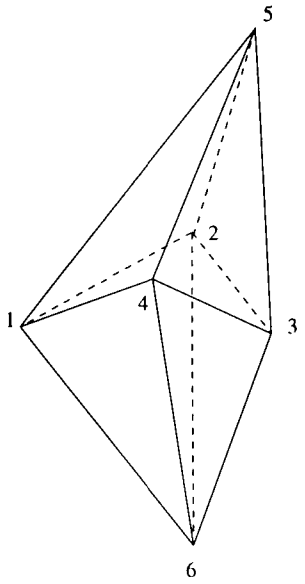


FIGURE 46.1.2
An octahedron that cannot be divided by a plane into two halves without creating undercuts. For example, the plane containing vertices 1, 2, *and* 3 *creates a projection under the chain* 1–4–3. *Undercuts can be avoided by choosing the parting line to be* 1–2–3–4–1 *(or* 2–5–4–6–2*), but this is no longer in a plane. (From* [MGJ96], *with permission.)*

clear—simply take the boundary, $L(\vec{d})$, of P, as viewed along lines of sight parallel to any direction $\vec{d}$. The ***flatness***, $\rho(\vec{d})$, of $L(\vec{d})$ is defined in [MGJ96] as the sum of the squares of the projected lengths of the segments of $L(\vec{d})$, where the projection is onto a plane normal to $\vec{d}$, divided by the sum of the squares of the lengths of the segments of $L(\vec{d})$. Thus, $\rho(\vec{d}) \leq 1$, with equality holding if and only if $L(\vec{d})$ lies in a plane. An $O(n^2)$-time algorithm is given to compute a direction $\vec{d}$ that maximizes $\rho(\vec{d})$. The algorithm blends together geometric techniques such as visibility cones, arrangements, and shortest paths in a simple polygon, with methods from continuous optimization.

In [BBvK94] the problem of deciding if a given n-vertex polyhedron can be parted by a single plane without creating undercuts is addressed. For a nonconvex (resp. convex) polyhedron, the running time is $O(n^{3/2+\epsilon})$ (resp. $O(n \log^2 n)$), where $\epsilon > 0$ is an arbitrarily small constant. A related result is presented in [CCW93a], where it is shown how an undercut-minimizing parting line for an n-vertex polyhedron with p dents can be found in time $O(np \log p)$; however, the parting line can be quite stepped.

Other work on part decomposition includes decomposition of two-dimensional molds, identification of criteria other than parting line shape and number of undercuts, and heuristics for computing a de-molding direction without too many undercuts.

OPEN PROBLEMS

1. It is unlikely that the $O(n^2)$-time algorithm in [BvKT93] for minimizing the number of air-venting holes can be improved (in view of the 3SUM-HARD-based lower bound). However, can a significantly faster algorithm be devised that approximates the minimum number of air-venting holes to within a constant factor?

2. The goals of maximizing the flatness of the parting line and minimizing the number of undercuts are usually at odds. Often, however, meeting specified thresholds suffices: for instance, given parameters u_0 and ρ_0, design an efficient algorithm to find a parting line with at most u_0 undercuts and flatness at least ρ_0.

3. A polyhedron P is ***1-castable*** if it can be parted by a plane without creating undercuts. The results in [BBvK94] allow one to decide 1-castability efficiently. However, there exist polyhedra that are not 1-castable (Figure 46.1.3). To extend the class of polyhedra that can be cast with planes, call a polyhedron P ***2-castable*** if there is a plane h such that the polyhedra $P \cap h^+$ and $P \cap h^-$ are both 1-castable. (Here h^+ and h^- denote the two halfspaces of h.) Give efficient algorithms to decide 2-castability and characterize the class of 2-castable polyhedra.

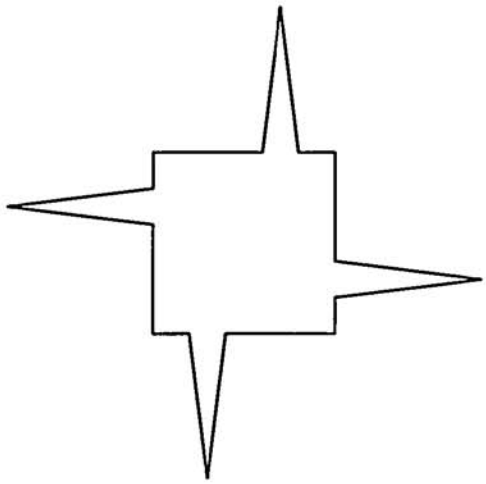


FIGURE 46.1.3
Cross-sectional view of a polyhedron that is not 1-*castable. The cross section tapers along the length of the polyhedron to a point and then expands again, so that the polyhedron consists of a "double pyramid." Any casting plane will create an undercut at one (or more) of the spikes or at some of the slanted facets corresponding to the horizontal and vertical segments in the cross section.*

46.2 NUMERICALLY CONTROLLED MACHINING

The dominant machining process today is *numerically controlled (NC) machining*, where parts are manufactured under computer control based on information extracted from a CAD model. Examples of NC-machines include milling machines and lathes. Typical questions of interest concern accessibility of the tool to the part and generation of toolpaths that satisfy certain optimization criteria.

GLOSSARY

Degrees of freedom (dof): The types of motion permitted of an NC-machine. Specified as a combination of translation and (full or partial) rotation with respect to the coordinate axes.

Visibility map (or VMap): The set of points on the unit sphere representing the directions along which a tool can approach (or "see"; cf. Chapter 25) *all* points on the surface in question without being blocked by other portions of the part. The VMap is a function of the surface geometry and the geometry of the cutting tool, and is in practice usually representable as a (spherical) polygon formed by the intersection of a certain set of hemispheres [GWT94]. For instance, the VMap of a plane is the hemisphere whose pole is the normal to the plane, the VMap of a half-cylinder is a half-great circle, the VMap of a hemisphere is a point, and the VMap of a dent in a polyhedron is the intersection of the set of hemispheres determined by the normals to the dent's faces.

Pocket: A region bounded by one or more closed curves, which delineates the area on the part from which material must be removed.

Spherical band of width b: The set of all points on the unit sphere that are at a distance of at most b on either side of a great circle, where the distance is measured along a great circle arc.

Part setup: The process of dismounting a part, and re-calibrating and re-mounting it in a new orientation on the worktable of an NC-machine.

Direction-parallel pocket machining: A machining discipline where the tool is constrained to stay within a pocket and, moreover, always moves from left to right with respect to a chosen reference line.

Zigzag pocket machining: Similar to direction-parallel machining, except that the tool moves from left to right, then right to left, and so on.

Contour-parallel pocket machining: The tool is constrained to move along a sequence of closed paths that are parallel to the pocket's contour.

RESULTS

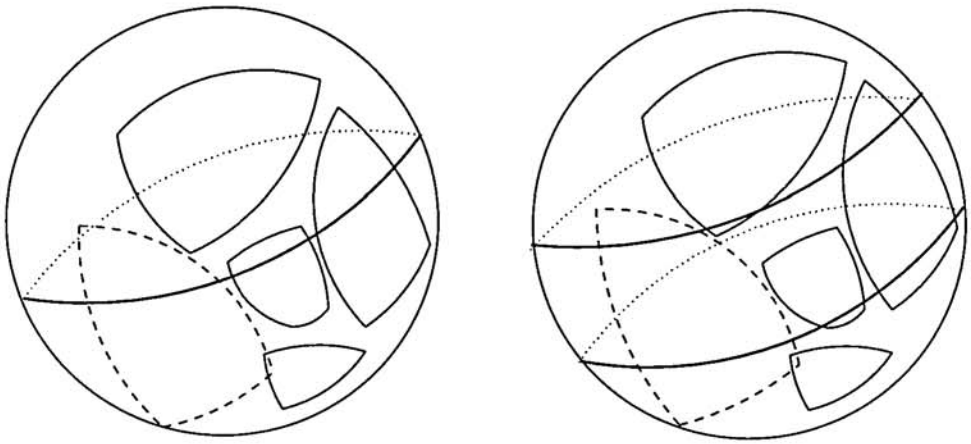
Two important parameters of an NC-machine are the dof of the machine and the type of cutting tool. The dof include translation along the principal coordinate directions (**3-*axis machine***), plus rotation of the worktable about one axis (**4-*axis machine***), plus partial swivel of the tool about a second axis (**5-*axis machine***). The dof determines global motion of the tool. For example, in a 4-axis machine, the directions in which the tool can move can be represented on the unit sphere as a great circle whose normal is the rotational axis of the worktable. In a 5-axis machine, if the tool can swivel by $\pm b/2$ radians, then the tool motion directions are given by a spherical band of width b, where the great circle associated with the band is as in the 4-axis case. Cutters are classified, according to the maximum angle θ that they can tilt from the local surface normal, as: ***flat-end*** ($\theta = 0$ radians), ***fillet-end*** ($\theta < \pi/2$ radians), and ***ball-end*** ($\theta = \pi/2$ radians). Thus the cutter geometry determines local motion of the tool: a flat-end cutter can approach a point p on a surface only along the surface normal at p, while a ball-end cutter can approach p along any direction lying within the hemisphere with pole p.

Part orientation. In order to machine a surface on a part, the tool must be able to approach (or "see") every point on the surface without being blocked by other portions of the part. For a given orientation of the part on the machine's worktable, only a subset of the surfaces that need to be machined might be so visible

to the tool. Therefore, after each such set of visible surfaces has been machined, a part setup is performed to bring a new set of surfaces into view. However, part setup can be quite time-consuming in relation to the actual machining time (hours versus minutes, sometimes). This motivates the following problem. Given the part geometry and the machine parameters, compute a sequence of part orientations that minimizes the number of setups. Unfortunately, this problem is NP-hard, and so attention has focused on obtaining efficient algorithms that approximate closely the minimum number of setups.

A natural approach is the greedy heuristic, which consists of finding repeatedly a part orientation that allows access to the maximum number of as-yet-unmachined surfaces [CCW93b, GJMW96]. Suppose, for example, that a 4-axis machine equipped with a ball-end cutter is used. Assume further that the VMaps for the part's surfaces are available; for a ball-end cutter, the VMaps are intersections of certain hemispheres and can be computed as described in [GWT94]. Recall that each VMap represents the directions along which every point on the corresponding surface can be seen by the tool. Therefore, to find an orientation in which the maximum number of surfaces can be seen is equivalent to finding a great circle, C, that intersects the maximum number of VMaps. (Here, C represents the directions in which the tool can move in a 4-axis machine.) Similarly, for a 5-axis (resp. 3-axis) machine, the problem is to find a spherical band B of width b (resp. a point P) that intersects the maximum number of VMaps (Figure 46.2.1).

FIGURE 46.2.1
A great circle for a 4*-axis machine* (a) *and a spherical band for a* 5*-axis machine* (b) *intersecting a set of VMaps. (From* [GJMW96]*, with permission.)*



Given m VMaps with a total of n vertices, this problem is solved in [CCW93b] in $O(nm \log m)$ time and $O(nm)$ space for a 3- and 4-axis machine equipped with a ball-end cutter. In [GJMW96], the time bound is improved to $O(n^2)$ in the worst case—when $m = \Theta(n)$—and, moreover, an $O(nm \log m)$-time and $O(nm)$-space al-

gorithm is given for 5-axis machines. These results are based on geometric duality, topological sweep (Section 21.4), and properties concerning intersections and covering of polygons on the unit sphere. In [GJMW96], an $O(n^2 + nm \log m)$-time and $O(nm)$-space algorithm is also given for fillet-end tools on 4- and 5-axis machines. All of these results imply an $O(\log m)$-approximation to the minimum number of setups, via the well-known approximation result for the set-cover problem.

Tool paths. A related problem is that of generating tool paths that meet certain optimization criteria, given the pocket geometry, the tool size and geometry, and a machining discipline such as direction-parallel machining, zigzag machining, or contour-parallel machining. The optimization criteria include minimizing the total length traveled by the tool, minimizing the number of ***tool retractions*** (i.e., the number of times the tool is lifted off the workpiece), and minimizing the number of times any point is machined by the tool. In [AHS96], a zigzag pocket machining algorithm is given and it is proved that the number of retractions is at most $5r + 6h$ for a pocket with $h \geq 0$ holes, where r is the minimum number of retractions. Moreover, no point is machined more than once. (Experiments in [AHS96] indicate a better approximation factor of 1.5.) The approach is based on constructing and processing a so-called machining graph. The algorithm runs in $O(n)$ time, where n is the number of vertices in the machining graph. (Here n is a function of the pocket geometry and the tool size.)

In [AFM93], the following related optimization problem is shown to be NP-hard: Given a polygonal pocket of size n and a tool represented by a unit disk or a square, find a closed path of minimum length that visits every point of the pocket at least once. It is shown, however, that one can compute a path that is at most three times longer than a shortest path in time $O(n \log n)$.

Heuristics have also been investigated for other tool-path generation problems—see, for instance, the references cited in [AHS96]. However, no approximation bounds have been proved.

OPEN PROBLEMS

1. The type of visibility considered in the part setup problem is between two points (one being the tool and the other being a point on the part's surface) along a straight line. Characterizations of such VMaps and efficient algorithms are given in [GWT94]. Give characterizations and efficient algorithms for VMaps under point-point visibility along circular trajectories (e.g., as produced by the rotary joints of a robot arm) or along parabolic trajectories (e.g., as executed by droplets under gravity in vapor deposition processes). Also of interest are segment-segment and plane-plane visibility along straight line trajectories.

2. Consider an augmented 4-axis (resp. 5-axis) machine, where the worktable can rotate fully (resp. tilt by $\pi/2$ radians) about a second axis. In the greedy framework described earlier, this reduces to finding a pair of orthogonal great circles (resp. spherical bands) that intersect the maximum number of VMaps. No algorithms are known for this problem.

3. Prove that the zigzag pocket machining problem that calls for the minimum

number of retractions and requires that no pocket point is machined more than once ([AHS96]) is NP-hard, or provide a polynomial-time algorithm.

4. Investigate tool-path generation problems for contour-parallel machining and provide provably good approximation algorithms.

46.3 INSPECTION

Inspection is the process of verifying that the shape, size, and position of features on a manufactured component conform to those of an ideal model, up to a specified tolerance. Modern inspection methods use *coordinate measuring machines* to sample data points from the component's surface, and require efficient algorithms for checking that the data meet the specified tolerances. Computational geometry research in this area has focused on efficient algorithms for checking various shape tolerances, including straightness, flatness, roundness, and cylindricity. Each of these specifies a zone within which the sampled points must lie.

GLOSSARY

Straightness: The zone is the region lying between two parallel lines whose separation d equals the specified tolerance (Figure 46.3.1(a)).

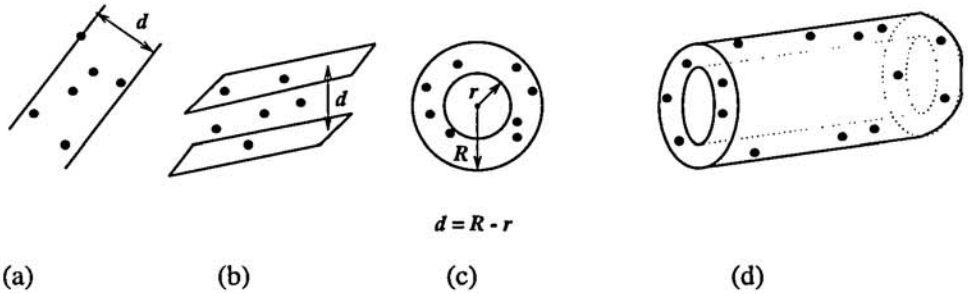
Flatness: The zone is the region lying between two parallel planes whose separation d equals the specified tolerance (Figure 46.3.1(b)).

Roundness: The zone is the region lying between two concentric circles or spheres (in 2D and 3D, respectively), whose radial difference d equals the specified tolerance (Figure 46.3.1(c)).

Cylindricity: The zone is the region lying between two coaxial cylinders whose radial difference d equals the specified tolerance (Figure 46.3.1(d)).

Minimum zone: A zone of minimum size (separation or radial difference in the above examples) that contains a set of sampled data points.

FIGURE 46.3.1
Tolerance zones for (a) *straightness,* (b) *flatness,* (c) *roundness, and* (d) *cylindricity.*



RESULTS

Algorithms have been designed for the minimum zone version of the above-mentioned tolerances. Given the minimum zone, one can infer readily if the specified tolerance is met—it is met iff it is no smaller than the size of the minimum zone. For example, for straightness tolerance, the sampled points lie in a strip of width d iff $d \geq d^\star$, where $d^\star$ is the width of the narrowest strip enclosing the points.

The minimum zone problem for straightness is well-known in computational geometry as the ***width*** problem in two dimensions: Given a set S of n points in the plane, find the narrowest strip enclosing the points. It can be computed in $O(n \log n)$ time [HT88] by using the observation that one of the bounding lines of the strip must contain an edge of the convex hull, $\mathrm{CH}(S)$, of S and the other must contain a vertex of $\mathrm{CH}(S)$. Once $\mathrm{CH}(S)$ is computed, in $O(n \log n)$ time, all candidate pairs of lines can be enumerated in additional $O(n)$ time, using the so-called rotating calipers method.

The minimum zone problem for flatness leads to the width problem in three dimensions, where we seek the narrowest slab that is bounded by two parallel planes and which encloses the point set. This problem can be solved deterministically in time $O(n^{8/5+\epsilon})$ [CEGS93], using parametric search (Chapter 37), and in time $O(n^{3/2+\epsilon})$ using randomization [AS95] (Chapter 34).

The minimum zone problem for roundness is also called the ***thinnest annulus*** problem. In two dimensions, the center of the thinnest annulus is a vertex of the nearest- or farthest-neighbor Voronoi diagram or the point of intersection of an edge from each diagram. Based on this, the thinnest annulus can be computed in $O(n \log n + I)$ time [RZ92], where I is the number of edge intersections; in the worst case, $I = \Theta(n^2)$. This has been improved recently to $O(n^{3/2+\epsilon})$ using randomization [AS95]. (If the sampled points are in convex position, then the problem is solvable in time $O(n)$ [Swa93].)

Other results related to roundness in two dimensions include the following:

1. The ***relative roundness*** of a component C is the minimum value of $(R-r)/r$, where R and r are the outer and inner radii of an annulus enclosing C. In [SY95], an algorithm is given for obtaining appropriate sample points based on the finger probe model (Chapter 26) and a linear-time algorithm is given for estimating the relative roundness of C whenever it is convex and has bounded relative roundness.

2. An annulus of minimum area that encloses a set of n given points can be computed in $O(n)$ time using linear programming.

No results are known for the minimum zone problem for roundness in three dimensions or for cylindricity. However, it has been shown recently that a minimum-radius cylinder enclosing n given points can be computed in time $O(n^4 \mathrm{polylog}(n))$ in the algebraic model by linearizing the problem in a higher dimension and applying parametric search [SSTY96].

In closing this section, we remark that other approaches to shape tolerance have also been proposed in the manufacturing literature. These methods are based on optimization techniques and employ other criteria, such as least-squares fit. They are generally heuristic and provide only approximations to the optimal solution.

OPEN PROBLEMS

1. The flatness and roundness algorithms given in [AS95, CEGS93] are very complex and are unlikely to be easily implementable. Can one devise simpler algorithms by taking advantage of properties of the sampled set? As noted in [Yap95], it is reasonable to assume that the profile being tested is almost circular (otherwise, it can usually be rejected by a visual inspection). Thus, the sampling of the surface should yield a cloud of points of nearly circular shape, which might be exploited in designing a simpler and faster algorithm.
2. Devise efficient algorithms for roundness in 3D and for cylindricity, again by exploiting properties of the point set.
3. Investigate tolerance algorithms for an assembly of components, e.g., an assembly of gears. The assembly has a specified global tolerance and each component has an individual tolerance. The goal is to verify if the assembly is within tolerance, possibly as the components move relative to each other.

46.4 OTHER TOPICS

Besides the three representative topics that we have addressed, there are other areas for fruitful interaction between computational geometry and manufacturing. These include: rapid prototyping and layered manufacturing, design of mechanisms and linkages (Section 41.1); geometric constraint systems (Section 47.3); interpretation and reconstruction of engineering drawings, assembly and disassembly of components (Section 41.3); geometric software for manufacturing applications, process planning and simulation, mesh generation (Section 22.4); VLSI design and layout, and vision, robotics (Chapter 41); and geometric modeling issues relevant to manufacturing (Chapter 45).

46.5 SOURCES AND RELATED MATERIAL

FURTHER READING

The following contain additional discussion and references related to the topics in this chapter.

[Bos95]: Provides a good exposition of the application of computational geometry techniques to problems in molding, casting, and StereoLithography, a layered manufacturing process.

[Woo94]: Discusses various kinds of visibility in the context of different manufacturing processes.

[Hel91]: Contains a detailed discussion of the application of geometric techniques to problems in pocket machining.

[Yap94]: Discusses the use of computational geometry in tolerancing. Also addresses the role of exact computation in designing reference software for metrology.

[TJWC89]: Contains references to other approaches used for minimum zone problems, including alternative search strategies, optimization, least-squares fit, and statistical analysis. (Also gives algorithms similar to [HT88] for straightness and flatness.)

[Bra86]: A good general reference on a variety of design and manufacturing processes, including casting, molding, forging, stamping, machining, etc.

RELATED CHAPTERS

Chapter 21: Arrangements
Chapter 25: Visibility
Chapter 26: Geometric reconstruction problems
Chapter 41: Robotics
Chapter 45: Splines and geometric modeling
Chapter 47: Solid modeling

REFERENCES

[AFM93] E. Arkin, S. Fekete, and J. Mitchell. The lawnmower problem. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 461–466, Waterloo, 1993.

[AHS96] E. Arkin, M. Held, and C. Smith. Optimization problems related to zigzag pocket machining. In *Proc. 7th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 419–428, 1996.

[AS95] P.K. Agarwal and M. Sharir. Efficient randomized algorithms for some geometric optimization problems. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 326–335, 1995.

[BBvK94] P. Bose, D. Bremner, and M. van Kreveld. Determining the castability of simple polyhedra. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 123–131, 1994.

[Bos95] P. Bose. *Geometric and Computational Aspects of Manufacturing Processes.* Ph.D. thesis, School of Computer Science, McGill Univ., Montréal, 1995.

[Bra86] J.G. Bralla. *Handbook of Product Design for Manufacturing.* McGraw-Hill, New York, 1986.

[BT95] P. Bose and G. Toussaint. Geometric and computational aspects of gravity casting. *Comput. Aided Design*, 27:455–464, 1995.

[BvKT93] P. Bose, M. van Kreveld, and G. Toussaint. Filling polyhedral molds. In *Proc. 3rd Workshop Algorithms Data Struct.*, volume 709 of *Lecture Notes in Comput. Sci.*, pages 210–221. Springer-Verlag, New York, 1993.

[CCW93a] L.-L. Chen, S.-Y. Chou, and T.C. Woo. Parting directions for mould and die design. *Comput. Aided Design*, 25:762–768, 1993.

[CCW93b] L.-L. Chen, S.-Y. Chou, and T.C. Woo. Separating and intersecting spherical polygons: Computing machinability on three-, four-, and five-axis numerically controlled machines. *ACM Trans. Graph.*, 12:305–326, 1993.

[CEGS93] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. Diameter, width, closest line pair and parametric searching. *Discrete Comput. Geom.*, 10:183–196, 1993.

[GJMW96] P. Gupta, R. Janardan, J. Majhi, and T. Woo. Efficient geometric algorithms for workpiece orientation in 4- and 5-axis NC-machining. *Comput. Aided Design*, 28:577–587, 1996.

[GO95] A. Gajentaan and M.H. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Comput. Geom. Theory Appl.*, 5:165–185, 1995.

[GWT94] J. Gan, T. Woo, and K. Tang. Spherical maps: Their construction, properties, and approximation. *J. Mech. Design*, 116:357–363, 1994.

[Hel91] M. Held. *On the Computational Geometry of Pocket Machining.* Volume 500 of *Lecture Notes in Comput. Sci.*, Springer-Verlag, Berlin, 1991.

[HT88] M.E. Houle and G.T. Toussaint. Computing the width of a set. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10:761–765, 1988.

[MGJ96] J. Majhi, P. Gupta, and R. Janardan. Computing a flattest, undercut-free parting line for a convex polyhedron, with application to mold design. In *Proc. 1st ACM Workshop Appl. Comput. Geom.*, volume 1148 of *Lecture Notes in Comput. Sci.*, pages 39–47. Springer-Verlag, Berlin, 1996.

[RZ92] U. Roy and X. Zhang. Establishment of a pair of concentric circles with the minimum radial separation for assessing roundness error. *Comput. Aided Design*, 24:161–168, 1992.

[SSTY96] E. Schömer, J. Sellen, M. Teichmann, and C. Yap. Efficient algorithms for the smallest enclosing cylinder problem. In *Proc. 8th Canad. Conf. Comput. Geom.*, pages 264–269, Ottawa, 1996.

[Swa93] K. Swanson. An optimal algorithm for roundness determination on convex polygons. In *Proc. 3rd Workshop Algorithms Data Struct.*, volume 709 of *Lecture Notes in Comput. Sci.*, pages 601–609. Springer-Verlag, New York, 1993.

[SY95] T. Shermer and C. Yap. Probing for near-centers and estimating relative roundness. Talk presented at *ASME Workshop Tolerancing Metrology*, Charlotte, 1995. `ftp://cs.nyu.edu/pub/local/yap/metrology/roundness.ps.gz`.

[TJWC89] M. Traband, S. Joshi, R. Wysk, and T. Cavalier. Evaluation of straightness and flatness tolerances using minimum zone. *Manufac. Rev.*, 2:189–195, 1989.

[Woo94] T. Woo. Visibility maps and spherical algorithms. *Comput. Aided Design*, 26:6–16, 1994.

[Yap94] C. Yap. Exact computational geometry and tolerancing metrology. In D. Avis and P. Bose, editors, *Snapshots of Computational and Discrete Geometry*, volume 3. Tech. Report SOCS-94.50, McGill Univ., 1994.

[Yap95] C. Yap. Aspects of dimensional tolerancing. Talk at *5th MSI-Stony Brook Workshop Comput. Geom.*, 1995.

47 SOLID MODELING

Christoph M. Hoffmann

INTRODUCTION

The objective of solid modeling is to represent, manipulate, and reason about the three-dimensional shape of solid physical objects, by computer.

Solid modeling is an application-oriented field that has a tradition of implementing systems and algorithms. Major application areas include design, manufacturing, computer vision, graphics, and virtual reality. Technically, the field draws on diverse sources including numerical analysis, symbolic algebraic computation, approximation theory, applied mathematics, point set topology, algebraic geometry, computational geometry, and data bases.

First, the major representations of solids are reviewed in Section 47.1. Then, major layers of abstraction in a typical solid modeling system are characterized in Section 47.2. The lowest level of abstraction comprises a substratum of basic service algorithms. At an intermediate level of abstraction there are algorithms for larger, more conceptual operations. Finally, a yet higher level of abstraction presents to the user a functional view that is typically targeted toward solid design.

Solid modeling is in transition. Classical design paradigms that concentrated on obtaining one specific final shape are being supplanted by feature-based, constraint-based design paradigms that are oriented more toward the design process and define classes of shape instances. These new paradigms venture into territory that has yet to be plumbed mathematically and computationally. Concurrent with this paradigm shift, there is also a shift in the system architecture toward modularized confederations of plug-compatible functional components. We explore these trends lightly in Section 47.3.

Open problems are gathered in Section 47.4.

47.1 MAJOR REPRESENTATION SCHEMATA

GLOSSARY

Solid representation: Any representation allowing a deterministic, algorithmic point membership test.

Constructive solid geometry (CSG): The solid is represented as union, intersection, and difference of primitive solid solids.

Boundary representation (Brep): The solid surface is represented as a quilt of vertices, edges, and faces.

Spatial subdivision: The solid is decomposed into a set of nonintersecting primitive volumes.

Medial surface transformation: Closure of the locus of centers of maximal inscribed spheres, and a function giving the minimum distance to the solid boundary. Usually called the ***MAT*** for "medial axis transformation."

A solid representation must allow the unambiguous, algorithmic determination of point membership: given any point $p = (x, y, z)$, there must be an algorithm that determines whether the point is inside, outside, or on the surface of the solid. Moreover, restrictions are placed on the topology of the solid and its embedding, excluding, for example, fractal solids.

These restrictions are eminently reasonable. Increasingly, however, solid modeling systems depart from this strict notion of solid and permit representing a mixture of solids, surfaces, curves, and points. The additional geometric structures are useful for certain design processes, for interfacing with applications such as meshing solid volumes, and for abstracting solid features, to name a few.

47.1.1 CONSTRUCTIVE SOLID GEOMETRY

GLOSSARY

Primitive solids: Traditionally block, sphere, cylinder, cone, torus. More general primitives are possible.

Sweep: Volume covered by sweeping a solid or a closed contour in space.

Extrusion: Sweep along a straight line segment.

Revolution: Circular sweep.

Regularized Boolean operation: The closure of the interior of a set-theoretic union, intersection, or difference.

Algebraic halfspace: Points such that $f(x, y, z) \leq 0$ where f is an irreducible polynomial.

Irreducible polynomial: Polynomial that cannot be factored over the complex numbers.

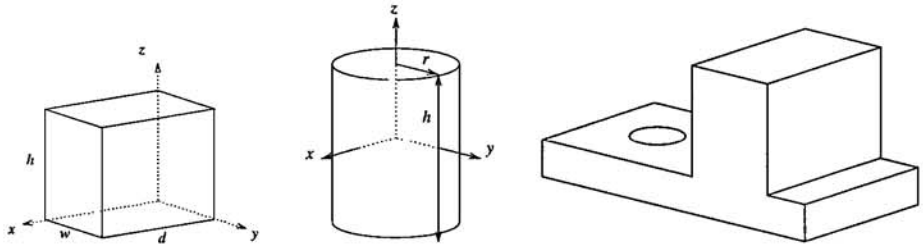
Classical Constructive Solid Geometry (CSG) represents a solid as a set-theoretic Boolean expression of *primitive* solid objects, of a simpler structure. Both the surface and the interior of the final solid are thereby defined, albeit implicitly. The CSG representation is valid if the primitives are valid. A solid's surface is closed and orientable and encloses a volume. The traditional CSG primitives are block, sphere, cylinder, cone, and torus.

A solid is represented as an algebraic expression that uses rigid motions and regularized set operations. The traditional operations are regularized union, intersection, and difference. A regularized set operation is obtained by taking the closure of the interior of the set-theoretic result. The effect is to obtain solids that do not contain lower-dimensional parts, such as interior (or dangling exterior) faces, edges, and vertices.

Each solid has a default coordinate system that can be changed with a rigid body transformation. A Boolean operation identifies the two coordinate systems

of the solids to be combined and makes it the default coordinate system of the resulting solid.

FIGURE 47.1.1
Left and middle: CSG primitives $\texttt{block}(w, d, h)$ *and* $\texttt{cylinder}(r, h)$ *with default coordinate systems. Right: T-bracket as union of two blocks minus a cylinder.*



As an example, consider Figure 47.1.1. Using the coordinate system conventions shown, the CSG representation of the bracket is the expression

$$\begin{aligned}&\texttt{block}(8,3,1) \cup^* \texttt{move}(\texttt{block}(2,2.5,3),(0,4.5,1))\\ &\qquad\qquad -^* \texttt{move}(\texttt{cylinder}(0.75,1),(1.5,1.5,-0.5))\end{aligned}$$

where the * indicates a regularized operation. (See also Figure 33.4.1.)

The basic operations one wishes to perform on CSG representations are classifying points, curves, and surfaces with respect to a solid; detecting redundancies in the representation; and approximating CSG objects systematically.

More general primitives are obtained by considering the volume covered by sweeping a solid along a space curve, or sweeping a planar contour bounding an area. Defining a sweep is delicate, requiring many parameters to be exactly defined, but simple cases are widely used. They are extrusion, i.e., sweep along a straight line; and revolution, i.e., a sweep about an axis. The evaluation of general sweeps can be done by a number of methods. An even more general set of primitives is algebraic halfspaces, point sets defined by

$$P = \{(x, y, z) \in \mathbb{R}^3 \mid f(x, y, z) \leq 0\},$$

where $f(x, y, z)$ is an irreducible polynomial in x, y, and z.

More general operations are obtained by using nonregularizing Boolean operations or by defining a nonstandard semantics for Boolean operations on surfaces and curves.

47.1.2 BOUNDARY REPRESENTATION

In boundary representation (Brep) the solid surface is represented as a quilt of faces, edges, and vertices. A distinction is drawn between the topological entities, vertex, edge, and face, related to each other by incidence and adjacency, and the geometric location and shape of these entities. See also Figure 47.1.2. For example, when polyhedra are represented, the faces are polygons described geometrically by

a face equation plus a description of the polygon boundary. Geometrically, the entities in a Brep are not permitted to intersect anywhere except in edges and vertices that are explicitly represented in the topology data structure. In addition to the classification operations mentioned for CSG, Boolean union, intersection, and difference operations are usually implemented for Brep systems. Both regularized and nonregularizing Boolean operations may occur.

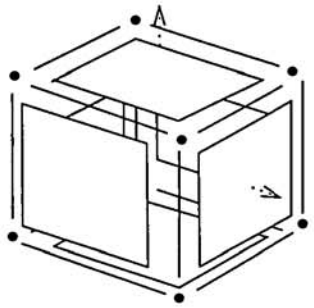


FIGURE 47.1.2
Topological entities of a box. Adjacency and incidence are recorded in Brep. Dotted arrows indicate face orientation.

Different Brep schemata appear in the literature, divided into two major families. One family restricts the solid surfaces to oriented manifolds. Here, every edge is incident to two faces, and every vertex is the apex of a single cone of incident edges and faces. The second family of Brep schemata allows oriented nonmanifolds in which edges are adjacent to an even number of faces. When these faces are ordered radially around the common edge, consecutive face pairs alternatingly bound solid interior and exterior. See Figure 47.1.3 for examples.

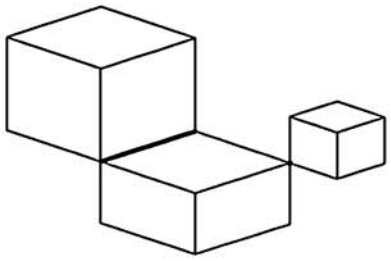


FIGURE 47.1.3
A nonmanifold solid without dangling or interior faces, edges, and vertices; the nonmanifold edges and vertices are drawn with a thicker pen.

More general nonmanifold Breps are used in systems that combine surface modeling with solid modeling. In such representation schemata, a solid may have interior (two-sided) faces, dangling edges, and so on. The current trend is to incorporate surface modeling capabilities into solid modelers.

The topology may be restricted in other ways. For instance, the interior of a face may be required to be homeomorphic to a disk, and edges required to have two distinct vertices. In that case, the Brep of a cylinder would have four faces, two planar and two curved. This may be desirable because of the geometric surface representation, or may be intended to simplify the algorithms operating on solids.

47.1.3 SPATIAL SUBDIVISION REPRESENTATIONS

GLOSSARY

Boundary conforming subdivision: Spatial subdivision of a solid that represents the boundary of the solid exactly.

Boundary approximating subdivision: Spatial subdivision that represents the boundary of the solid only approximately.

Regular subdivision: A subdivision whose cells are congruent. Grids are regular subdivisions.

Irregular subdivision: A subdivision with noncongruent cells.

Octree: Recursive selective subdivision of a cuboid volume into eight subcuboids.

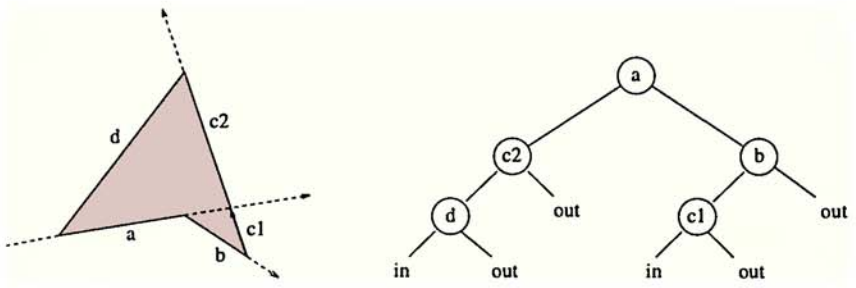
Binary space partition (BSP) tree: Recursive irregular subdivision of space, traditionally by halfplanes. See also Section 33.5.

Spatial subdivision decomposes a solid into cells, each with a simple topological structure and often also with a simple geometric structure. Subdivision representations are divided into *boundary conforming* and *boundary approximating*.

Important boundary conforming subdivision schemata are *meshes* and the *BSP tree*. Mesh representations are used in finite element analysis, a method for solving continuous physical problems. The mesh elements can be geometric tetrahedra, hexahedra, or other simple polyhedra, or they can be deformations of topological polyhedra so that curved boundaries can be approximated exactly. See Sections 22.4–5.

Binary space partition trees are recursive subdivisions of 3-space. Each interior node of the tree separates space into two disjoint point sets. In the simplest case, the root denotes a separator plane. All points of $\mathbb{R}^3$ below or on the plane are represented by one subtree, all points above the plane are represented by the other subtree. The two point sets are recursively subdivided by halfplanes at the subtree nodes. The leaves of the tree represent cells that are labeled IN or OUT. The (half) planes are usually face planes of a polyhedron, and the union of all cells labeled IN is the polyhedron. For an example in $\mathbb{R}^2$ see Figure 47.1.4. Note that algebraic halfspaces can be used as separators, so that curved solids can be represented exactly.

FIGURE 47.1.4
A polygon and a BSP tree representing it.



Boundary approximating representations are *grids* and *octrees*. In grids, space is subdivided in conformity with a coordinate system. For Cartesian coordinates, the division is into hexahedra whose sides are parallel to the coordinate planes. In cylindrical coordinate systems, the division is into concentric sectors, and so on. The grids may be regular or adaptive, and may be used to solve continuous physical problems by differencing schemes. Rectilinear grids that are geometrically deformed can be boundary-conforming. Otherwise, they approximate curved boundaries.

An octree divides a cube into eight subcubes. Each subcube may be further subdivided recursively. Cubes and their subdivision cubes are labeled white, black, or grey. A grey cube is one that has been subdivided and contains both white and black subcubes. A subcube is black if it is inside the solid to be represented, white if it is outside. Quadtrees, the two-dimensional analogue of octrees, are used in many geographical information systems. See Figure 33.5.1.

47.1.4 MEDIAL SURFACE REPRESENTATIONS

GLOSSARY

MAT: Medial axis transform, the two-dimensional version of the medial surface representation. Some authors use "medial axis transform" regardless of the dimension of the domain.

Maximal inscribed disk: Disk inscribed in a domain and not properly contained in another inscribed disk.

Medial axis and medial surface can unambiguously represent two-dimensional domains and three-dimensional solids, respectively. The representations are not widely used for this purpose at this time; more frequently they are used for shape recognition (see Section 43.4). However, as explained below, some sophisticated meshing algorithms are based on the medial axis and the medial surface.

The medial axis of a two-dimensional domain is defined as the closure of the locus of centers of disks inscribed within the domain. A disk is maximal if no other disk properly contains it. An example is shown in Figure 47.1.5 along with some maximal disks.

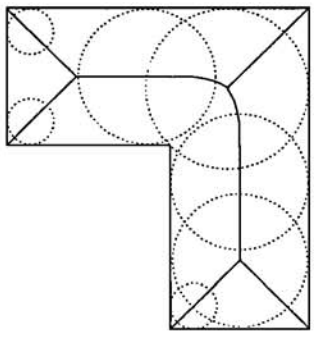


FIGURE 47.1.5
L-shaped domain and associated medial axis. Some maximal inscribed circles contributing to the medial axis are also shown.

The medial surface of a solid is the closure of the locus of centers of maximal

inscribed spheres. When we know the radius (the limit radius in case of closure points) of the corresponding sphere for each point on the medial surface, then an unambiguous solid representation is obtained that is sometimes called the medial axis transform (MAT). The MAT has a number of intriguing mathematical properties. For example, by enlarging the radius values by a constant, the MAT of a dilatation of the solid is obtained.

47.1.5 CONVERSION BETWEEN REPRESENTATIONS

Most solid modeling systems use Brep. Conversion from CSG to Brep is well understood, and is implemented as regularized Boolean operations on Brep solids. An extensive literature addresses these complex algorithms.

The conversion from Brep to CSG is not fully understood. In the polyhedral case, the conversion is essentially the same as the conversion from Brep to BSP tree. Conversion involving higher degree surfaces is largely open. However, some progress has been made recently by Naylor and Rogers in the case of Bézier curves and B-splines (for definitions, see Section 45.1). Roughly speaking, a coarse BSP tree is constructed that encloses sections of the curve in convex polygonal regions. On demand, the tree can be extended dynamically, thereby refining the enclosing regions. In this way, points may be classified efficiently with respect to the curve to a required resolution.

There are several algorithms for converting from CSG or Brep to the MAT. Some are based on geometric principles, some on a Delaunay triangulation of an approximated boundary, and some on a grid subdivision of ambient space. Because simple boundary geometry elements can produce very complicated curves and surface elements in the MAT, approximation approaches are favored in practice. The conversion from MAT to Brep has been addressed by Vermeer. Note that a polyhedral MAT produces a solid boundary that can contain spherical, conical, and cylindrical elements.

The conversion from CSG or Brep to mesh representations is a partially solved problem when the conversion is done for finite element analysis or other numerical treatment of continuum problems. In that context, the problem is not a geometric problem alone: the quality of the subdivision must also be judged by nongeometric criteria that come from the nature of the physical problem and the numerical algorithm used to solve it. Many approaches are based on octree subdivision, on Delaunay triangulation, and on MAT computations.

These conversion relationships are summarized in Table 47.1.1.

GEOMETRIC COVERAGE

The range and geometric representation of solid surfaces is referred to as "geometric coverage." Polyhedral modeling restricts to planes. Classical CSG allows only planes, cones, cylinders, spheres, and tori. Experimental modelers have been built allowing arbitrary algebraic halfspaces. Most commercial and many research modelers use B-splines (uniform or nonuniform, nonrational or rational) or Bézier surfaces. The properties and algorithmic treatment of these surfaces is studied by computer-aided geometric design. See Chapter 45 of this Handbook, as well as the monographs and surveys [Far88, Hos92, HL93].

TABLE 47.1.1 Representation conversion.

CONVERSION	REMARKS
CSG → Brep	Many methods, e.g., [Chi88, Hof89, Män88]. Active research seeks better tradeoff between speed, accuracy, and geometric coverage.
Brep → CSG	Largely open. Polyhedral case similar to BSP tree construction [Hof93b]; quadric cases treated in [Sha91, SV93]. See also [NR95] for parametric case.
Brep, CSG → MAT	[CHL91] uses grid approximation, [SAR95] uses Delaunay approximation of domain.
MAT → Brep	[Ver94] converts polyhedral MAT.
Brep, CSG → spatial subdivision	Many approaches; see, e.g., [Hof95, TWM85]. Active research seeks better techniques.

47.2 LEVELS OF ABSTRACTION

GLOSSARY

Substratum: Basic computational primitives of a solid modeler, such as incidence tests, vector arithmetic, etc.

Algorithmic infrastructure: Major algorithms implementing conceptual operations, such as surface intersection, edge blending, etc.

Graphical user interface (GUI): Visual presentation of the functionality of the system.

Application procedural interface (API): Presentation of system functionality in terms of methods and routines that can be included in user programs.

Substratum problem: Unreliability of logical decisions based on floating-point computations.

Large software systems should be structured into layers of abstraction. Doing so simplifies the implementation effort because the higher levels of abstraction can be compactly programmed in terms of the functionality of the lower levels. Moreover, the complexity of the system is reduced. A solid modeling system spans several levels of abstraction:

1. On the lowest level, there is the substratum of arithmetic and symbolic computations that are used as primitives by the algorithmic infrastructure. This level contains point and vector manipulation routines, incidence tests, and so on.

2. Next, there is an intermediate level comprising the algorithmic infrastructure. This level implements the conceptual operations available in the user interface, as well as a wide range of auxiliary tools needed by these operations. There is often an application programming interface available with which programs can be written that use the algorithmic infrastructure of the modeling system.

3. A graphical user interface (GUI) presents to the user a view of the functional capabilities of the system. Interaction with the GUI exercises these functions, for instance, for solid design. Tools for editing and archiving design are included.

Ideally, the levels of abstraction should be kept separate, with the higher levels leveraging the functionality of the lower levels. However, this separation is fundamentally limited by the interaction of numeric and symbolic computation.

47.2.1 THE SUBSTRATUM

The substratum consists of many low-level computations and tests; for example, vector computations, simple incidence tests, and computations for ordering points along a simple curve in space. Ideally, these operations create an abstract machine whose functionality simplifies the algorithms at the intermediate level of abstraction. But it turns out that this abstract machine is unreliable in a subtle way when implemented using floating-point arithmetic. Exact arithmetic would remedy this unreliability, but is widely held to be unacceptably inefficient when dealing with solids that have curved boundaries. See Section 35.4.

To illustrate how inexact arithmetic at the substratum level can impact the geometric computation, consider modeling polyhedral solids, the simplest possible situation for solid modeling. All computational decisions that arise in the course of a regularized Boolean operation on polyhedra can be reduced to determining the sign of 4×4 determinants. Geometrically, this is a test of whether a point is above, on, or below a plane. When the determinant's value is nearly zero, floating-point evaluation will decide based on a tolerance. But the decision is unreliable because logically equivalent tests may arise as different determinants in the course of the algorithm, and some of the determinants could have small, others large values. This gives an opportunity for the algorithm to build inconsistent data structures and fail. The problems are magnified when dealing with curved solids.

For a deeper evaluation of the problem, and for some approaches to solving it, see Chapter 35.

47.2.2 ALGORITHMIC INFRASTRUCTURE

Algorithmic infrastructure is a prominent research subject in solid modeling. Among the many questions addressed is the development of efficient and robust algorithms for carrying out the geometric computations that arise in solid modeling. The problems include point/solid classification, computing the intersection of two solids, determining the intersection of two surfaces, interpolating smooth surfaces to eliminate sharp edges on solids, and many more. See the reference section for a sampling of the literature.

There seems to be no academic work on structuring *application procedural interfaces* (API's) that encapsulate the functional capabilities of solid modelers so they can be used in other programs. Nevertheless, such API's play a prominent role in applications because they allow building on existing software functionality and constructing different abstraction hierarchies than the one implemented by a full-service solid modeling system.

An important consideration when devising infrastructure is that the algorithms

are often used by other programs, whether or not there is an API. Therefore, they must be ultra-reliable and in most cases must not require user intervention for exceptional situations.

The major geometric computations implemented at the infrastructure level have to balance the conflicting goals of efficiency, accuracy, and robustness. For this reason, many operations continue to be researched in efforts to seek new perceived optima. Moreover, new mathematical surface representations continue to be devised that necessitate different approaches. Some of the major operations on which research continues are the following.

Surface intersection. Given two bounded areas of two surfaces, determine all intersection curve components. A major difficulty of the problem is to identify correctly all components of the intersection, including isolated points and singularities. Since this computation is done in $\mathbb{R}^3$, classical algebraic geometry is of limited help.

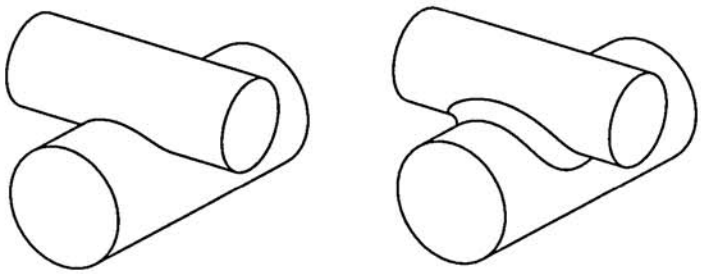
Offsetting. Given a surface, its offset is the set of all points that have fixed minimum distance from the surface. Offsets can have self-intersections that must be culled, and there is a technical relationship between offsetting and forming the MAT. Note that offsetting is used to determine certain blending surfaces, and is also used in the solid operation of *shelling* that creates thin-walled solids.

Blending. Given two intersecting surfaces, a third surface is interpolated between them to smooth the intersection edge. A simple example is shown in Figure 47.2.1.

Several issues arise. First, given two surfaces and contact curves on both, find a third surface touching the two given ones along the contact curves. Techniques exist both for implicit and for parametric surfaces. A second, less well-understood issue is how to devise the contact curves when dealing with solids, so that the curves connect properly at adjacent faces, behave correctly at vertices, and so on.

Deformations. Given a solid body, deform it locally or globally. The deformation could be required to obey constraints such as preserving volume or optimizing physical constraints. For example, we could deform the basic shape of a ship hull to minimize drag in fluids of various viscosities.

FIGURE 47.2.1
Left: two cylinders intersecting in a closed edge. Right: edge blended with a constant-radius, rolling-ball blend; the bounding curves of the blend are shown.



47.2.3 USER INTERFACES

Ultimately, the functional capabilities of a solid modeling system have to be presented to a user, typically through a *graphical user interface* (GUI). It would be a mistake to dismiss GUI design as a simple exercise. If the GUI merely presents the functionality of the infrastructure literally, an opportunity for operational leveraging has been lost. Instead, the GUI should conceptualize the functionalities an application needs. As in programming language design, this conceptual view can be convenient or inconvenient for a particular application. Research on GUI's therefore is largely done with a particular application area in mind.

For example, in mechanical engineering product design, an important aspect of the GUI might be to allow the user to specify the shape conveniently and precisely. This might be accomplished using geometric constraints and constraints of length, radius, and angle. In GUI's for virtual environment definition and navigation, on the other hand, approximate constraints and direct manipulation interfaces would be better.

47.3 FEATURES AND CONSTRAINTS

GLOSSARY

Form feature: Any stereotypical shape detail that has application significance.

Geometric constraints: Prescribed distance, angle, collinearity, concentricity, etc.

Generic design: Solid design with constraints and parameters without regard to specific values.

Design instance: Resulting solid after substituting specific values for parameters and constraints.

Parametric constraint solving: Solving a system of nonlinear equations that has a fixed triangular structure.

Variational constraint solving: Solving a system of nonlinear simultaneous equations.

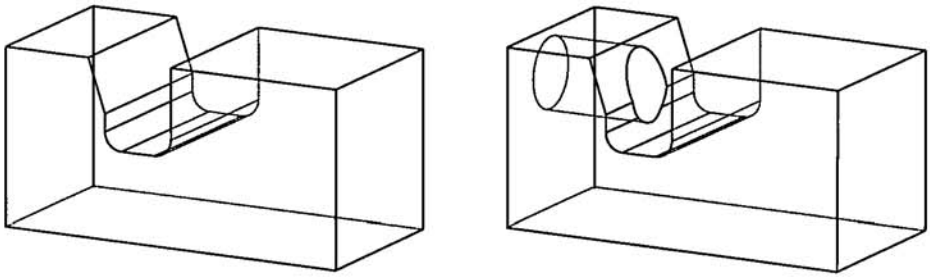
In solid modeling, two design paradigms are emerging for manufacturing applications, *feature-based design* and *constraint-based design.* The new paradigms expose a need to reconsider solid representations at a different level of abstraction. The representations reviewed before are for individual solids. However, we need to represent entire *classes* of solids, comprising a generic design. Roughly speaking, solids in a class are built structurally in the same way, from complex shape primitives, and are instantiated subject to constraints that interrelate specific shape elements and parameters. How these families should be defined precisely, how each generic design should be represented, and how designs should be edited are all important research issues of considerable depth.

47.3.1 FEATURE-BASED DESIGN

Feature-based design is usually understood to mean designing with shape elements such as slots, holes, pockets, etc., that have significance to manufacturing applications relating to function, manufacturing process, performance, cost, and so on. Focusing on shape primarily, we can conceptualize solid design in terms of three classes of features: generative, modifying, and referencing features. A feature is added to an existing design using attachment attributes and placement constraints. Subsequent editing may change both types of attachment information.

As an example, consider the solid shown to the right in Figure 47.3.1. A hole was added to the design on the left, and this could be specified by giving the diameter of the hole, placing its cross section, a circle, on the side face, and requiring that the hole extend to the next face. Should the slot at which the hole ends be moved or altered by subsequent editing, then the hole would automatically be adjusted to the required depth.

FIGURE 47.3.1
Left: solid block with a profiled slot. Right: After adding a hole with the attribute "through next face," an edited solid is obtained. If the slot is moved later, the hole will adjust automatically.

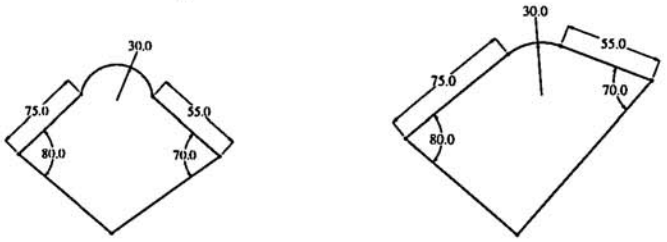


47.3.2 CONSTRAINT-BASED DESIGN

Constraint-based design refers to specifying shape with the help of constraints, when placing features or when defining shape parameters. For instance, assume that we are to design a cross section for use in defining a solid of revolution. A rough topological sketch is prepared (Figure 47.3.2, left), annotated with constraints, and instantiated to a sketch that satisfies the constraints exactly (Figure 47.3.2, right). Auxiliary geometric structures can be added, such as an axis of rotation. There is an extensive literature on constraint solving, from a variety of perspectives.

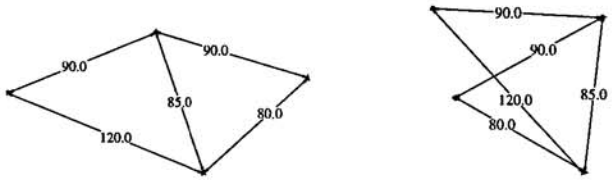
Increasingly, solid modeling systems use both features and constraints in the design interface. Often, the constraints on cross sections and other two-dimensional structures are unordered, but the constraints on three-dimensional geometry are usually considered in a fixed sequence. Solving systems of unordered constraints is sometimes referred to as *variational constraint solving.* Mathematically, it is equivalent to solving a system of nonlinear simultaneous equations. Solving constraints in a fixed sequence is also known as *parametric constraint solving.* The latter is equivalent to solving a system of nonlinear equations that has a fixed, triangular structure.

FIGURE 47.3.2
Geometric constraint solving. Input to the constraint solver shown on the left. Here, the arc should be tangent to the adjacent segments, and the two other segments should be perpendicular. Output of the constraint solver shown on the right.



A well-constrained geometric constraint problem corresponds naturally to a system of nonlinear algebraic equations with a finite set of solutions. In general, there will be several solutions of a single, well-constrained geometric problem. An example is shown in Figure 47.3.3. This raises the interesting question of exactly how a constraint solver should select one of those solutions efficiently, and why.

FIGURE 47.3.3
The well-constrained geometric problem of placing 4 points by 5 distances has two distinct solutions.



From symbolic computation we know that there are algorithms to convert a nontriangular system of nonlinear equations into a triangular system. The distinction between parametric and variational constraint solving is therefore artificial in theory. However, full-scale triangularization of systems of nonlinear equations is not tractable in many cases, so the distinction is relevant in practice. Moreover, a predetermined sequential evaluation of constraints is simple to implement and can be interfaced easily with conditional constraint evaluation, thereby increasing the expressive power of the constraint system without raising new semantic issues. For these reasons, many developers of solid modeling systems leverage core modeling capabilities by such extensions.

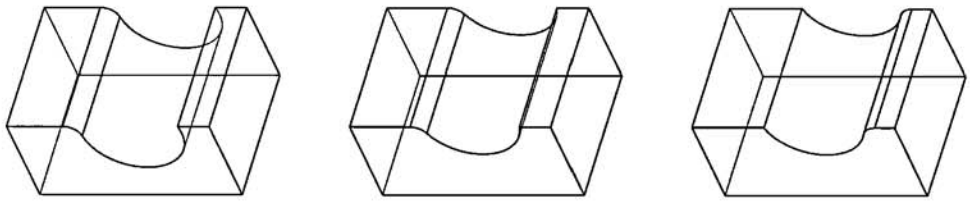
47.3.3 SEMANTIC PROBLEMS

When constraints and parameters are used in solid design, a ***generic design*** is obtained. Generic designs are instantiated by constraint values, and may be edited by changing the constraint values, the constraint schema, and the feature attributes. A design so edited can then be automatically re-instantiated by the solid modeler. A central difficulty in implementing this scenario, however, is that the generic design is usually defined visually on the basis of a particular instance, and when the design changes, the instance geometry is no longer present. Thus, visually

identified instance structures must be suitably described, so that re-instantiation can be carried out correctly.

As an example, consider the solid shown in Figure 47.3.4, left. It was constructed as follows. First, a rectangle was drawn and extruded into a block. On the front face of the block, a circle was drawn as a profile of a slot across the top of the block. Then, an edge was visually identified for rounding. This design is edited by altering the position of the circular slot profile. The edge to be rounded is not an explicit design entity, however. Hence, the edge has to be described implicitly, perhaps by the intersection of the circle and the top edge of the face on which the circle has been drawn. This description does not distinguish between the two straight edges of the slot, however, so additional information has to be used. Such information would have to allow a consistent identification under all possible constraint values, the difficult ***persistent naming*** problem.

FIGURE 47.3.4
A block with a slot and round on the left edge is shown left. After editing, in this case decreasing the depth of the slot, re-instantiation should produce the solid shown in the middle. However, some systems may re-instantiate as shown to the right, an error.



There seems to be little academic work on this topic, although it is of intense interest in applications. In particular, the formalization of the design information has profound implications on system architectures because it formalizes, in effect, the information flow between functional components. Whenever such formalization seeks independence from the specific implementation of the system components, system modularization is facilitated. Ultimately, this will accelerate the current trend of decomposing solid modeling systems into standardized components that can function interchangeably and can be combined in a variety of ways.

47.4 OPEN PROBLEMS

Most major problems in solid modeling contain a conceptualization aspect. That is, a precise, technical formulation of the problem commits to a specific conceptualization of the larger context that may be contentious. For example, consider the following technical problem. *Given an implicit algebraic surface S and a distance d, find the "offset" of S by d.* Assuming a precise definition of offset, and a restriction to irreducible algebraic surfaces S, the problem statement ignores the fact that a solid model is not bounded by a single, implicit surface, and that implicit surfaces of high algebraic degree may cause severe computational problems when used in a solid modeler.

CONSTRAINT SOLVING

Geometric constraint solvers trade efficiency for generality. Some very interesting techniques have been developed that are fast but not very general. They could be extended in various ways without substantially impacting on efficiency. Such extensions, for constraint solving in the plane, include the incorporation of parametric curve segments as geometric elements, more general constraint configurations, relations among distances, and angles. Spatial geometric constraint solving poses a number of open problems, including determining whether a constraint problem is generically well-constrained, or how to construct a line at prescribed distance from four fixed points.

FEATURES

Manufacturing applications need cogent definitions of features to accelerate design. Such definitions ought to be in terms of generic mechanisms of form and of function. Also needed are mapping algorithms interrelating different feature schemata.

SEMANTICS OF CONSTRAINT-BASED DESIGN

A solid shape design in terms of constraints can be changed simply by changing constraint values. To date, all such changes have been specified in terms of the procedures and algorithms that effect the change. What is needed is an abstract definition of shape change under such constraint changes to obtain a semantic definition of generic design and constraint-based editing. Such a definition must be visually intuitive.

47.5 SOURCES AND RELATED MATERIAL

FURTHER READING

Monographs on solid modeling. Monographs and surveys provide an excellent entry into solid modeling. Major monographs on solid modeling are [Chi88, Hof89, Män88]. Books on the related field of CAGD (computer-aided geometric design) may also contain material on solid modeling but concentrate primarily on curve and surface design and manipulation.

Surveys of solid modeling. Like monographs, surveys present a good road map for entry into the field. They are also more explicit on the perceived emerging trends and further developments. See [HV91, Req80, Req88].

Solid representations and conversion. There is a large and diverse literature on representations and representation conversion. Classical work focused primarily on the semantic foundations of CSG and Brep and includes [Req77, Wei86]. The mesh and octree representations are treated in [BN90, Hof95, Sam89a, Sam89b, TWM85], including the associated conversion problems. The medial axis representation of solids, and how to compute with it, are considered in [Hof92, SAR95, Ver94]. Implicit algebraic halfspaces as solid primitives are discussed in [BDL+91]. The

conversion between boundary representation and CSG can be considered a generalization of the binary space partition tree and is explored in [Hof93b, Nay90, NR95, Sha91, SV93]. Curve and surface representations, and their manipulation, are the subject of [Far88, Hos92, HL93]. More specialized treatment of offsets and sweeps is found in [BL90, CHL91].

Substratum, infrastructure and user interfaces. The substratum robustness issue is presented in greater depth in Chapter 35; [SI89] explores the use of exact arithmetic in polyhedral modeling. Infrastructure work is traditionally quite extensive. Surface intersection is treated in [Hos92]; this thesis contains an excellent summary of previous work. Global solid operations are considered in [BW89, For95, PS95, RSB95]; local solid operations are discussed in [HH87, Pet92]. User interface work seems to be almost neglected, but there are signs that this may change soon. Work in symbolic algebraic computation (Chapter 29) has foundational importance, for instance in regard to converting between surface representations. Some of the applications of symbolic computation are explored in [BCK88, Cho87, Hof90].

Features and constraints. Neither topic is new, so there is a sizable literature on both. The confluence of the two issues in recent solid modeling systems, however, is new. It raises a number of questions that have only recently been articulated and addressed. [SHL92, KRU94] discuss feature work. Constraints are the subject of [BFH+95, HV94, Kra92]. The confluence of the two strands and some of the implications are discussed in [HJ92]. Some of the technical issues that must be addressed are explained in [Hof93a, CH95], and there is more work emerging on this subject.

RELATED CHAPTERS

Chapter 29: Computational real algebraic geometry
Chapter 35: Robust geometric computation
Chapter 42: Computer graphics
Chapter 43: Pattern recognition
Chapter 45: Splines and geometric modeling

REFERENCES

[BCK88] B. Buchberger, G. Collins, and B. Kutzler. Algebraic methods for geometric reasoning. *Annu. Rev. Comput. Sci.*, 3:85–120, 1988.

[BDL+91] A. Bowyer, J.H. Davenport, and D.A. Lavender. A geometric algebra system. In D. Kapur, editor, *Integration of Symbolic and Numeric Methods.* MIT Press, Cambridge, 1991.

[BFH+95] W. Bouma, I. Fudos, C. Hoffmann, J. Cai, and R. Paige. A geometric constraint solver. *Comput. Aided Design*, 27:487–501, 1995.

[BL90] D. Blackmore and M. Leu. A differential equations approach to swept volume. In *Proc. 2nd Internat. Conf. Comput.-Integrated Manufac.*, pages 143–149, Troy, 1990.

[BN90] P. Brunet and I. Navazo. Solid representation and operation using extended octrees. *ACM Trans. Graph.*, 9:170–197, 1990.

[BW89] M. Bloor and M. Wilson. Generating blending surfaces with partial differential equations. *Comput. Aided Design*, 21:165–171, 1989.

[CH95] X. Chen and C. Hoffmann. Editing feature based design. *Comput. Aided Design*, 27:905–914, 1995.

[Chi88] H. Chiyokura. *Solid Modeling with Designbase.* Addison-Wesley, Reading, 1988.

[CHL91] C.-S. Chiang, C. Hoffmann, and R. Lynch. How to compute offsets without self-intersection. In *Proc. SPIE Conf. Curves Surf. Comput. Vision Graph.*, pages 76–87. Internat. Soc. Optical Engr., Boston, 1991.

[Cho87] C.-S. Chou. *Mechanical Theorem Proving.* Reidel, Dordrecht, 1987.

[Far88] G. Farin. *Curves and Surfaces for Computer-Aided Geometric Design.* Academic Press, New York, 1988.

[For95] M. Forsyth. Shelling and offsetting bodies. In *Proc. 3rd Annu. ACM Sympos. Solid Modeling*, pages 373–382. 1995.

[HH87] C. Hoffmann and J. Hopcroft. The potential method for blending surfaces and corners. In G. Farin, editor, *Geometric Modeling*, pages 347–365. SIAM, Philadelphia, 1987.

[HJ92] C. Hoffmann and R. Juan. Erep, an editable, high-level representation for geometric design and analysis. In P. Wilson, M. Wozny, and M. Pratt, editors, *Geometric Modeling for Product Realization*, pages 129–164. North Holland, Amsterdam, 1992.

[HL93] J. Hoschek and D. Lasser. *Computer Aided Geometric Design.* A K Peters, Boston, 1993.

[Hof89] C. Hoffmann. *Geometric and Solid Modeling.* Morgan Kaufmann, San Mateo, 1989.

[Hof90] C. Hoffmann. Algebraic and numerical techniques for offsets and blends. In S. Micchelli, M. Gasca, and W. Dahmen, editors, *Computations of Curves and Surfaces*, pages 499–528. Kluwer, Boston, 1990.

[Hof92] C. Hoffmann. Computer vision, descriptive geometry, and classical mechanics. In B. Falcidieno and I. Herman, editors, *Computer Graphics and Mathematics*, Eurographics Series, pages 229–244. Springer-Verlag, Berlin, 1992.

[Hof93a] C. Hoffmann. On the semantics of generative geometry representations. In *Proc. 19th ASME Design Automat. Conf.*, volume 2, pages 411–420, 1993.

[Hof93b] C. Hoffmann. On the separability problem of real functions and its significance in solid modeling. In K. Fischer, P. Loustaunau, J. Shapiro, E. Green, and D. Farkas, editors, *Computational Algebra*, volume 151 of *Lecture Notes in Pure and Appl. Math.*, pages 191–204. Marcel Dekker, New York, 1993.

[Hof95] C. Hoffmann. Geometric approaches to mesh generation. In I. Babuska, J. Flaherty, W. Henshaw, J. Hopcroft, J. Oliger, and T. Tezduyar, editors, *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations.* Springer-Verlag, New York, 1995.

[Hoh92] M. Hohmeyer. *Surface Intersection.* Ph.D. thesis, Dept. Comput. Sci., Univ. California, Berkeley, 1992.

[Hos92] M. Hosaka. *Modeling of Curves and Surfaces in CAD/CAM.* Springer-Verlag, New York, 1992.

[HV91] C. Hoffmann and G. Vaněček. Fundamental techniques for geometric and solid modeling. In C.T. Leondes, editor, *Advances in Control and Dynamics*, pages 101–165. Academic Press, New York, 1991.

[HV94] C. Hoffmann and P. Vermeer. Geometric constraint solving in R^2 and R^3. In D.Z. Du and F. Hwang, editors, *Computing in Euclidean Geometry*, 2nd edition. World Scientific, Singapore, 1994.

[Kra92] G. Kramer. *Solving Geometric Constraint Systems.* MIT Press, Cambridge, 1992.

[KRU94] F.-L. Krause, E. Rieger, and A. Ulbrich. Feature processing as kernel for integrated CAE systems. In *Proc. IFIP Internat. Conf.: Feature Modeling Recogn. Adv. CAD/CAM Syst.*, volume II, pages 693–716, Valciennes, 1994.

[Män88] M. Mäntylä. *An Introduction to Solid Modeling.* Computer Science Press, Rockville, 1988.

[Nay90] B. Naylor. Binary space partitioning trees as an alternative representation of polytopes. *Comput. Aided Design*, 22:250–252, 1990.

[NR95] B. Naylor and L. Rogers. Constructing paritioning trees from Bézier-curves for efficient intersections and visibility. In *Proc. Graphics Interface '95*, pages 44–55, 1995.

[Pet92] J. Peters. Joining smooth patches around a vertex to form a C^k surface. *Comput. Aided Geom. Design*, 9:387–411, 1992.

[PS95] A. Pasko and V. Savchenko. Algebraic sums for deformation of constructive solids. In *Proc. 3rd Annu. ACM Sympos. Solid Modeling*, pages 403–408. 1995.

[Req77] A. Requicha. Mathematical models of rigid solids. Tech. Rep. PAP Tech Memo 28, Univ. Rochester, 1977.

[Req80] A. Requicha. Representations for rigid solids: Theory, methods, and systems. *ACM Comput. Surv.*, 12:437–464, 1980.

[Req88] A. Requicha. Solid modeling—a 1988 update. In B. Ravani, editor, *CAD Based Programming for Sensory Robots*, pages 3–22. Springer-Verlag, New York, 1988.

[RSB95] A. Rappoport, A. Sheffer, and M. Bercovier. Volume-preserving free-form solids. In *Proc. 3rd Annu. ACM Sympos. Solid Modeling*, pages 361–372, 1995.

[Sam89a] H.J. Samet. *Applications of Spatial Data structures: Computer Graphics, Image Processing, and GIS.* Addison-Wesley, Reading, 1989.

[Sam89b] H.J. Samet. *Design and Analysis of Spatial Data Structures: Quadtrees, Octrees, and Other Hierarchical Methods.* Addison-Wesley, Reading, 1989.

[SAR95] D. Sheehy, C. Armstrong, and D Robinson. Computing the medial surface of a solid from a domain Delaunay triangulation. In *Proc. 3rd Annu. ACM Sympos. Solid Modeling*, pages 201–212, 1995.

[Sha91] V. Shapiro. *Representations of Semialgebraic Sets in Finite Algebras Generated by Space Decompositions.* Ph.D. thesis, Sibley School of Mech. Engr., Cornell Univ., Ithaca, 1991.

[SHL92] J. Shah, D. Hsiao, and J. Leonard. A systematic approach for design-manufacturing feature mapping. In P. Wilson, M. Wozny, and M. Pratt, editors, *Geometric Modeling for Product Realization*, pages 205–222. North Holland, Amsterdam, 1992.

[SI89] K. Sugihara and M. Iri. A solid modeling system free from topological inconsistency. *J. Inform. Process.*, 12:380–393, 1989.

[SV93] V. Shapiro and D. Vossler. Separation for boundary to CSG conversion. *ACM Trans. Graph.*, 12:35–55, 1993.

[TWM85] J. Thompson, Z. Warsi, and W. Mastin. *Numerical Grid Generation.* North Holland, Amsterdam, 1985.

[Ver94] P. Vermeer. *Medial Axis Transform to Boundary Representation Conversion.* Ph.D. thesis, Dept. Comput. Sci., Purdue Univ., Lafayette, 1994.

[Wei86] K. Weiler. *Topological Structures for Geometric Modeling.* Ph.D. thesis, Dept. Comput. Syst. Engr., Rensselaer Polytechnic Inst., Troy, 1986.

48 GEOMETRIC APPLICATIONS OF THE GRASSMANN-CAYLEY ALGEBRA

Neil L. White

INTRODUCTION

Grassmann-Cayley algebra is first and foremost a means of translating synthetic projective geometric statements into invariant algebraic statements in the bracket ring, which is the ring of projective invariants. A general philosophical principle of invariant theory, sometimes referred to as ***Gram's theorem***, says that any projectively invariant geometric statement has an equivalent expression in the bracket ring; thus we are providing here the practical means to carry this out. We give an introduction to the basic concepts, and illustrate the method with several examples from projective geometry, rigidity theory, and robotics.

48.1 BASIC CONCEPTS

Let P be a $(d-1)$-dimensional projective space over the field F, and V the canonically associated d-dimensional vector space over F. Let S be a finite set of n points in P and, for each point, fix a homogeneous coordinate vector in V. We assume that S spans V, hence also that $n \geq d$. Initially, we choose all of the coordinates to be distinct, algebraically independent indeterminates in F, although we can always specialize to the actual coordinates we want in applications. For $p_i \in S$, let the coordinate vector be $(x_{1,i}, \ldots, x_{d,i})$.

GLOSSARY

Bracket: A $d \times d$ determinant of the homogeneous coordinate vectors of d points in S. Brackets are relative projective invariants, meaning that under projective transformations their value changes only in a very predictable way (in fact, under a basis change of determinant 1, they are literally invariant). Hence brackets may also be thought of as coordinate-free symbolic expressions. The bracket of $u_1, \ldots, u_d$ is denoted by $[u_1, \ldots, u_d]$.

Bracket ring: The ring B generated by the set of all brackets of d-tuples of points in S, where $n = |S| \geq d$. It is a subring of the ring $F[x_{1,1}, x_{1,2}, \ldots, x_{d,n}]$ of polynomials in the coordinates of points in S.

Straightening algorithm: A normal form algorithm in the bracket ring.

Join of points: An exterior product of k points, $k \leq d$, computed in the exterior algebra of V. We denote such a product by $a_1 \vee a_2 \vee \cdots \vee a_k$, or simply $a_1 a_2 \cdots a_k$, rather than $a_1 \wedge a_2 \wedge \cdots \wedge a_k$, which is commonly used in exterior algebra. A

concrete version of this operation is to compute the Plücker coordinate vector of (the subspace spanned by) the k points, that is, the vector whose components are all $k \times k$ minors (in some prespecified order) of the $d \times k$ matrix whose columns are the homogeneous coordinates of the k points.

Extensor of step k*, or *decomposable k-tensor: A join of k points. Extensors of step k span a vector space $V^{(k)}$ of dimension $\binom{d}{k}$. (Note that not every element of $V^{(k)}$ is an extensor.)

Antisymmetric tensor: Any element of the direct sum $\Lambda V = \oplus_k V^{(k)}$.

Copoint: Any antisymmetric tensor of step $d-1$. A copoint is always an extensor.

Join: The exterior product operation on ΛV. The join of two tensors can always be reduced by distributivity to a linear combination of joins of points.

Integral: $E = u_1 u_2 \cdots u_d$, for any vectors $u_1, u_2, \ldots, u_d$ such that $[u_1, u_2, \ldots, u_d] = 1$. Every extensor of step d is a scalar multiple of the integral E.

Meet: If $A = a_1 a_2 \cdots a_j$ and $B = b_1 b_2 \cdots b_k$, with $j + k \geq d$, then

$$A \wedge B = \sum_{\sigma} \operatorname{sgn}(\sigma)[a_{\sigma(1)}, \ldots, a_{\sigma(d-k)}, b_1, \ldots, b_k] a_{\sigma(d-k+1)} \cdots a_{\sigma(j)}$$
$$\equiv [\overset{\bullet}{a}_1, \ldots, \overset{\bullet}{a}_{d-k}, b_1, \ldots, b_k] \overset{\bullet}{a}_{d-k+1} \cdots \overset{\bullet}{a}_j \ .$$

The sum is taken over all permutations σ of $\{1, 2, \ldots, j\}$ such that $\sigma(1) < \sigma(2) < \cdots < \sigma(d-k)$ and $\sigma(d-k+1) < \sigma(d-k+2) < \cdots < \sigma(j)$. Each such permutation is called a ***shuffle*** of the $(d-k, j-(d-k))$ split of A, and the dots represent such a signed sum over all the shuffles of the dotted symbols.

Grassmann-Cayley algebra: The vector space $\Lambda(V)$ together with the operations $\vee$ and $\wedge$.

PROPERTIES OF GRASSMANN-CAYLEY ALGEBRA

(i) $A \vee B = (-1)^{jk} B \vee A$ and $A \wedge B = (-1)^{(d-k)(d-j)} B \wedge A$, if A and B are extensors of steps j and k.

(ii) $\vee$ and $\wedge$ are associative and distributive over addition and scalar multiplication.

(iii) $A \vee B = (A \wedge B) \vee E$ if $\operatorname{step}(A) + \operatorname{step}(B) = d$.

(iv) A meet of two extensors is again an extensor.

(v) The meet is dual to the join, where duality exchanges points and copoints.

(vi) ***Alternative Law:*** Let $a_1, a_2, \ldots, a_k$ be points and $\gamma_1, \gamma_2, \ldots, \gamma_s$ copoints. Then if $k \geq s$,

$$(a_1 a_2 \cdots a_k) \wedge (\gamma_1 \wedge \gamma_2 \wedge \cdots \wedge \gamma_s) = [\overset{\bullet}{a}_1, \gamma_1][\overset{\bullet}{a}_2, \gamma_2] \cdots [\overset{\bullet}{a}_s, \gamma_s] \overset{\bullet}{a}_{s+1} \vee \cdots \vee \overset{\bullet}{a}_k \ .$$

Here the dots refer to all shuffles over the $(1, 1, \ldots, 1, k-s)$ split of $a_1 \cdots a_k$, that is, a signed sum over all permutations of the a's such that the last $k-s$ of them are in increasing order.

48.2 GEOMETRY ↔ G.-C. ALGEBRA → BRACKET ALGEBRA

If X is a projective subspace of dimension $k-1$, pick a basis $a_1, a_2, \ldots, a_k$ and let $A = a_1 a_2 \cdots a_k$ be an extensor. We call $X = \overline{A}$ the ***support*** of A.

(i) If $A \neq 0$ is an extensor, then A determines $\overline{A}$ uniquely.

(ii) If $\overline{A} \cap \overline{B} \neq \emptyset$, then $\overline{A \vee B} = \overline{A} + \overline{B}$.

(iii) If $\overline{A} \cup \overline{B}$ spans V, then $\overline{A \wedge B} = \overline{A} \cap \overline{B}$.

TABLE 48.2.1 Examples of geometric conditions and corresponding Grassmann-Cayley algebra statements.

GEOMETRIC CONDITION	DIM	G.-C. ALGEBRA STATEMENT	BRACKET STATEMENT
Point $\overline{a}$ is on the line $\overline{bc}$ (or $\overline{b}$ is on $\overline{ac}$, etc.)	2	$a \wedge bc = 0$	$[abc] = 0$
Lines $\overline{ab}$ and $\overline{cd}$ intersect	3	$ab \wedge cd = 0$	$[abcd] = 0$
Lines $\overline{ab}$, $\overline{cd}$, $\overline{ef}$ concur	2	$ab \wedge cd \wedge ef = 0$	$[\overset{\bullet}{a}cd][\overset{\bullet}{b}ef] = 0$
Planes $\overline{abc}$, $\overline{def}$, and $\overline{ghi}$ have a line in common	3	$abc \wedge def \wedge ghi = 0$	$[\overset{\bullet}{a}def][\overset{\bullet}{b}ghi][\overset{\bullet}{c}xyz] = 0 \ \forall x, y, z$
The intersections of $\overline{ab}$ with $\overline{cd}$ and of $\overline{ef}$ with $\overline{gh}$ are collinear with $\overline{i}$	2	$(ab \wedge cd) \vee (ef \wedge gh) \vee i = 0$	$[\overset{\bullet}{a}cd][\overset{\diamond}{e}gh][\overset{\bullet}{b}\overset{\diamond}{f}i] = 0$

The geometric conditions in Table 48.2.1 should be interpreted projectively. For example, the concurrency of three lines includes as a special case that the three lines are mutually parallel, if one prefers to interpret the conditions in affine space. Degenerate cases are always included, so that the concurrency of three lines includes as a special case the equality of two or even all three of the lines, for example.

Most of the interesting geometric conditions translate into Grassmann-Cayley conditions of step 0 (or, equivalently, step d), and therefore expand into bracket conditions directly. When the Grassmann-Cayley condition is not of step 0, as in the example in Table 48.2.1 of three planes in three-space containing a common line, then the Grassmann-Cayley condition may be joined with an appropriate number of universally quantified points to get a conjunction of bracket conditions. The joined points may also be required to come from a specified basis to make this a conjunction of a finite number of bracket conditions.

In this fashion, any incidence relation in projective geometry may be translated into a conjunction of Grassmann-Cayley statements, and, conversely, Grassmann-Cayley statements may be translated back to projective geometry just as easily, provided they involve only join and meet, not addition.

Many identities in the Grassmann-Cayley algebra yield algebraic, coordinate-free proofs of important geometric theorems. These proofs typically take the form "the left-hand side of the identity is 0 if and only if the right-hand side of the identity is 0," and the resulting equivalent Grassmann-Cayley conditions translate to interesting geometric conditions as above.

TABLE 48.2.2 Examples of Grassmann-Cayley identities and corresponding geometric theorems, in dimension 2.

GEOMETRIC THEOREM	G.-C. ALGEBRA IDENTITY
Desargues's theorem: Derived points $ab \wedge a'b'$, $ac \wedge a'c'$, and $bc \wedge b'c'$ are collinear if and only if abc or $a'b'c'$ are collinear or aa', bb', and cc' concur.	$(ab \wedge a'b') \vee (ac \wedge a'c') \vee (bc \wedge b'c') = [abc][a'b'c'](aa' \wedge bb' \wedge cc')$
Pappus's theorem and Pascal's theorem: If abc and $a'b'c'$ are both collinear sets, then $(bc' \wedge b'c)$, $(ca' \wedge c'a)$, and $(ab' \wedge a'b)$ are collinear.	$[ab'c'][a'bc'][a'b'c]\underline{[abc]} - [abc'][ab'c]\underline{[a'bc]}\underline{[a'b'c']} = (bc' \wedge b'c) \vee (ca' \wedge c'a) \vee (ab' \wedge a'b)$
Pappus's theorem (alternate version): If $aa'x$, $bb'x$, $cc'x$, $ab'y$, $bc'y$, and $ca'y$ are collinear, then ac', ba', cb' concur.	$aa' \wedge bb' \wedge cc' + ab' \wedge bc' \wedge ca' + ac' \wedge ba' \wedge cb' = 0$
Fano's theorem: If no three of a, b, c, d are collinear, then $(ab \wedge cd)$, $(bc \wedge ad)$, and $(ca \wedge bd)$ are collinear if and only if char $F = 2$.	$(ab \wedge cd) \vee (bc \wedge ad) \vee (ca \wedge bd) = 2\,[abc][abd][acd][bcd]$

The identities in Table 48.2.2 are proved by expanding both sides, using the rules for join and meet, and then verifying the equality of the resulting expressions by using the straightening algorithm of bracket algebra (see [Stu93]).

The right-hand side of the identity for the first version of Pappus's theorem is also the Grassmann-Cayley form of the geometric construction used in Pascal's theorem, and hence is 0 if and only if the six points lie on a common conic (Pappus's theorem being the degenerate case of Pascal's theorem in which the conic consists of two lines). Hence the left-hand side of the same identity is the bracket expression that is 0 if and only if the six points lie on a common conic. In particular, if abc and $a'b'c'$ are both collinear, we see immediately from the underlined brackets that the left-hand side is 0.

Numerous other projective geometry incidence theorems may be proved using the Grassmann-Cayley algebra. We illustrate this with an example modified from [RS76]. Other examples may be found in the same reference.

THEOREM 48.2.1

In 3-space, if triangles abc and $a'b'c'$ are in perspective from the point d, then the lines $a'bc \wedge ab'c'$, $b'ca \wedge bc'a'$, $c'ab \wedge ca'b'$, and $a'b'c' \wedge abc$ are all coplanar.

Proof. We prove the general case, where a, b, c, d, a', b', c' are all distinct, triangles abc and $a'b'c'$ are nondegenerate, and d is in neither the plane abc nor the plane $a'b'c'$. Then, since a, a', d are collinear, we may write $\alpha a' = \beta a + d$ for nonzero scalars α and β. Since we are using homogeneous coordinates for points, a, and similarly a', may be replaced by nonzero scalar multiples of themselves without changing the geometry. Thus, without loss of generality, we may write $a' = a + d$. Similarly, $b' = b + d$ and $c' = c + d$. Now

$$L_1 := a'bc \wedge ab'c' = [a'ab'c']bc - [bab'c']a'c + [cab'c']a'b$$
$$= [dabc]bc + [badc]ca + [cabd]ab + [badc]cd + [cabd]db$$
$$= [abcd](-bc - ac + ab + cd - bd).$$

Similarly,

$$L_2 := b'ca \wedge bc'a' = [abcd](ac + ab + bc + ad - cd),$$
$$L_3 := c'ab \wedge ca'b' = [abcd](-ab + bc - ac + bd - ad),$$
$$L_4 := a'b'c' \wedge abc = [abcd](bc - ac + ab).$$

Now we check that any two of these lines intersect. For example,

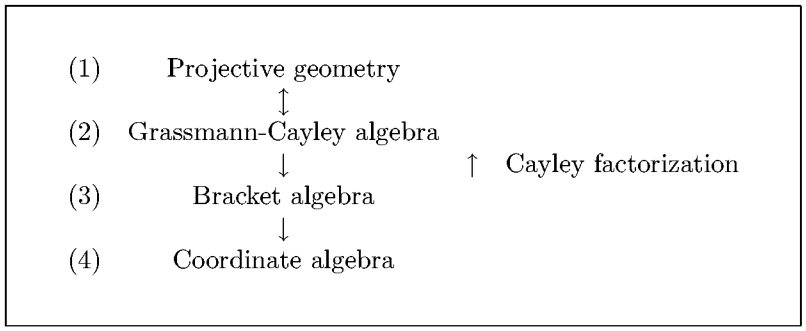
$$L_1 \wedge L_2 = [abcd]^2(-bc - ac + ab + cd - bd) \wedge (ac + ab + bc + ad - cd) = 0.$$

However, this shows only that either all four lines are coplanar or all four lines concur. To prove the former, it suffices to check that the intersection of $\overline{L_1}$ and $\overline{L_4}$ is distinct from that of $\overline{L_2}$ and $\overline{L_4}$. Notice that $L_1 \wedge L_4$ does not tell us the point of intersection, because $\overline{L_1}$ and $\overline{L_4}$ do not jointly span V, by our previous computation. But if we choose a generic vector x representing a point in general position, it follows from $\overline{L_1} \neq \overline{L_4}$, which must hold in our general case, that $(L_1 \vee x) \wedge L_4$ is nonzero and does represent the desired point of intersection. Then we compute

$$\begin{aligned}(L_1 \vee x) \wedge L_4 &= [abcd]^2(-bcx - acx + abx + cdx - bdx) \wedge (bc - ac + ab) \\ &= [abcd]^2(2[abcx] - [bcdx] - [acdx])(c - b) \\ &= \alpha(c - b)\end{aligned}$$

for some nonzero scalar α. Similarly, $(L_2 \vee x) \wedge L_4 = \beta(c - a)$ for some nonzero scalar β. By the nondegeneracy of the triangle abc, these two points of intersection are distinct. □

48.3 CAYLEY FACTORIZATION: BRACKET ALGEBRA → GEOMETRY



(1)↔(2)→(3) in the chart above is explained in Section 48.2 above, with (2)→(1) being straightforward only in the case of a Grassmann-Cayley expression involving only joins and meets. (3)→(4) is the trivial expansion of a determinant into a polynomial in its d^2 entries. (4)→(3) is possible only for invariant polynomials (under the special linear group); see [Stu93] for an algorithm.

PHILOSOPHY OF INVARIANT THEORY: It is best for many purposes to avoid level (4), and to work instead with the symbolic coordinate-free expressions on levels (2) and (3).

Cayley factorization, (3)→(2), refers to the translation of a bracket polynomial into an equivalent Grassmann-Cayley expression involving only joins and meets. The input polynomial must be homogeneous (i.e., each point must occur the same number of times in the brackets of each bracket monomial of the polynomial), and Cayley factorization is not always possible. No practical algorithm is known in general, but an algorithm [Whi91] is known that finds such a factorization—or else announces its impossibility—in the multilinear case (each point occurs exactly once in each monomial). This algorithm is practical up to about 20 points.

MULTILINEAR CAYLEY FACTORIZATION

The multilinear Cayley factorization (MCF) algorithm is too complex to present here in detail; instead, we give an example and indicate roughly how the algorithm proceeds on the example.

Let

$$\begin{aligned} P \;=\; & -[acj][deh][bfg] - [cdj][aeh][bfg] - [cdj][abe][fgh] \\ & + [acj][bdf][egh] - [acj][bdg][efh] + [acj][bdh][efg]. \end{aligned}$$

Note that P is multilinear in the 9 points. The MCF algorithm now looks for sets of points $x, y, \ldots, z$ such that the extensor $xy \cdots z$ could be part of a Cayley factorization of P. For this choice of P, it turns out that no such set larger than a pair of elements occurs. An example of such a pair is a, d; in fact, if d is replaced by a in P, leaving two a's in each term of P, although in different brackets, the resulting bracket polynomial is equal to 0, as can be verified using the straightening algorithm. The MCF algorithm, using the straightening algorithm as a subroutine, finds that $(a, d), (b, h), (c, j), (f, g)$ are all the pairs with this property.

The algorithm now looks for combinations of these extensors that could appear as a meet in a Cayley factorization of P. (For details, see [Whi91].) It finds in our example that $ad \wedge cj$ is such a combination. As soon as a single such combination is found, an algebraic substitution involving a new variable, $z = ad \wedge cj$, is performed, and a new bracket polynomial of smaller degree involving this new variable is derived; the algorithm then begins anew on this polynomial. If no such combination is found, the input bracket polynomial is then known to have no Cayley factorization. In our example, this derived polynomial turns out to be $P = [zef][gbh] - [zeg][fbh]$, which of necessity is still multilinear. The MCF algorithm proceeds to find (and we can directly see by consulting Table 48.2.1) that $P = ze \wedge fg \wedge bh$. Thus, our final Cayley factorization is output as

$$P = ((ad \wedge cj) \vee e) \wedge fg \wedge bh.$$

It is significant that this algorithm requires no backtracking. For example, once $ad \wedge cj$ is found as a possible meet in a Cayley factorization of P, it is known that if P has a Cayley factorization at all, then it must also have one using the factor $ad \wedge cj$; hence we are justified in factoring it out, i.e., substituting a new variable for it. Other Cayley factorizations may be possible, for example,

$$P = ((fg \wedge bh) \vee (ad \wedge cj)) \wedge e.$$

Note that these two factorizations have the same geometric meaning.

48.4 APPLICATIONS

48.4.1 ROBOTICS

GLOSSARY

Robot arm: A set of rigid bodies, or links, connected in series by joints that allow relative movement of the successive links, as described below. The first link is regarded as fixed in position, or tied to the ground, while the last link, called the ***end-effector***, is the one that grasps objects or performs tasks.

Revolute joint: A joint between two successive links of a robot arm that allows only a rotation between them. In simpler terms, a revolute joint is a hinge connecting two links.

Prismatic joint: A joint between two successive links of a robot arm that allows only a translational movement between the two links.

Screw joint: A joint between two successive links of a robot arm that allows only a screw movement between the two links.

TABLE 48.4.1 Modeling instantaneous robotics.

ROBOTICS CONCEPT	GRASSMANN-CAYLEY EQUIVALENT
Revolute joint on axis $\overline{ab}$	$\alpha(a \vee b)$, a 2-extensor
Rotation about line $\overline{ab}$	$\beta(a \vee b)$
Motion of point p in rotation about line $\overline{ab}$	$\beta(a \vee b) \vee p$
Screw joint	indecomposable 2-tensor
Prismatic joint	2-extensor at infinity
Motion space of the end-effector, where $j_1, j_2, \ldots, j_k$ are joints in series	span of the extensors $< j_1, j_2, \ldots, j_k >$

We are considering here only the instantaneous kinematics or statics of robot arms, that is, positions and motions at a given instant in time. A robot arm has a ***critical configuration*** if the joint extensors become linearly dependent. If the arm has six joints in three-space, a critical configuration means a loss of full mobility. If the arm has a larger number of joints, criticality is defined as any six of the joint extensors becoming linearly dependent. This can mean severe problems with the driving program in real-life robots, even when the motion space retains full dimensionality.

In one sense, criticality is trivial to determine, since we need only compute a determinant function, called the ***superbracket***, on the six-dimensional space $\Lambda^2(V)$. However, if we want to know all the critical configurations of a given robot arm, this becomes a nontrivial question, that of determining all of the zeroes of the superbracket. To answer it, we need to express the superbracket in terms of

ordinary brackets. This has been done in [MW91], where the superbracket of the six 2-extensors $a_1a_2, b_1b_2, \ldots, f_1f_2$ is given by

$$\begin{aligned}
&[[a_1a_2, b_1b_2, c_1c_2, d_1d_2, e_1e_2, f_1f_2]] \;= \\
&\quad - [a_1a_2b_1b_2][c_1c_2\,\overset{\bullet}{d_1}\overset{\diamond}{e_1}][\overset{\bullet}{d_2}\overset{\diamond}{e_2}\,f_1f_2] \\
&\quad + [a_1a_2\overset{\bullet}{b_1}\overset{\diamond}{c_1}][\overset{\bullet}{b_2}\overset{\diamond}{c_2}\,d_1d_2][e_1e_2f_1f_2] \\
&\quad - [a_1a_2\overset{\bullet}{b_1}\overset{\diamond}{c_1}][\overset{\bullet}{b_2}\,d_1d_2\,\overset{\triangleleft}{e_1}][\overset{\diamond}{c_2}\overset{\triangleleft}{e_2}\,f_1f_2] \\
&\quad + [a_1a_2\overset{\bullet}{b_1}\overset{\diamond}{d_1}][\overset{\bullet}{b_2}\,c_1c_2\,\overset{\triangleleft}{e_1}][\overset{\diamond}{d_2}\overset{\triangleleft}{e_2}\,f_1f_2].
\end{aligned}$$

(Here the dots, diamonds, and triangles have the same meaning as the dots in Section 48.1.)

Consider the particular example of the six-revolute-joint robot arm illustrated in Figure 48.4.1, whose first two joints lie on intersecting lines, whose third and fourth joints are parallel, and whose last two joints also lie on intersecting lines. The larger cylinders in the figure represent the revolute joints. To express the superbracket, we must choose two points on each joint axis. We may choose $b_1 = a_2$, $d_1 = c_2$ (where this point is at infinity), and $f_1 = e_2$, as shown by the black dots. The thin cylinders represent the links; for example, the first link, between a_2 and b_2, is connected to the ground (not shown) by joint a_1a_2, and can therefore only rotate around the axis $\overline{a_1a_2}$.

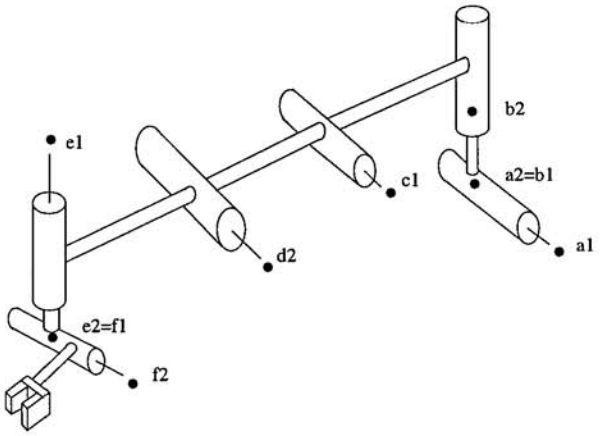


FIGURE 48.4.1
Six-revolute-joint robot arm.

Plugging in and deleting terms with a repeated point inside a bracket, we get

$$- [a_1a_2b_2\overset{\bullet}{c_1}][a_2c_2d_2e_2][\overset{\bullet}{c_2}e_1e_2f_2] \tag{48.4.1}$$

$$+ [a_1a_2b_2\overset{\bullet}{c_2}][a_2c_1c_2e_2][\overset{\bullet}{d_2}e_1e_2f_2] \tag{48.4.2}$$

$$= [\overset{\bullet}{c_1}a_1a_2b_2][\overset{\bullet}{d_2}a_2c_2e_2][\overset{\bullet}{c_2}e_1e_2f_2]\,, \tag{48.4.3}$$

where each of (48.4.1) and (48.4.2) has two terms because of the dotting, and the same four terms constitute (48.4.3), since two of the six terms generated by the dotting are zero because of the repetition of c_2 in the second bracket.

Finally, we recognize (48.4.3) as the bracket expansion of

$$(c_1 d_2 c_2) \wedge (a_1 a_2 b_2) \wedge (a_2 c_2 e_2) \wedge (e_1 e_2 f_2).$$

We then recognize that the geometric conditions for criticality are any positions that make this Grassmann-Cayley expression 0, namely

(i) one or more of the planes $\overline{c_1 c_2 d_2}, \overline{a_1 a_2 b_2}, \overline{a_2 c_2 e_2}, \overline{e_1 e_2 f_2}$ is degenerate, or

(ii) the four planes have nonempty intersection.

Notice that in an actual robot arm of the type we are considering, none of the degeneracies in (i) can actually occur.

See Section 41.1 for more information.

48.4.2 BAR FRAMEWORKS

Consider a generically $(d-1)$-isostatic graph G (see Section 49.1 of this Handbook), that is, a graph for which almost all realizations in $(d-1)$-space as a bar framework are minimally first-order rigid. Since first-order rigidity is a projective invariant (see Theorem 49.1.23), we would like to know the projective geometric conditions that characterize all of its nonrigid (first-order flexible) realizations. By Gram's theorem, these conditions must be expressible in terms of bracket conditions, and [WW83] shows that the first-order flexible realizations are characterized by the zeroes of a single bracket polynomial C_G, called the *pure condition* (see Theorem 49.1.25). Furthermore, [WW83] gives an algorithm to construct the pure condition C_G directly from the graph G. Then we require Cayley factorization to recover the geometric incidence condition, if it is not already known. Consider the following examples, illustrated in Figure 48.4.2.

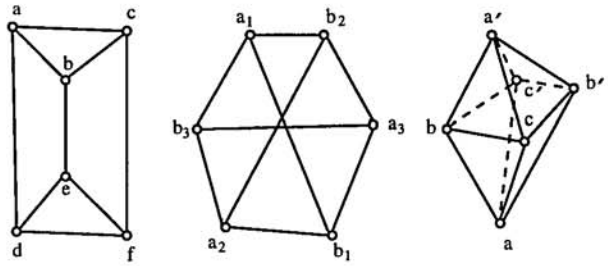


FIGURE 48.4.2
Three examples of bar frameworks. (i) (ii) (iii)

(i) The graph G is the edge skeleton of a triangular prism, realized in the plane. We have $C_G = [abc][def]([abe][dfc] - [dbe][afc])$, and we may recognize the factor in parentheses as the third example in Table 48.2.1. Thus $C_G = 0$, and the framework is first-order flexible, if and only if one of the triangles $\overline{abc}$ or $\overline{def}$ is degenerate, or the three lines $\overline{ad}$, $\overline{be}$, $\overline{cf}$ are concurrent, or one or more of these lines is degenerate.

(ii) The graph G is $K_{3,3}$, a complete bipartite graph, realized in the plane. Then $C_G = [a_1 a_2 a_3][a_1 b_2 b_3][b_1 a_2 b_3][b_1 b_2 a_3] - [b_1 b_2 b_3][b_1 a_2 a_3][a_1 b_2 a_3][a_1 a_2 b_3]$, and

this is the second example in Table 48.2.2. Thus $C_G = 0$, and the framework is first-order flexible, if and only if the six points lie on a common conic or, equivalently by Pascal's theorem, the three points $\overline{a_1b_2 \wedge a_2b_1}$, $\overline{a_1b_3 \wedge a_3b_1}$, $\overline{a_2b_3 \wedge a_3b_2}$ are collinear.

(iii) The graph G is the edge skeleton of an octahedron, realized in Euclidean 3-space. Then $C_G = [abc'a'][bca'b'][cab'c'] + [abc'b'][bca'c'][cab'a']$, and this can be recognized directly as the expansion of the Grassmann-Cayley expression $abc \wedge a'bc' \wedge a'b'c \wedge ab'c'$. Thus $C_G = 0$, and the framework is first-order flexible, if and only if the four alternating octahedral face planes $\overline{abc}$, $\overline{a'bc'}$, $\overline{a'b'c}$, and $\overline{ab'c'}$ concur, or any one or more of these planes is degenerate. This, in turn, is equivalent to the same condition on the other four face planes, $\overline{abc'}$, $\overline{ab'c}$, $\overline{a'bc}$, $\overline{a'b'c'}$.

48.4.3 BAR-AND-BODY FRAMEWORKS

A ***bar-and-body framework*** consists of a finite number of $(d-1)$-dimensional rigid bodies, free to move in Euclidean $(d-1)$-space, and connected by rigid bars, with the connections at the ends of each bar allowing free rotation of the bar relative to the rigid body; i.e., the connections are "universal joints." Each rigid body may be replaced by a first-order rigid bar framework in such a way that the result is one large bar framework, thus in one sense reducing the study of bar-and-body frameworks to that of bar frameworks. Nevertheless, the combinatorics of bar-and-body frameworks is quite different from that of bar frameworks, since the original rigid bodies are not allowed to become first-order flexible in any realization, contrary to the case with bar frameworks.

A generically isostatic bar-and-body framework has a pure condition, just as a bar framework has, whose zeroes are precisely the special positions in which the framework has a first-order flex. However, this pure condition is a bracket polynomial in the *bars* of the framework, as opposed to a bracket polynomial in the vertices, as was the case with bar frameworks. An algorithm to directly compute the pure condition for a bar-and-body framework, somewhat similar to that for bar frameworks, is given in [WW87]. We illustrate with the example in Figure 48.4.3, consisting of three rigid bodies and six bars in the plane. We may interpret the word "plane" here as "real projective plane."

Hence $V = \mathbb{R}^3$, and we let $W = \Lambda^2(V) \cong V^* \cong \mathbb{R}^3$. We think of the endpoints of the bars as elements of V, and hence the lines determined by the bars are two-extensors of these points, or elements of W. The algorithm produces the pure condition $[abc][def] - [abd][cef]$. This bracket polynomial may be Cayley factored as $ab \wedge cd \wedge ef$, as seen in Table 48.2.1. Now we switch to thinking of $a, b, \ldots, f$ as 2-extensors in V rather than elements of W, and recall that there is a duality between V and W, hence between $\Lambda(V)$ and $\Lambda(W)$. Thus, the framework has a first-order flex if and only if $(a \wedge b) \vee (c \wedge d) \vee (e \wedge f) = 0$ in $\Lambda(V)$. Hence the desired geometric condition for the existence of a first-order flex is that the three points $\overline{a \wedge b}$, $\overline{c \wedge d}$, and $\overline{e \wedge f}$ are collinear. Now $\overline{a \wedge b}$ is just the center of relative (instantaneous) motion for the two bodies connected by those two bars: think of fixing one of the bodies and then rotating the other body about this center; the lengths of the two bars are instantaneously preserved. The geometric result we have obtained is just a restatement of the classical theorem of Arnhold-Kempe that in

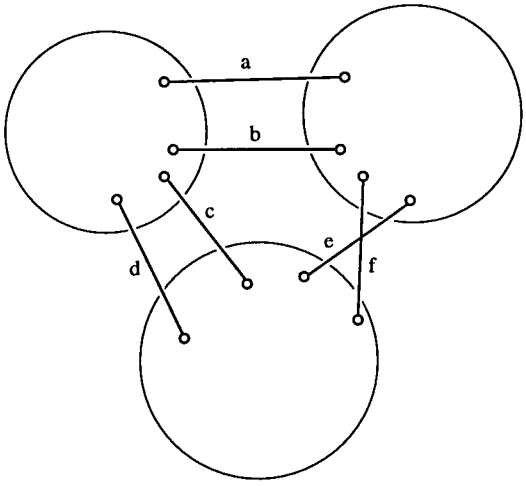


FIGURE 48.4.3
A bar-and-body framework.

any flex of three rigid bodies, the centers of relative motion of the three pairs of bodies must be collinear.

48.4.4 AUTOMATED GEOMETRIC THEOREM-PROVING

J. Richter-Gebert [RG95] uses Grassmann-Cayley algebra to derive bracket conditions for projective geometric incidences in order to produce coordinate-free automatic proofs of theorems in projective geometry. By introducing two circular points at infinity, the same can be done for theorems in Euclidean geometry [CRG95].

Richter-Gebert's technique is to reduce each hypothesis to a *binomial* equation, that is, an equation with a single product of brackets on each side. For example, as we have seen, the concurrence of three lines $\overline{ab}, \overline{cd}, \overline{ef}$ may be rewritten as $[acd][bef] = [bcd][aef]$. Similarly, the collinearity of three points $\overline{a}, \overline{b}, \overline{c}$ may be expressed as $[abd][bce] = [abe][bcd]$, avoiding the much more obvious expression $[abc] = 0$ since it is not of the required form. If all binomial equations are now multiplied together, and provided they were appropriately chosen in the first place, common factors may be canceled (which involves nondegeneracy assumptions, so that the common factors are nonzero), resulting in the desired conclusion. A surprising array of theorems may be cast in this format, and this approach has been successfully implemented.

48.5 SOURCES AND RELATED MATERIAL

SURVEYS

[DRS74] and [BBR85]: These two papers survey the properties of the Grassmann-Cayley algebra (called the "double algebra" in [BBR85]).

[Whi95]: A more elementary survey than the two above.

[Whi94]: Emphasizes the concrete approach via Plücker coordinates, and gives more detail on the connections to robotics.

RELATED CHAPTERS

Chapter 41: Robotics
Chapter 49: Rigidity and scene analysis

REFERENCES

[BBR85] M. Barnabei, A. Brini, and G.-C. Rota. On the exterior calculus of invariant theory. *J. Algebra*, 96:120–160, 1985.

[CRG95] H. Crapo and J. Richter-Gebert. Automatic proving of geometric theorems. In N. White, editor, *Invariant Methods in Discrete and Computational Geometry*, pages 167–196. Kluwer, Dordrecht, 1995.

[DRS74] P. Doubilet, G.-C. Rota, and J. Stein. On the foundations of combinatorial theory: IX, combinatorial methods in invariant theory. *Stud. Appl. Math.*, 53:185–216, 1974.

[MW91] T. McMillan and N. White. The dotted straightening algorithm. *J. Symbolic Comput.*, 11:471–482, 1991.

[RG95] J. Richter-Gebert. Mechanical theorem proving in projective geometry. *Ann. Math. Artif. Intell.*, 13:139–172, 1995.

[RS76] G.-C. Rota and J. Stein. Applications of Cayley algebras. In *Colloquio Internazionale sulle Teorie Combinatorie*, pages 71–97. Accademia Nazionale dei Lincei, 1976.

[Stu93] B. Sturmfels. *Algorithms in Invariant Theory.* Springer-Verlag, New York, 1993.

[Whi91] N. White. Multilinear Cayley factorization. *J. Symbolic Comput.*, 11:421–438, 1991.

[Whi94] N. White. Grassmann-Cayley algebra and robotics. *J. Intell. Robot. Syst.*, 11:91–107, 1994.

[Whi95] N. White. A tutorial on Grassmann-Cayley algebra. In N. White, editor, *Invariant Methods in Discrete and Computational Geometry*, pages 93–106. Kluwer, Dordrecht, 1995.

[WW83] N. White and W. Whiteley. The algebraic geometry of stresses in frameworks. *SIAM J. Algebraic Discrete Meth.*, 4:481–511, 1983.

[WW87] N. White and W. Whiteley. The algebraic geometry of motions in bar-and-body frameworks. *SIAM J. Algebraic Discrete Meth.*, 8:1–32, 1987.

49 RIGIDITY AND SCENE ANALYSIS

Walter Whiteley

INTRODUCTION

Rigidity and flexibility of frameworks (motions preserving lengths of bars) and scene analysis (liftings from plane polyhedral pictures to spatial polyhedra) are two core examples of a general class of geometric problems:

(a) Given a discrete configuration of points, lines, planes, ... in Euclidean space, and a set of geometric constraints (fixed lengths for rigidity, fixed incidences, and fixed projections of points for scene analysis), what is the set of solutions and what is its local form: discrete? k-dimensional?

(b) Given a structure satisfying the constraints, is it unique, or at least locally unique, up to trivial changes, such as congruences for rigidity?

(c) How does this answer depend on the combinatorics of the structure and how does it depend on the specific geometry of the initial data or object?

The rigidity of frameworks examines points constrained by fixed distances between pairs, using vocabulary and techniques drawn from structural engineering: bars and joints, first-order flexes, static self-stresses, and minimum energy (Section 49.1). Scene analysis and the dual concept of parallel drawings are described in Section 49.2. Finally, reciprocal diagrams form a fundamental geometric connection between liftings of polyhedral pictures and self-stresses in frameworks (Section 49.3). A number of other geometric problems with related mathematical and algorithmic patterns are mentioned in Sections 49.1.4, 49.2.3, and 49.3. For more general geometric reconstruction problems, see Chapter 26.

49.1 RIGIDITY OF BAR FRAMEWORKS

Given a set of points in space, with certain distances to be preserved, what other configurations have the same distances? If we make small changes in the distances, will there be a small (linear scale) change in the position? What is the structure, locally and globally, of the algebraic variety of these "realizations"?

We concentrate on three concepts of "rigidity": rigidity, first-order rigidity, and generic rigidity. First-order rigidity is the linearized (and therefore linear algebra) version of rigidity, while generic rigidity refers to what happens for "almost all" geometric positions of the underlying combinatorial structure. After the initial results relating these three concepts (Section 49.1.1), the study divides into the combinatorics of generic rigidity, using graphs (Section 49.1.2); the geometry of special positions in first-order rigidity, using projective geometry (Section 49.1.3); and extensions to tensegrity frameworks, using geometric results and minima of energy functions for rigidity (Section 49.1.4).

49.1.1 BASIC CONCEPTS OF RIGIDITY

GLOSSARY

Configuration of points in d-space: An assignment $p = (p_1, \ldots, p_v)$ of points $p_i \in \mathbb{R}^d$ to an index set V, where $v = |V|$.

Congruent configurations: Two configurations p and q in d-space, on the same set V, related by an isometry T of $\mathbb{R}^d$ (with $T(p_i) = q_i$ for all $i \in V$).

Bar framework in d-space $G(p)$ (or ***framework***): A graph $G = (V; E)$ (no loops or multiple edges) and a configuration p in d-space for the vertices V (Figure 49.1.1A).

Bar: An edge $\{i, j\} \in E$ for a framework $G(p)$.

Bar equivalence: Two frameworks $G(p)$ and $G(q)$ such that all bars have the same length in both configurations: $|p_i - p_j| = |q_i - q_j|$ for all bars $\{i, j\} \in E$.

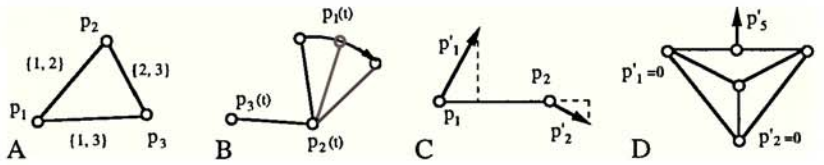
Analytic flex: An analytic function $p(t) : [0, 1) \to \mathbb{R}^{vd}$ such that $G(p(0))$ is bar equivalent to $G(p(t))$ for all t (i.e., all bars have constant length).

Flexible framework: A bar framework $G(p)$ in $\mathbb{R}^d$ with an analytic flex $p(t)$ such that $p(0) = p$ but p is not congruent to $p(t)$ for all $0 < t$ (Figure 49.1.1B).

Rigid framework: A bar framework $G(p)$ in d-space that is not flexible (Figure 49.1.1A,D).

Generically rigid graph in d-space: A graph G for which the frameworks $G(p)$ are rigid on an open dense subset of configurations p in d-space (Figures 49.1.1A, 49.1.2A).

FIGURE 49.1.1



First-order flex or ***infinitesimal motion:*** For a bar framework $G(p)$, an assignment of velocities $p' : V \to \mathbb{R}^d$, such that for each edge $\{i, j\} \in E$: $(p_j - p_i) \cdot (p'_j - p'_i) = 0$ (Figure 49.1.1.C,D, where the arrows represent nonzero velocities).

Trivial first-order flex: A first-order flex p' that is the derivative of a flex of congruent frameworks (Figure 49.1.1C). (There is a fixed skew-symmetric matrix S (a rotation) and a fixed vector t (a translation) such that, for all vertices $i \in V$, $p'_i = p_i S + t$.)

First-order flexible framework: A framework $G(p)$ with a nontrivial first-order flex (Figure 49.1.1D).

First-order rigid framework: A bar framework $G(p)$ for which every first-order flex is trivial (Figures 49.1.1A, 49.1.2A).

Rigidity matrix: For a framework $G(p)$ in d-space, $R_G(p)$ is the $|E| \times d|V|$ matrix for the system of equations: $(p_j - p_i) \cdot (p'_j - p'_i) = 0$ in the unknown velocities

p'_i. The first-order flex equations are expressed as

$$R_G(p){p'}^T = \begin{bmatrix} \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ 0 & \cdots & (p_i - p_j) & \cdots & (p_j - p_i) & \cdots & 0 \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \end{bmatrix} \times {p'}^T = 0^T.$$

Self-stress: For a framework $G(p)$, a row dependence ω for the rigidity matrix: $\omega R_G(p) = 0$. Equivalently, an assignment of scalars ω_{ij} to the edges such that at each vertex i, $\sum_{\{j|\ \{i,j\}\in E\ \}} \omega_{ij}(p_j - p_i) = 0$ (placing these "internal forces" $\omega_{ij}(p_j - p_i)$ in equilibrium at vertex i).

Independent framework: A bar framework $G(p)$ for which the rigidity matrix has independent rows.

Isostatic framework: A framework $G(p)$ that is first-order rigid and independent.

BASIC CONNECTIONS

Because the constraints $|p_i - p_j| = |q_i - q_j|$ are algebraic in the coordinates of the points (after squaring), many alternate definitions of a "rigid framework" are equivalent.

THEOREM 49.1.1 *Alternate Rigidity Definitions*

For a bar framework $G(p)$ the following conditions are equivalent:

(a) the framework is rigid;

(b) for every continuous path, or **continuous flex** *of $G(p)$, $p(t) \in \mathbb{R}^{vd}$, $0 \le t < 1$ and $p(0) = p$, such that $G(p(t))$ is bar equivalent to $G(p)$ for all t, $p(t)$ is congruent to p for all t;*

(c) there is an $\epsilon > 0$ such that if $G(p)$ and $G(q)$ are bar equivalent and $|p-q| < \epsilon$, then p is congruent to q.

The dimension of the space of trivial first-order motions of d-space is $\binom{d+1}{2}$ provided $|V| \ge d$ (the velocities generated by d translations and by $\binom{d}{2}$ rotations form a basis).

THEOREM 49.1.2 *First-order Rank*

A framework $G(p)$ with $|V| \ge d$ is first-order rigid if and only if the rigidity matrix $R_G(p)$ has rank $d|V| - \binom{d+1}{2}$.

A framework $G(p)$ with few vertices, $|V| \le d$, is isostatic if and only if the rigidity matrix $R_G(p)$ has rank $\binom{v}{2}$ (if and only if G is the complete graph on V and the points p_i do not lie in an affine space of dimension $|V| - 2$).

First-order rigidity is linear algebra, with first-order rigid frameworks, self-stresses, and isostatic frameworks playing the roles of spanning sets, linear dependence, and bases of the row space for the rigidity matrix of the complete graph on the configuration p.

THEOREM 49.1.3 *Isostatic Frameworks*

For a framework $G(p)$ in d-space, with $|V| \geq d$, the following are equivalent:

(a) $G(p)$ is isostatic (first-order rigid and independent);

(b) $G(p)$ is first-order rigid with $|E| = d|V| - \binom{d+1}{2}$;

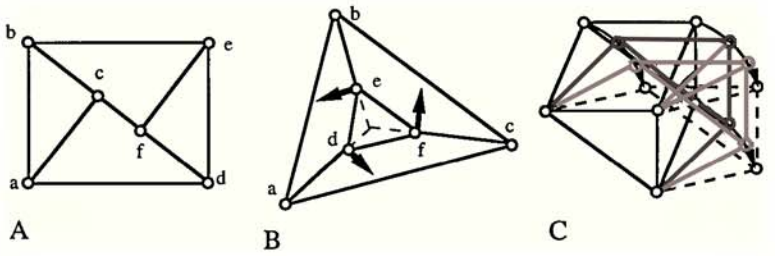
(c) $G(p)$ is independent with $|E| = d|V| - \binom{d+1}{2}$;

(d) $G(p)$ is first-order rigid, and removing any one bar (but no vertices) leaves a first-order flexible framework.

First-order rigidity of a framework $G(p)$ is a robust property: a small change in the configuration p preserves this rigidity. Independence implies that the distances are robust: any small change in these distances can be realized by a nearby configuration. On the other hand, self-stresses mean that one of the distances is algebraically dependent on the others: some small changes in the distances may have no realizations, while other small changes may result in relatively large changes in the configuration.

Figure 49.1.2 illustrates a single graph with plane configurations that produce: (A) a first-order rigid framework; (B) a first-order flexible, but rigid, framework, and (C) a flexible framework. The graph itself is generically 2-rigid.

FIGURE 49.1.2



Essentially, the first derivative of a nontrivial analytic flex is a nontrivial first-order flex: $D_t\big((p_i(t) - p_j(t))^2 = c_{ij}\big)\big|_{t=0} \Rightarrow 2(p_j - p_i) \cdot (p'_j - p'_i) = 0$. (If this first derivative is trivial, then the earliest nontrivial derivative is a first-order motion.)

THEOREM 49.1.4 *First-order Rigid to Rigid*

If a bar framework $G(p)$ is first-order rigid, then $G(p)$ is rigid.

Some first-order flexes are not the derivatives of analytic flexes (Figures 49.1.1D and 49.1.2B). A nontrivial first-order flex for a framework does guarantee a pair of nearby noncongruent, bar equivalent frameworks.

THEOREM 49.1.5 *Averaging Theorem*

If the points of a configuration p affinely span d-space, then the assignment p' is a nontrivial first-order flex of $G(p)$ if and only if the frameworks $G(p + p')$ and $G(p - p')$ are bar equivalent and not congruent.

Rigidity and first-order rigidity are equivalent in some situations.

THEOREM 49.1.6 *Rigid to First-order Rigid*

If bar framework $G(p)$ is independent, then $G(p)$ is first-order rigid if and only if $G(p)$ is rigid.

THEOREM 49.1.7 *Generic Rigidity Theorem*

For a graph G and a fixed dimension d the following are equivalent:

(a) G is generically rigid in d-space;

(b) for all $q \in U \subset \mathbb{R}^{dv}$, U some nonempty open set, the frameworks $G(q)$ are rigid;

(c) for all $q \in U \subset \mathbb{R}^{dv}$, U some open dense set, $G(q)$ is first-order rigid (and rigid);

(d) for each configuration p using algebraically independent numbers over the rationals as coordinates, the framework $G(p)$ is first-order rigid;

(e) $G(p)$ is first-order rigid for some configuration $p \in \mathbb{R}^{dv}$.

49.1.2 COMBINATORICS FOR GENERIC RIGIDITY

The major goal in generic rigidity is a combinatorial characterization of graphs that are generically rigid in d-space. The companion problem is to find efficient combinatorial algorithms to test graphs for generic rigidity. For the plane (and the line), this is solved. Beyond the plane the results are essentially incomplete, but some partial results are available.

GLOSSARY

Generically d-independent: A graph G for which some (equivalently, almost all) configurations p produce independent frameworks in d-space.

Generically d-isostatic graph: A graph G for which some (equivalently, almost all) configurations p produce isostatic frameworks in d-space.

Complete bipartite graph: A graph $K_{m,n} = (A \cup B, A \times B)$, where A and B are disjoint sets of cardinality $|A| = m$ and $|B| = n$.

Triangulated d-pseudomanifold: A finite set of d-simplices (complete graphs on $d+1$ points) with the property that each d subset (facet) occurs in exactly two simplices, any two simplices are connected by a path of simplices and shared facets, and any two simplices sharing a vertex are connected through other simplices at this vertex. (For example, the triangles, edges, and vertices of a closed triangulated 2-surface without boundary, such as a sphere or torus, form a 2-pseudomanifold.) Cf. Section 15.3.

Henneberg d-construction for a graph G: A sequence $(V_d, E_d), ..., (V_n, E_n)$ of graphs, such that:

(i) For each index $d < j \leq n$, (V_j, E_j) is obtained from (V_{j-1}, E_{j-1}) by

vertex addition: attaching a new vertex by d edges (Figure 49.1.4A for $d = 2$), or

edge splitting: replacing an edge from (V_{j-1}, E_{j-1}) with a new vertex joined to its ends and to $d-1$ other vertices (Figure 49.1.4B for $d=2$); and

(ii) (V_d, E_d) is the complete graph on d vertices, and $(V_n, E_n) = G$ (Figure 49.1.6A).

Proper 3Tree2 partition: A partition of the edges of a graph into three trees, such that each vertex is attached to exactly two of these trees and no nontrivial subtrees of distinct trees T_i have the same support (i.e., the same vertices) (Figure 49.1.6B).

d-connected graph: A graph G such that removing any $d-1$ vertices (and all incident edges) leaves a connected graph. (Equivalently, a graph such that any two vertices can be connected by at least d paths that are disjoint except for their endpoints.)

BASIC PROPERTIES IN ALL DIMENSIONS

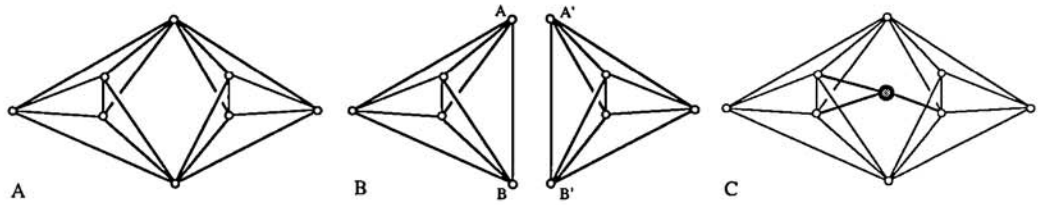
THEOREM 49.1.8 *Necessary Counts Theorem*

If a graph G is generically d-isostatic, then, for every subgraph on $|V'| \geq d$ vertices with edges E' in $V' \times V'$, $|E'| \leq d|V'| - \binom{d+1}{2}$.

THEOREM 49.1.9 *Necessary Connectivity Theorem*

If $G = (V, E)$ is a generically d-isostatic graph with $|V| > d$, then (V, E) is a d-connected graph.

FIGURE 49.1.3



For dimensions $d > 2$, these two conditions are not enough to characterize the graphs. Figure 49.1.3A shows a generically flexible counterexample for the sufficiency of the counts in dimension 3. This example is generated by a "circuit exchange" on two over-counted graphs (Figure 49.1.3B). Figure 49.1.3C adds 3-connectivity, but preserves the flexibility and the counts. For dimensions 1 and 2, the first count alone is sufficient for generic rigidity (see below).

THEOREM 49.1.10 *Bipartite Graphs*

A complete bipartite graph $K_{m,n}$, with $m > 1$, is generically rigid in dimension d if and only if $m + n \geq \binom{d+2}{2}$ and $m, n > d$.

INDUCTIVE CONSTRUCTIONS FOR ISOSTATIC GRAPHS

Inductive constructions for graphs that preserve generic rigidity are used both to prove theorems for general classes of frameworks and to analyze particular graphs.

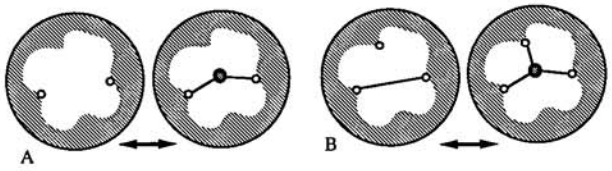
THEOREM 49.1.11 *Vertex Addition Theorem*

Let $G = (V, E)$ be a graph with a vertex i of valence d; let $H = (U, F)$ denote the subgraph obtained by deleting i and the edges incident with it. Then G is generically d-isostatic if and only if H is generically d-isostatic (Figure 49.1.4A for $d = 2$).

THEOREM 49.1.12 *Edge Splits Theorem*

Let $G = (V, E)$ be a graph with a vertex i of valence $d+1$, let S be the set of vertices adjacent to i, and let $H = (U, F)$ be the subgraph obtained by deleting i and its $d+1$ incident edges. Then G is generically d-isostatic if and only if there is a pair j, k of vertices of V such that the edge $\{j, k\}$ is not in F and the graph $H' = (U, F \cup \{j, k\})$ is generically d-isostatic (Figure 49.1.4B for $d = 2$).

FIGURE 49.1.4



THEOREM 49.1.13 *Construction Theorem*

If a graph G is obtained by a Henneberg d-construction, then G is generically d-isostatic (Figure 49.1.6A for $d = 2$).

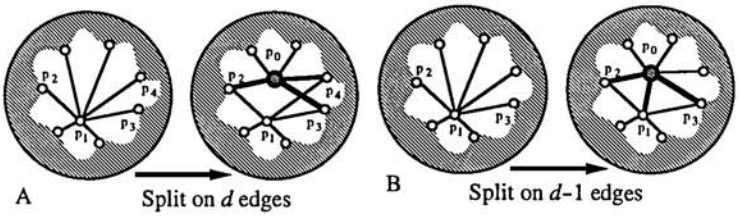
THEOREM 49.1.14 *Gluing Lemma*

If $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are generically d-rigid graphs sharing at least d vertices, then $G = (V_1 \cup V_2, E_1 \cup E_2)$ is generically d-rigid.

THEOREM 49.1.15 *Vertex Splitting Theorem*

If the graph G' is a vertex split of a generically d-rigid graph G on d edges (Figure 49.1.5A for $d = 3$) *or a vertex split on $d - 1$ edges* (Figure 49.1.5B for $d = 3$), *then G' is generically d-rigid.*

FIGURE 49.1.5



PLANE ISOSTATIC GRAPHS

Many plane results are expressed in terms of trees in the graph, due to a simple correspondence between rigidity on the line and the connectivity of the graph.

THEOREM 49.1.16 *Line Rigidity*

For graph G and configuration p on the line with $p_i \neq p_j$ for all $\{i,j\} \in E$, the following are equivalent:

(a) $G(p)$ is minimal among rigid frameworks on the line with these vertices;

(b) $G(p)$ is isostatic on the line;

(c) G is a spanning tree on the vertices;

(d) $|E| = |V| - 1$ and for every nonempty subset E' with vertices V', $|E'| \leq |V'| - 1$.

THEOREM 49.1.17 *Plane Isostatic Graphs Theorem*

For a graph G with $|V| \geq 2$, the following are equivalent:

(a) G is generically isostatic in the plane;

(b) $|E| = 2|V| - 3$, and for every subgraph (V', E') with $|V'| \geq 2$ vertices, $|E'| \leq 2|V'| - 3$ (Laman's Theorem);

(c) there is a Henneberg 2-construction (V, E) for G (Henneberg's Theorem);

(d) E has a proper 3Tree2 partition (Crapo's Theorem);

(e) for each $\{i,j\} \in E$, the multigraph obtained by doubling the edge $\{i,j\}$ is the union of two spanning trees (Recski's Theorem).

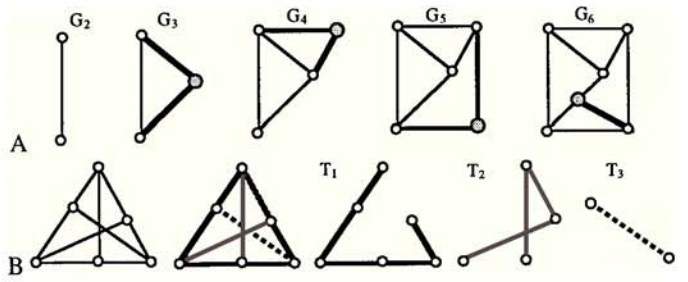


FIGURE 49.1.6

Figure 49.1.6A shows the Henneberg plane construction for the isostatic graph of Figure 49.1.2. Figure 49.1.6B shows a proper 3Tree2 partition of the edges of the isostatic complete bipartite graph $K_{3,3}$. With an added edge, joining T_2 to T_3, this partition creates several of the pairs of spanning trees predicted by Recski's Theorem.

THEOREM 49.1.18 *Sufficient Connectivity*

If a graph G is 6-connected, then G is generically rigid in the plane.

There are examples of 5-connected graphs that are not generically rigid in the plane.

ALGORITHMS FOR GENERIC 2-RIGIDITY

Each of the combinatorial characterizations has an associated algorithm for verifying whether a graph is generically 2-isostatic:

(i) Counts: This can be checked by an $O(|V|^2)$ algorithm based on bipartite matchings on an associated graph [Sug86].

(ii) 2-construction: Existence of a 2-construction can be checked by an $O(2^{|V|})$ algorithm, but a proposed 2-construction can be verified correct in $O(|V|)$ time.

(iii) 3Tree2 covering: Existence can be checked by an $O(|V|^2)$ matroid algorithm [Cra].

(iv) Double tree partition: All required double-tree partitions can be found by a matroidal algorithm of order $O(|V|^3)$.

ISOSTATIC GRAPHS IN HIGHER DIMENSIONS

Most of the results are covered by the initial summary for all dimensions d. Special results apply to the graphs of convex polytopes, as well as more general surfaces.

THEOREM 49.1.19 *Triangulated Pseudomanifolds Theorem*

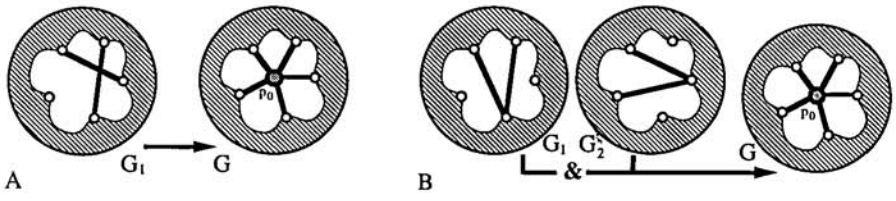
For $d \geq 2$, the graph of any triangulated d-pseudomanifold is generically $(d+1)$-rigid.

In particular, the graph of any closed triangulated 2-surface without boundary is generically rigid in 3-space (Fogelsanger's Theorem), and the graph of any triangulated sphere is generically 3-isostatic (Glück's Theorem).

OPEN PROBLEMS

There is no combinatorial characterization of generically 3-isostatic graphs. There are several related conjectures, due to Dress, Graver, and Tay and Whiteley, which may be correct but are unproven.

FIGURE 49.1.7



CONJECTURE 49.1.20 *3-D Replacement Conjecture*

The X-replacement in Figure 49.1.7A *takes a graph G_1 that is generically rigid in 3-space to a graph G that is generically rigid in 3-space.*

The double V-replacement in Figure 49.1.7B *takes two graphs* G_1, G_2 *that are generically rigid in* 3*-space to a graph* G *that is generically rigid in* 3*-space.*

Every 3-isostatic graph is generated by an "extended Henneberg 3-construction," which adds these two moves to the simpler edge splitting and vertex addition. What is unproven is that *only* 3-isostatic graphs are generated in this way.

The plane analogue of X-replacement is true for plane generic rigidity (without adding the fifth bar), and the 4-space analogue is false for some graphs in 4-space (with two extra bars added). If these conjectured steps prove correct in 3-space, then we would have inductive techniques to generate the graphs of all isostatic frameworks in 3-space.

A graph can be checked for generic 3-rigidity by a "brute force" $O(2^{2^{|V|}})$ algorithm. Assign the points independent variables as coordinates, form the rigidity matrix, then check the rank by symbolic computation. On the other hand, if numerical coordinates are chosen for the points "at random," then the rank of this numerical matrix ($O(|E|^3)$) will be the generic value, with probability 1. This problem has a randomized polynomial-time algorithm, but there is no known deterministic algorithm that runs in polynomial time.

For 4-space, there is no conjecture that has held up against the known counterexamples (based on generically flexible complete bipartite graphs such as $K_{7,7}$).

CONJECTURE 49.1.21 *Sufficient Connectivity Conjecture*

If a graph G *is* 12*-connected, then* G *is generically rigid in* 3*-space.*

49.1.3 GEOMETRY OF FIRST-ORDER RIGIDITY

GLOSSARY

Special position of a graph G in d-space: Any configuration p in d-space such that the rigidity matrix $R_G(p)$, or any submatrix, has rank smaller than the generic rank (the rank at a configuration with algebraically independent coordinates).

Projective transform of a d-configuration p: A d-configuration q on the same vertices, such that there is an invertible matrix T of size $d+1 \times d+1$ making $T(p_i, 1) = \lambda_i(q_i, 1)$ (where $(p_i, 1)$ is the vector p_i extended with an additional 1—the affine coordinates of p_i).

Affine spanning set for d-space: A configuration p of points such that every point $q_0 \in \mathbb{R}^d$ can be expressed as an affine combination of the p_i: $q_0 = \sum_i \lambda_i p_i$, with $\sum_i \lambda_i = 1$. (Equivalently, the affine coordinates $(p_i, 1)$ span the vector space $\mathbb{R}^{d+1}$.)

Cone graph: The graph $G * u$ obtained from $G = (V, E)$ by adding a new vertex u and the $|V|$ edges (u, i) for all vertices $i \in V$.

Cone projection from p_0: For a $(d+1)$-configuration p on V, a configuration $q = \Pi_0(p)$ in d-space (placed as a hyperplane in $(d+1)$-space) on the vertices $V \backslash 0$, such that $p_i \neq p_0$ is on the line $q_i p_0$ for all $i \neq 0$.

BASIC RESULTS

THEOREM 49.1.22 *First-order Flex Test*

If the points of a configuration p on the vertices V affinely span d-space, then a first-order motion p' is nontrivial if and only if there is some pair h, k (not a bar) such that: $(p_h - p_k) \cdot (p'_h - p'_k) \neq 0$.

THEOREM 49.1.23 *Projective Invariance*

If a framework $G(p)$ is first-order rigid (isostatic, independent) and $q = T(p)$ is a projective transform of p, then $G(q)$ is first-order rigid (isostatic, independent, respectively).

Rigidity itself is not projectively invariant—or even affinely invariant. It is a purely Euclidean property. The following property is closely related.

THEOREM 49.1.24 *Coning Theorem*

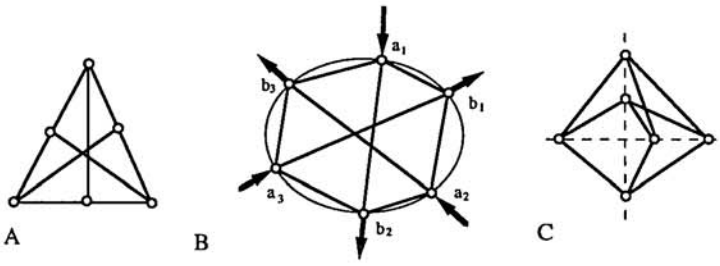
A framework $G(\Pi_0 p)$ is first-order rigid (isostatic, independent) in d-space if and only if the cone $(G * u)(p)$ is first-order rigid (isostatic, independent, respectively) in $(d+1)$-space.

The special positions of a graph in d-space are rare, since they form a proper algebraic variety (essentially generated by minors of the rigidity matrix with variables for the coordinates of points). For a generically isostatic graph, this set of special positions can be described by the zeros of a single polynomial.

THEOREM 49.1.25 *Pure Condition*

For any graph G that is generically isostatic in d-space, there is a homogeneous polynomial $C_G(x_{1,1}, \ldots, x_{1,d}, \ldots, x_{|V|,1}, \ldots, x_{|V|,d})$ such that $G(p)$ is first-order flexible if and only if $C_G(p_1, \ldots, p_{|V|}) = 0$. C_G is of degree $(\mathrm{val}_i + 1 - d)$ in the variables $(x_{i,1}, \ldots, x_{i,d})$ for each vertex i of valence val_i in the graph.

FIGURE 49.1.8



Since Grassmann algebra (Chapter 48) is the appropriate language for projective properties, these pure conditions C_G are polynomials in the Grassmann algebra. Section 48.4 contains several examples of these polynomial conditions.

THEOREM 49.1.26 *Quadratics for Bipartite Graphs*

For a complete bipartite graph $K_{m,n}$ and $d > 1$, the framework $K_{m,n}(p)$, with $p(A)$ and $p(B)$ each affinely spanning d-space, is first-order flexible if and only if all the points $p(A \cup B)$ lie on a quadric surface of d-space (Figure 49.1.8).

The following classical result describes an important open set of configurations that are not special for triangulated spheres.

THEOREM 49.1.27 *Extended Cauchy Theorem*

If $G(p)$ consists of the vertices and edges of a convex simplicial d-polytope, then $G(p)$ is first-order rigid in d-space.

Connelly [Con82] gives a nonconvex (but not self-intersecting) triangulated sphere (with nine vertices) that is flexible. For many graphs, such as a triangulated torus (Theorem 49.1.19), we do not have even one specific configuration that gives a first-order rigid framework, only the guarantee that "almost all" configurations will work.

OTHER RELATED STRUCTURES

A number of related structures have also been investigated for first-order rigidity. One, which appears in engineering, robotics, and chemistry, is the "body and hinge framework." Rigid bodies, indexed by V, are connected in pairs along hinges (lines in 3-space), indexed by edges of a graph. The bodies each move, preserving the contacts at the hinges. Such hinged frameworks could be modeled as bar and joint frameworks, with each hinge replaced by a pair of joints and each body replaced by a first-order rigid framework on the joints of its hinges (and other joints if desired); cf. Sections 41.1 and 48.4. A second "model" treats the atoms of a molecule as the bodies, and the lines of the bond lines as hinges. Unlike the unsolved problems for generic rigidity of frameworks in 3-space, the generic behavior of body and hinge structures has been completely solved. We state two sample results—one combinatorial, the other geometric.

THEOREM 49.1.28 *Tay's Theorem*

For a graph G the following are equivalent:

(a) for some hinge assignment of lines $h_{i,j}$ in 3-space to the edges $\{i,j\}$ of G, the body and hinge framework $G(h)$ is first-order rigid;

(b) for almost all hinge assignments h, the body and hinge framework $G(h)$ is first-order rigid;

(c) if each edge of the graph is replaced by five copies, the resulting multigraph contains six edge-disjoint spanning trees.

Tay's Theorem extends directly to all dimensions d (finding $\binom{d+1}{2}$ edge-disjoint spanning trees inside $\binom{d+1}{2} - 1$ copies of the graph).

THEOREM 49.1.29 *Spherical Flexes and Stresses*

Given an abstract spherical structure (see Section 49.3) *$S = (V, F; \underline{E})$, and an assignment of distinct points $p_i \in \mathbb{R}^3$ to the vertices, the following two conditions are equivalent:*

(a) the bar framework $G(p)$ on $G = (V, E)$ has a nontrivial self-stress;

(b) the body and hinge framework on the dual graph $G^* = (F, E^*)$ with hinge lines $p_i p_j$ for each edge $\{i,j\}$ of G is first-order flexible.

In particular, any body and hinge framework on the faces and edges of a strictly convex spherical polyhedron is first-order rigid.

49.1.4 TENSEGRITY FRAMEWORKS

In a tensegrity framework, we replace some (or all) of the equalities for bars with inequalities for the distances—corresponding to *cables* (the distance can shrink but not expand) and *struts* (the distance can expand but not shrink). The study of these inequalities introduces techniques from linear programming.

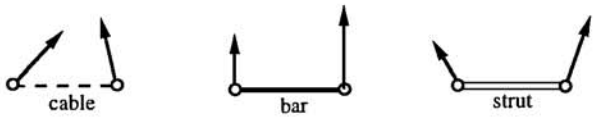
GLOSSARY

Signed graph: A graph with a partition of the edges into three classes, written $G_\pm = (V; E_-, E_0, E_+)$.

Tensegrity framework $G_\pm(p)$ **in** $\mathbb{R}^d$: A signed graph $G_\pm = (V; E_-, E_0, E_+)$ and a configuration p on V.

Cables, bars, struts: For a tensegrity framework $G_\pm(p)$, the members of E_-, of E_0, and of E_+, respectively. In figures, cables are indicated by dashed lines, struts by double thin lines, and bars by single thick lines (see Figure 49.1.9).

FIGURE 49.1.9



$G_\pm(p)$ ***dominates*** $G_\pm(q)$***:*** For each edge, the appropriate condition holds:

$$\begin{aligned} |p_i - p_j| &\geq |q_i - q_j| \quad \text{when} \quad \{i,j\} \in E_- \\ |p_i - p_j| &= |q_i - q_j| \quad \text{when} \quad \{i,j\} \in E_0 \\ |p_i - p_j| &\leq |q_i - q_j| \quad \text{when} \quad \{i,j\} \in E_+. \end{aligned}$$

Rigid tensegrity framework $G_\pm(p)$***:*** For every analytic path $p(t)$ in $\mathbb{R}^{vd}$, $0 \leq t < 1$, if $p(0) = p$ and $G(p)$ dominates $G(p(t))$ for all t, then p is congruent to $p(t)$ for all t.

First-order flex of a tensegrity framework $G_\pm$: An assignment $p' : V \to \mathbb{R}^d$ of velocities to the vertices such that, for each edge $\{i,j\} \in E$ (Figure 49.1.9),

$$\begin{aligned} (p_j - p_i) \cdot (p'_j - p'_i) &\leq 0 \quad \text{for cables} \quad \{i,j\} \in E_- \\ (p_j - p_i) \cdot (p'_j - p'_i) &= 0 \quad \text{for bars} \quad \{i,j\} \in E_0 \\ (p_j - p_i) \cdot (p'_j - p'_i) &\geq 0 \quad \text{for struts} \quad \{i,j\} \in E_+. \end{aligned}$$

Trivial first-order flex: A first-order flex p' of a tensegrity framework $G_\pm(p)$ such that $p'_i = Sp_i + t$ for all vertices i, with a fixed skew-symmetric matrix S and vector t.

First-order rigid: A tensegrity framework $G_\pm(p)$ is first-order rigid if every first-order flex is trivial, and ***first-order flexible*** otherwise.

Proper self-stress on a tensegrity framework $G_\pm(p)$: An assignment ω of scalars to the edges of G such that:

(a) $\omega_{ij} \geq 0$ for cables $\{i,j\} \in E_-$;

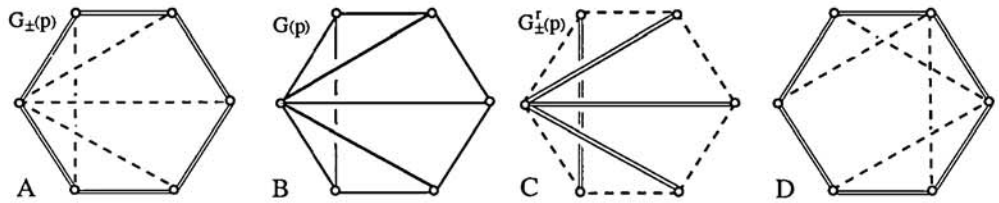
(b) $\omega_{ij} \leq 0$ for struts $\{i,j\} \in E_+$; and

(c) for each vertex i, $\sum_{\{j \mid \{i,j\} \in E\}} \omega_{ij}(p_j - p_i) = 0$.

Strict self-stress: A proper self-stress ω with the inequalities in (a) and (b) strict.

Underlying bar framework: For a tensegrity framework $G_\pm(p)$, the bar framework $G(p)$ on the unsigned graph $G = (V, E)$, where $E = E_- \cup E_0 \cup E_-$ (Figure 49.1.10A,B).

FIGURE 49.1.10



BASIC RESULTS

The equivalent definitions of "rigidity" and the basic connections between rigidity and first-order rigidity all transfer directly to tensegrity frameworks [RW81].

THEOREM 49.1.30 *First-Order Stress Test*

A tensegrity framework $G_\pm(p)$ is first-order rigid if and only if the underlying bar framework $G(p)$ is first-order rigid and there is a strict self-stress on $G_\pm(p)$ (Figure 49.1.10A,B).

This connection to self-stresses means that any first-order rigid tensegrity framework with at least one cable or strut has $|E| > d|V| - \binom{d+1}{2}$ edges.

THEOREM 49.1.31 *Reversal Theorem*

A tensegrity framework $G_\pm(p)$ is first-order rigid if and only if the reversed framework $G_\pm^r(p)$ is first-order rigid, where the graph $G_\pm^r$ interchanges cables and struts (Figure 49.1.10A,C).

There is no single "generic" behavior for a signed graph $G_\pm$. If some configuration produces a first-order rigid framework for a graph $G_\pm$, then the set of all such configurations is open but not dense. The algebraic variety of "special positions" of the underlying unsigned graph divides the configuration space into open subsets, in some of which all configurations are rigid, and in others, none are.

The first-order rigidity of a tensegrity framework is projectively invariant, with the proviso that a cable (strut) $\{i, j\}$ is switched to a strut (cable) whenever $\lambda_i \lambda_j < 0$ for the projective transformation.

THEOREM 49.1.32 *Stress Existence*

If a tensegrity framework $G_{\pm}(p)$ with at least one cable or strut is rigid, then there is a nonzero proper self-stress.

A number of results relate minima of quadratic energy functions to the rigidity of tensegrity frameworks. These energy results are not invariant under projective transformations, but such rigidity is preserved under "small" affine transformations. This is one result, drawn from results on second-order rigidity [CoW96].

THEOREM 49.1.33 *Rigidity Stress Test*

A tensegrity framework $G_{\pm}(p)$ is rigid if, for each nontrivial first-order motion p' of $G_{\pm}(p)$, there is a proper self-stress $\omega^{p'}$ making $\sum_{ij} \omega_{ij}^{p'}(p'_i - p'_j) \cdot (p'_i - p'_j) > 0$.

A special result for modified frameworks—with some vertices fixed or pinned—further illustrates the role of tensegrity frameworks. A ***spider web*** is a partitioned graph $G_- = (V_0, V_1, E_-)$, with pinned vertices V_0, with $E_- \imath V_1 \times [V_0 \cup V_1]$ and a configuration p for $V_0 \cup V_1$. A ***spider web self-stress*** for $G_-(p)$ is an assignment ω of nonnegative scalars to E_- such that for each unpinned vertex $i \in V_1$, $\sum_{\{j|\{i,j\} \in E_-\}} \omega_{ij}(p_j - p_i) = 0$. A ***spider web flex*** for $G_-(p)$ is a flex $p(t)$ of the induced tensegrity framework on the spider web, with all pinned vertices fixed $(p_k(t) = p_k)$.

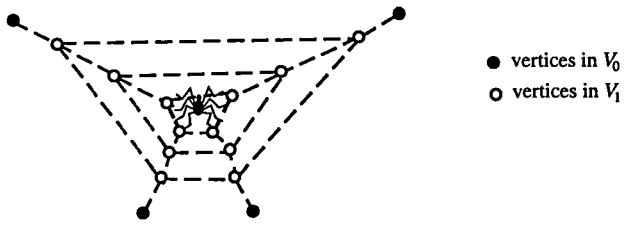


FIGURE 49.1.11

THEOREM 49.1.34 *Spider Web Rigidity*

Any spider web $G_-(p)$ in d-space with a spider web self-stress, positive on all cables, is rigid in d-space.

Some graph drawing programs (Chapter 44) use minima (or critical points) of quadratic energy functions on the edge lengths. All critical points of functions of squared edge-lengths correspond to proper self-stresses of a tensegrity framework, with members E_- for positive coefficients and E_+ for negative coefficients in the energy function. As a corollary, for planar graphs they generate pictures of polyhedra (Section 49.3).

Related to sphere packings (Chapter 50) are "reversed spider webs": tensegrity frameworks with vertices at the centers of the spheres (fixed joints for external pressures or constraints) and struts when two spheres contact. Such strut frameworks are rigid (corresponding to locally maximal density of the packing) if and only if they are first-order rigid (again with vertices in V_0 fixed) [Con88].

49.2 SCENE ANALYSIS

The problem of reconstructing spatial objects (polyhedra or polyhedral surfaces) from a single plane picture is basic to several applications. This section summarizes the combinatorial results for "generic pictures" (Section 49.2.1). This is followed by a polar "parallel configurations" interpretation of the same abstract mathematics (Section 49.2.2), and its connections to several fields of discrete geometry (Section 49.2.3).

49.2.1 COMBINATORICS OF PLANE POLYHEDRAL PICTURES

GLOSSARY

Polyhedral incidence structure S: An abstract set of ***vertices*** V, an abstract set of ***faces*** F and a set of ***incidences*** $I \subset V \times F$.

d-scene for an incidence structure $S = (V, F; I)$: A pair of location maps, $p : V \to \mathbb{R}^d$, $p_i = (x_i, \ldots, z_i, w_i)$ and $P : F \to \mathbb{R}^d$, $P^j = (A^j, \ldots, C^j, D^j)$, such that, for each $(i,j) \in I$: $A^j x_i + \ldots + C^j z_i + w_i + D^j = 0$. (We assume that no hyperplane is vertical, i.e., is parallel to the vector $(0, 0, \ldots, 0, 1)$.)

(d−1)-picture of an incidence structure S: A location map $r : V \to \mathbb{R}^{d-1}$, $r_i = (x_i, \ldots, z_i)$ (Figure 49.2.1A).

Lifting of a $(d-1)$-picture $S(r)$: A d-scene $S(p, P)$ with vertical projection $\Pi(p) = r$ (Figure 49.2.1B). (I.e., if $p_i = (x_i, \ldots, z_i, w_i)$, then $r_i = (x_i, \ldots, z_i) = \Pi(p_i)$).

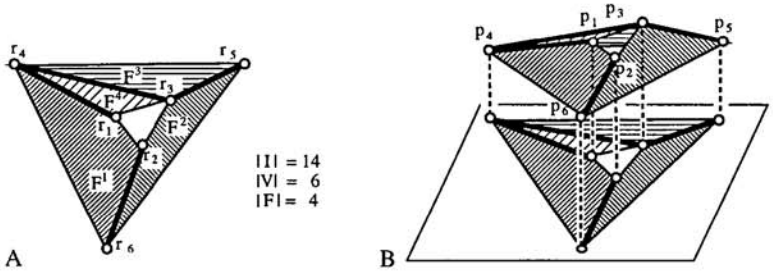


FIGURE 49.2.1

Lifting matrix for a picture $S(r)$: The $|I| \times (|V| + d|F|)$ coefficient matrix $M_S(r)$ of the system of equations for liftings of a picture $S(r)$: for each $(i,j) \in I$, $A^j x_i + \ldots + C^j z_i + w_i + D^j = 0$, where the variables are ordered:

$$\ldots, w_i, \ldots \; ; \; \ldots, A^j, \ldots, C^j, D^j, \ldots .$$

Sharp picture: A $(d-1)$-picture $S(r)$ that has a lifting $S(p, P)$ with a distinct hyperplane for each face (Figure 49.2.1A,B).

Since the incidence equations are linear, there is no distinction between "continuous liftings" and "first-order liftings." Since the rank of the lifting matrix is determined by a polynomial process on the entries, "generic properties" of pictures have several characterizations.

THEOREM 49.2.1 *Generic Pictures*

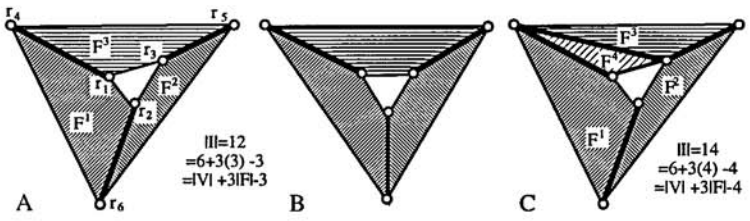
For a structure S and a dimension d, the following are equivalent:

(a) the structure is generically sharp in d-space (an open dense subset of configurations r produce sharp d-pictures);

(b) $S(r)$ is sharp for a configuration r with algebraically independent coordinates.

The generic properties of a structure are robust: all small changes in such a sharp picture are also sharp pictures and small changes in the points of a sharp picture require only small changes in the sharp lifting. Even special positions of such structures will always have nontrivial liftings, although these may not be sharp. However, up to numerical round-off, all pictures "are generic." Other structures that are not generically sharp (Figure 49.2.2A) may have sharp pictures in special positions (Figure 49.2.2B), but a small change in the position of even one point can destroy this sharpness. (Numerical round-off will typically produce such a small change in position.)

FIGURE 49.2.2



The incidence equations allow certain "trivial" changes to a lifted scene that will preserve the picture—generated by adding a single plane H^0 to all existing planes: $P_*^j = H^0 + P^j$; and by changes in vertical scale in the scene: $w_i^* = \lambda w_i$. This space of ***lifting equivalences*** has dimension $d+1$, provided the points of the scene do not lie in a single hyperplane.

THEOREM 49.2.2 *Picture Theorem*

A generic picture of an incidence structure $S = (V, F; I)$ with at least two faces has a sharp lifting, unique up to lifting equivalence, if and only if $|I| = |V| + d|F| - (d+1)$ and, for all subsets I' of incidences on at least two faces, $|I'| \leq |V'| + d|F'| - (d+1)$ (Figure 49.2.1A,C).

A generic picture of an incidence structure $S = (V, F; I)$ has independent rows in the lifting matrix if and only if for all nonempty subsets I' of incidences, $|I'| \leq |V'| + d|F'| - d$ (Figure 49.2.2A).

ALGORITHMS

Any part of a structure with $|I'| = |V'| + d|F'| - d$ independent incidences will be forced to be coplanar over a picture with algebraically independent coordinates for the points. If the structure is not generically sharp, then an effective, robust lifting algorithm consists of selecting a subset of vertices for which the incidences are sharp, then "correcting" the position of the other vertices based on calculations in the resulting scene. This requires effective algorithms for selecting such a set of incidences. Sugihara and Imai have implemented $O(|I|^2)$ algorithms for finding generically sharp (independent) structures using modified bipartite matching on the incidence structure [Sug86].

49.2.2 PARALLEL DRAWINGS

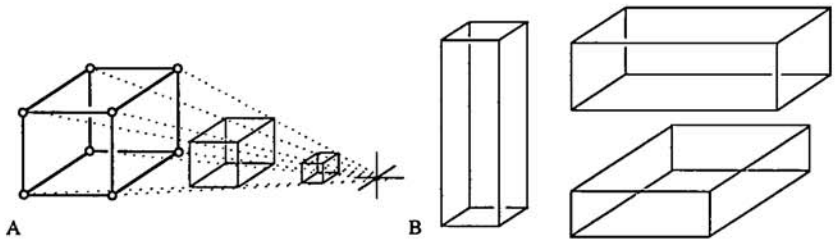
The mathematical structure defined for polyhedral pictures has another, dual interpretation: the polar of a "point constrained by one projection" is a "hyperplane constrained by an assigned normal." Two configurations sharing the prescribed normals are "parallel drawings" of one another. These geometric patterns, used by engineering draftsmen in the nineteenth century, have reappeared in a number of branches of discrete geometry. This dual interpretation also establishes a basic connection between the geometry and combinatorics of scene analysis and the geometry and combinatorics of first-order rigidity of frameworks.

GLOSSARY

Parallel d-scenes for an incidence structure: Two d-scenes $S(p, P)$, $S(q, Q)$ such that for each face j, $P^j || Q^j$ (that is, the first $d-1$ coordinates are equal) (Figure 49.2.3). (For convenience, not necessity, we stick with the "nonvertical" scenes of the previous section.)

Nontrivially parallel d-scene for a d-scene $S(p, P)$: A parallel d-scene $S(q, Q)$, such that the configuration q is not a translation or dilatation of the configuration p (Figure 49.2.3B for $d = 2$).

FIGURE 49.2.3



Directions for the faces: An assignment of d-vectors $D^j = (A^j, \ldots, C^j)$ to $j \in F$.

d-scene realizing directions D: A d-scene $S(p, P)$ such that for each face $j \in F$, the first $d-1$ coordinates of P^j and D^j coincide.

Parallel drawing matrix for directions D in d-space: The $|I| \times (|V| + d|F|)$

matrix $M_S(D)$ for the system of equations for each incidence $(i,j) \in I$: $A^j x_i + B^j y_i + \ldots + C^j z_i + w_i + D^j = 0$, where the variables are ordered:

$$\ldots, D^j, \ldots \; ; \; \ldots, x_i, y_i, \ldots, z_i, w_i, \ldots .$$

BASIC RESULTS

All results for polyhedral pictures dualize to parallel drawings. Again, for parallel drawings there is no distinction between continuous changes and first-order changes. The trivially parallel drawings, generated by d translations and one dilatation towards a point, form a vector space of dimension $d+1$, provided there are at least two distinct points (Figure 49.2.3A). (A trivially parallel drawing may even have all points coincident, though the faces will still have assigned directions (Figure 49.2.3A).)

THEOREM 49.2.3 *Parallel Drawing Theorem*

For generic selections of the directions D in d-space for the faces, a structure $S = (V, F; I)$ has a realization $S(p, P)$ with all points p distinct if and only if, for every nonempty set I' of incidences involving at least two points $V(I')$ and faces $F(I')$, $|I'| \leq d|V(I')| + |F(I')| - (d+1)$ (Figure 49.2.3A).

In particular, a configuration p, P with distinct points realizing generic directions for the incidence structure is unique, up to translation and dilatation, if and only if $|I| = d|V| + |F| - (d+1)$ *and* $|I'| \leq d|V'| + |F'| - (d+1)$.

Of course other nontrivially parallel drawings will also occur if the rank is smaller than $d|V'| + |F'| - (d+1)$ (Figure 49.2.3 B, with a generic rank 1 less than required for $d = 2$, and a geometric rank, as drawn, 2 less than required).

Figure 49.2.3 may also be interpreted as the parallel drawings of a "cube in 3-space." For spherical polyhedra, there is an isomorphism between the nontrivially parallel drawings in 3-space (the parallel drawings modulo the trivial drawings) and the nontrivially parallel drawings in a plane projection [CrW94]. Only the dimension (4 vs. 3) of the trivially parallel drawings will change with the projection.

49.2.3 CONNECTIONS TO OTHER FIELDS

FIRST-ORDER RIGIDITY

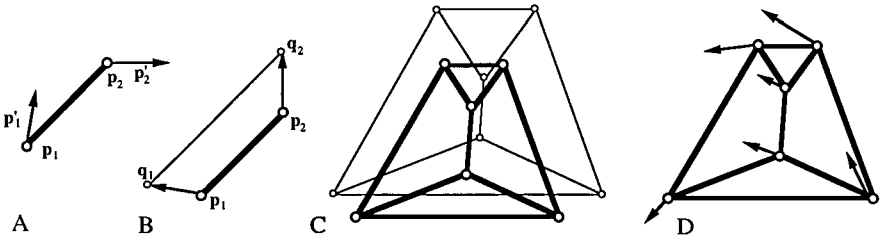
For any plane framework, if we turn the vectors of a first-order motion 90° (say clockwise), they become the vectors joining p to a parallel drawing q of the framework (Figure 49.2.4A,B). The converse is also true.

THEOREM 49.2.4

A plane framework $G(p)$ has a nontrivial first-order flex if and only if the configuration $G(p)$ has a nontrivially parallel drawing $G(q)$ (Figure 49.2.4C,D).

Because of this connection, combinatorial and geometric results for plane first-order rigidity and for plane parallel drawings have numerous deep connections. For

FIGURE 49.2.4



example, Laman's Theorem (Theorem 49.1.17) is a corollary of the Parallel Drawing Theorem, for $d = 2$. In higher dimensions, the connection is one-way: a nontrivial parallel drawing of a "framework" (the "direction of an edge" is represented by $d - 1$ facets through the two points) induces one (or more) nontrivial first-order motions of the corresponding bar framework. The theory of parallel drawing in higher space has the more complete theory and simpler algorithms than the theory of first-order rigidity, generalizing almost all results for plane first-order rigidity and plane parallel drawings.

MINKOWSKI DECOMPOSIBILITY

By a theorem of Shephard, a polytope is decomposable as the Minkowski sum of two simpler polyhedra if and only if the faces and vertices of the polytope (or the edges and vertices of the polytope) have a nontrivially parallel drawing. Many characterizations of Minkowski indecomposable polytopes can be deduced directly from results for parallel d-scenes (or equivalently, for polyhedral pictures of the polar polytope).

ANGLES IN CAD

In plane computer-aided design, many different patterns of constraints (lengths, angles, incidences of points and lines, etc.) are used to design or describe configurations of points and lines, up to congruence or local congruence. With distances between points, the geometry becomes that of first-order rigidity. If angles and incidences are added, even the problems of "generic rigidity" of constraints are unsolved (and perhaps not solvable in polynomial time). However, special designs, mixing lengths, distances of points to lines, and trees of angles have recently been solved, using direct extensions of the techniques and results for plane frameworks and plane parallel drawings [SW95, Whi94, Whi95].

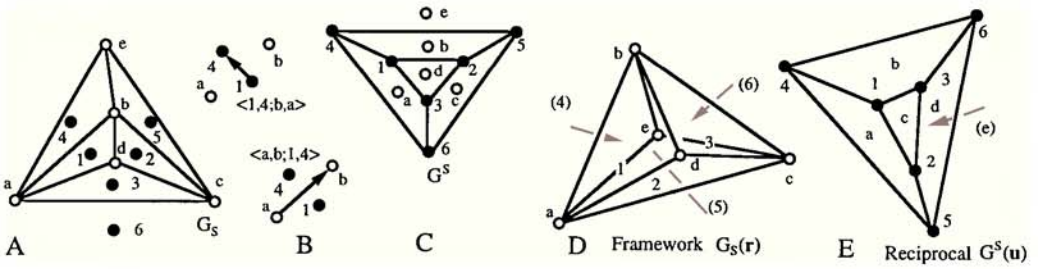
49.3 RECIPROCAL DIAGRAMS

The reciprocal diagram is a single geometric construction that has appeared, independently over a 130-year span, in areas such as "graphical statics" (drafting techniques for resolving forces in frameworks), scene analysis, and computational geometry.

GLOSSARY

Abstract spherical polyhedron $S = (V, F; \underline{E})$: For a 2-connected planar graph $G_S = (V, E_S)$, drawn without self-intersection on a sphere (or in the plane), we record the vertices as V and the regions as faces F, and rewrite the directed edges $\underline{E}$ as ordered 4-tuples $\underline{e} = \langle h, i; j, k \rangle$, where the edge from vertex h to vertex i has face j on the right and face k on the left. (The reversed edge $-\underline{e} = \langle i, h; k, j \rangle$ runs from i to h, with k on the right.)

FIGURE 49.3.1



Dual abstract spherical polyhedron: The abstract spherical polyhedron S^* formed by switching the roles of V and F, and switching the pairs of indices in each ordered edge $\underline{e} = \langle h, i; j, k \rangle$ into $\underline{e}^* = \langle j, k; i, h \rangle$. (Also the abstract spherical polyhedron formed by the dual planar graph $G^S = (F, E^S)$ of the original planar graph (Figure 49.3.1A,C).)

Proper spatial spherical polyhedron: An assignment of points $p_i = (x_i, y_i, z_i)$ to the vertices and planes $P^j = (A^j, B^j, D^j)$ to the faces of an abstract spherical polyhedron $(V, F; \underline{E})$, such that if vertex i and face j share an edge, then the point lies on the plane: $A^j x_i + B^j y_i + z_i + D^j = 0$; and at each edge the two vertices are distinct points and the two faces have distinct planes.

Projection of a proper spatial polyhedron $S(p, P)$: The plane framework $G_S(r)$, where r is the vertical projection of the points p (i.e., $r_i = \Pi p_i = (x_i, y_i)$) (Figure 49.3.2).

Gradient diagram of a proper spatial polyhedron $S(p, P)$: The plane framework $G^S(s)$, where $s_j = (A^j, B^j)$ is (minus) the gradient of the plane P^j (Figure 49.3.2).

Reciprocal diagrams: For an abstract spherical polyhedron S, two frameworks $G_S(r)$ and $G^S(s)$ on the graph and the dual graph of the polyhedron, such that for each directed edge $\langle h, i; j, k \rangle \in \underline{E}$, $(r_h - r_i) \cdot (s_j - s_k) = 0$ (Figure 49.3.1D,E).

BASIC RESULTS

Reciprocal diagrams have deep connections to both of our previous topics:

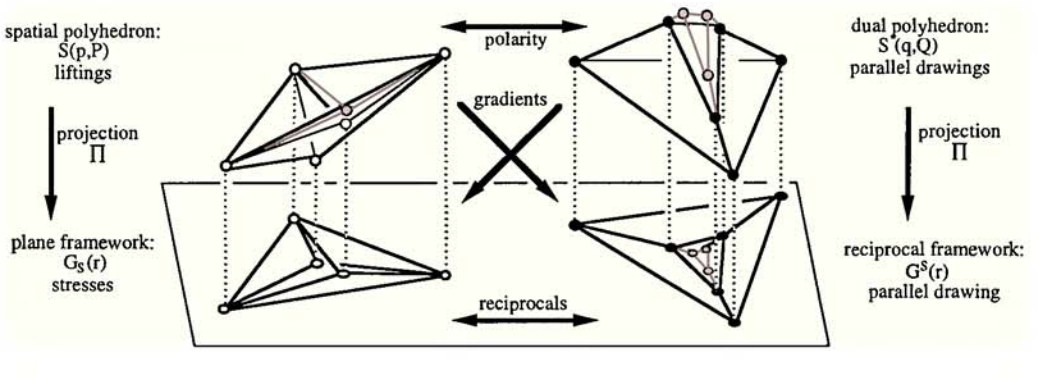
(a) Given a spatial scene on a spherical structure, with no faces vertical, the verti-

cal projection and the gradient diagram are reciprocal diagrams. (This follows because the difference of the gradients at an edge is a vector perpendicular to the vertical plane through the edge.)

(b) Given a pair of reciprocal diagrams on $S = (V, F; E)$, then for each edge $\underline{e} = \langle h, i; j, k \rangle$ the scalars ω_{ij} defined by $\omega(r_h - r_i) = (s_j - s_k)^{\perp}$ (where $^{\perp}$ means rotate by 90° clockwise) form a self-stress on the framework $G_S(r)$. (This follows because the closed polygon of a face in $G^S(s)$ is, after $^{\perp}$, the vector sum for the "vertex equilibrium" in the self-stress condition.)

These facts can be extended to other oriented polyhedra and their projections. The real surprise is that, for spherical polyhedra, the converses hold and all these concepts are equivalent (an observation dating back to Clerk Maxwell and the drafting techniques of graphical statics).

FIGURE 49.3.2



THEOREM 49.3.1 *Maxwell's Theorem*

For an an abstract spherical polyhedron $(V, F; \underline{E})$, the following are equivalent:

(a) The framework $G_S(r)$, with the vertices of each edge distinct, has a self-stress nonzero on all edges;

(b) $G_S(r)$ has a reciprocal framework $G^S(s)$ with the vertices of each edge distinct;

(c) $G_S(r)$ is the vertical projection of a proper spatial polyhedron $S(p, P)$;

(d) $G_S(r)$ is the gradient diagram of a proper spatial polyhedron $S^*(q, Q)$.

This pattern of "reciprocal constructions" and the connection to liftings to polytopes in the next space generalizes to higher dimensions [CrW93]. There are other refinements of this theorem, which connect the space of self-stresses of $G_S(r)$ with the space of parallel drawings (and first-order flexes) of $G^S(s)$, the space of polyhedra $S(p, P)$ with the same projection, and the space of parallel drawings of $S^*(q, Q)$ [CrW94] (Figure 49.3.2). A second refinement connects the local convexity of the edge of the polyhedron with the sign of the self-stress.

THEOREM 49.3.2 *Convex Self-Stress*

The vertical projection of a strictly convex polyhedron, with no faces vertical, produces a plane framework with a self-stress that is < 0 *on the boundary edges (the edges bounding the infinite region of the plane) and* > 0 *on all edges interior to this boundary polygon.*

A plane Delaunay triangulation also has a basic "reciprocal" relationship to the plane Voronoi diagram: the edges joining vertices at the centers of the regions are perpendicular to edges of the polygon of the Voronoi regions surrounding the vertex. This pair of reciprocals is directly related to the projection of a spatial polyhedral cap. See Section 20.1.

49.4 SOURCES AND RELATED MATERIALS

SURVEYS AND BASIC SOURCES

All results not given an explicit reference can be traced through these surveys:

[CoW96]: A presentation of basic results for concepts of rigidity between first-order rigidity and rigidity for tensegrity frameworks.

[CrW]: A thorough introduction to a number of topics on the rigidity of frameworks.

[GSS93]: A monograph devoted to combinatorial results for the graphs of generically rigid frameworks, but with an extensive bibliography on many aspects of rigidity.

[Sug86]: A monograph on reconstruction of spatial polyhedral objects from plane pictures.

[Whi93]: A survey of results relating first-order rigidity to matroid theory and related matroids for scene analysis, and to multivariate splines.

[Whi96]: An expository article presenting matroidal aspects of first-order rigidity, scene analysis, and multivariate splines.

RELATED CHAPTERS

Chapter 20: Voronoi diagrams and Delaunay triangulations
Chapter 26: Geometric reconstruction problems
Chapter 44: Graph drawing
Chapter 48: Geometric applications of the Cayley-Grassmann algebra
Chapter 50: Sphere packing and coding theory

REFERENCES

[Con82] R. Connelly. A flexible sphere. *Math. Intelligencer*, 1:130–131 (1978).

[Con88] R. Connelly. Rigidity and sphere packing I, II. *Structural Topology*, 14:43–60 (1988) and 16:59–75 (1990).

[CoW96] R. Connelly and W. Whiteley. Second-order rigidity and pre-stress stability for tensegrity frameworks. *SIAM J. Discrete Math.*, 9:453–492 (1996).

[Cra] H. Crapo. On the generic rigidity of structures in the plane. *Adv. in Appl. Math.*, to appear.

[CrW93] H. Crapo and W. Whiteley. 3-stresses in 3-space and projections of 4-polytopes: Reciprocals, liftings and parallel configurations. Preprint, Dept. of Mathematics and Statistics, York Univ., North York, Ontario, 1993.

[CrW94] H. Crapo and W. Whiteley. Spaces of stresses, projections and parallel drawings for spherical polyhedra. *Contrib. Alg. Geom.*, 35:259–281 (1994).

[CrW] H. Crapo and W. Whiteley, editors. *The Geometry of Rigid Structures*, to appear. (Draft preprint chapters, Dept. of Mathematics and Statistics, York Univ., North York, Ontario.)

[GSS93] J. Graver, B. Servatius, and H. Servatius. *Combinatorial Rigidity.* Number 2 of *AMS Monographs.* Amer. Math. Soc., Providence, 1993.

[RW81] B. Roth and W. Whiteley. Tensegrity frameworks. *Trans. Amer. Math. Soc.*, 265:419–446 (1981).

[SW95] B. Servatius and W. Whiteley. Constraining plane configurations in CAD: combinatorics of directions and lengths. Preprint, York Univ., North York, Ontario, 1995.

[Sug86] K. Sugihara. *Machine Interpretation of Line Drawings.* MIT Press, Cambridge, 1986.

[Whi93] W. Whiteley. Matroids and rigidity. In Neil White, editor, *Matroid Applications*, pages 1–53. Cambridge University Press, 1993.

[Whi94] W. Whiteley. How to describe or design a polyhedron. *J. Intelligent Robotic Syst.*, 11:135–160 (1994).

[Whi95] W. Whiteley. Constraining plane configurations in CAD: geometry of directions and lengths. Preprint, York Univ., North York, Ontario, 1995.

[Whi96] W. Whiteley. Some matroids from discrete applied geometry. In J. Bonin, J. Oxley, and B. Servatius, editors, *Matroid Theory*, Volume 197 of *Contemp. Math.*, pages 171–311. Amer. Math. Soc., Providence, 1996.

50 SPHERE PACKING AND CODING THEORY

J. A. Rush

INTRODUCTION

In sphere packing, one asks how densely it is possible to fill Euclidean n-space, $\mathbb{R}^n$, with nonoverlapping balls of a fixed (and by homogeneity, irrelevant) radius. Touching of the boundaries is permitted.

In posing a code-theoretic analogue to the previous question, a finite field $GF(q)$ and a metric $D(x, y)$ on the finite vector space $GF(q)^n$ are specified; often $q = 2$ and D is the Hamming metric. Then one asks the size of a maximal subset of $GF(q)^n$ for which any two points are distance at least d apart. This situation is analogous in that balls of radius slightly less than $d/2$ around the points of the subset can be regarded as packing the finite vector space densely.

One frequently requires the centers of a sphere packing in $\mathbb{R}^n$ to form a lattice. The analogous code-theoretic requirement is that the centers be not merely a sub*set* but more stringently a sub*space*, i.e., that the codes be *linear*.

In Section 50.1 we consider sphere packing. We look at codes (including nonlinear and non-Hamming-metric codes) in Section 50.2, and at the construction of sphere packings, as well as packings of more general bodies, from error-correcting codes, in Section 50.3.

50.1 SPHERE PACKING AND QUADRATIC FORMS

SPHERE PACKING IN $\mathbb{R}^n$

The word "sphere," as used in packing theory, usually denotes a solid ball. This is in contrast to the usage in the rest of mathematics, where "sphere" almost always refers to the outer surface alone. For historical reasons, the subject seems destined always to be called "sphere packing," even though the terms "sphere" and "ball" are interchangeable within the sphere-packing literature.

GLOSSARY

The ***ball of radius r*** around the origin is

$$B^n(r) = \{x = (x_1, \ldots, x_n) \in \mathbb{R}^n \mid x_1^2 + \cdots + x_n^2 \leq r^2\}.$$

Its volume is $V_n r^n$, where

$$V_n = \int_{x \in B^n} dx_1 \cdots dx_n = \frac{2^n \Gamma(1 + 1/2)^n}{\Gamma(1 + n/2)} = \frac{\pi^{n/2}}{\Gamma(1 + n/2)}$$

is the volume of a unit ball $B^n = B^n(1)$.

Sphere packing: An arrangement of balls of the same radius, whose interiors are disjoint.

Lattice: The integral span of a basis of $\mathbb{R}^n$. Equivalently, a nonsingular linear transform of the points $\mathbb{Z}^n$ with integer coordinates.

Lattice packing of spheres: The centers of the balls in the packing are all the points of a lattice.

Density of a sphere packing: Let $\overline{P}$ be the union of balls in the packing P. The density of P is

$$\Delta(P) = \lim_{r\to\infty} \frac{\mathrm{Vol}(\overline{P} \cap B^n(r))}{V_n r^n}.$$

Maximum packing density of the sphere: This is $\delta(B^n) = \sup \Delta(P)$, where the supremum is over packings P of B^n.

Determinant of a lattice: The volume of the parallelepiped spanned by a basis for the lattice Λ, written $\det \Lambda$; it is independent of the basis. (Some authors call the determinant of a lattice the *square* of that volume. We refer to the squared volume as the ***determinant of the quadratic form associated with the lattice***; see below.)

Density of a lattice packing of spheres: If the minimum distance between points of the lattice Λ is $2r$, then Λ provides a packing for balls of radius r, and its density is $\Delta(\Lambda) = V_n r^n / \det \Lambda$.

The ***center density*** of a packing P is $\Delta(P)/V_n$, the number of ball centers per unit volume of space when the minimum distance between the centers is normalized to 2.

Maximum lattice-packing density of the sphere: The quantity $\delta_L(B^n) = \sup \Delta(\Lambda)$, the supremum being taken over lattice packings of B^n.

The main problem in the theory of sphere packing is the determination of the quantities δ and δ_L in a given dimension n. Dense packing was problem 18 of Hilbert's famous turn-of-the-century problem list [Hil01]. Some authors express results in terms of center density instead, or in terms of $\log_2$ of the center density.

QUADRATIC FORMS IN n VARIABLES

GLOSSARY

Quadratic form associated with a lattice: If a lattice is the integral span of the rows of the $n \times n$ matrix $L = (l_{ij})$, then this is the positive definite quadratic form

$$f_L(x) = f_L(x_1, \ldots, x_n) = \sum_{i,j=1}^{n} a_{ij} x_i x_j,$$

where $A = (a_{ij}) = LL^T$. (Here t means transpose.) The symmetric positive definite matrix A is called an ***inner product matrix*** for the lattice Λ, and $\det f = \det A$ is the determinant of the quadratic form associated with the lattice.

The ***arithmetic minimum*** of the positive definite quadratic form $f(x_1, \ldots, x_n)$ is $M(f) =$ the smallest value taken on by f on $\mathbb{Z}^n \setminus \{O\}$.

Hermite's constant is $\gamma_n = \sup(M(f)/\sqrt[n]{\det f})$, the supremum taken as f varies over positive definite quadratic forms in n variables.

If $M(f)/\sqrt[n]{\det f} = \gamma_n$, then f is called ***absolutely extreme***.

Hermite's constant is related to the maximum lattice-packing density of a sphere by

$$\left(\frac{\sqrt{\gamma_n}}{2}\right)^n = \delta_L(B^n).$$

Thus, the geometric problem of densest lattice packing of a sphere is equivalent to the number-theoretic problem of maximizing the arithmetic minimum of a positive definite quadratic form of fixed determinant. This well-known equivalence is often unstated; papers on arithmetic minima frequently don't mention sphere packing, and vice versa. The historical trend is toward stating results in terms of sphere packing.

GENERAL BOUNDS ON SPHERE-PACKING DENSITY

Clearly $\delta_L(B^n) \le \delta(B^n)$. It was shown recently by K. M. Ball [Bal93] that

$$\delta_L(B^n) \ge (n-1)2^{1-n}\zeta(n).$$

The right-hand side is $2^{-n(1+o(1))}$ for large n, and bounds of that form have been known for a long time [Min69]. Ball's result was a refinement, for spheres, of the ***Minkowski-Hlawka bound*** [Hla43],

$$\delta_L(G) \ge 2^{1-n}\zeta(n),$$

which is applicable to all compact, convex, O-symmetric bodies G.

In the other direction, there is the ***Kabatianski-Levenshtein bound*** [KL78], which asymptotically has the form

$$\delta(B^n) \le 2^{-(.599\ldots)n(1+o(1))}$$

for large n. This they obtained by estimating $A(n,\theta)$, the maximum number of points that can be placed on the surface of B^n, such that for any two of them, x and y say, $x \cdot y \le \cos\theta$, applying their rather complicated estimate to a bound due to Yaglom,

$$\delta(B^n) \le \left(\sin\frac{\theta}{2}\right)^n A(n+1,\theta) \quad \text{for} \quad 0 < \theta \le \pi,$$

and choosing θ to their best advantage, namely $\theta = 1.0995$. A sketch of Yaglom's proof can be found in [CS93, p. 265]. The exact form of the Kabatianski-Levenshtein bound is complicated; the reader can consult [CS93, Chapter 9].

For $n \le 42$, the Kabatianski-Levenshtein bound is not as good as the ***Rogers bound***, which is elegantly presented in [Rog64], and is given by $\delta(B^n) \le \sigma_n$, where σ_n is the fraction of a solid regular simplex of edge 2 in R^n that is covered by the $n+1$ unit balls centered at its vertices. The quantity σ_n is bounded above by

$$\frac{n^{\frac{n+3}{2}}\sqrt{\pi}\,2^{u-n}}{e^{1+\frac{n}{2}}\,\Gamma(1+\frac{n}{2})},$$

where $u \le \frac{21}{4n+10}$. For large n, we have $\sigma_n = \frac{n}{e}2^{-n/2}(1+o(1))$.

The bounds given above for δ_L of course yield bounds on γ_n.

LAMINATED LATTICES

Define Λ_0 as the trivial lattice consisting of one point. For $n = 1, 2, 3, ...$, we understand a laminated lattice Λ_n to be any n-dimensional lattice with these three properties: First, its minimum distance is 2. Second, some Λ_{n-1} is a sublattice. And third, Λ_n has minimal determinant among lattices satisfying the first two conditions.

Notice that it is not apparent from the definition how many laminated lattices there are. It turns out that there are two Λ_{11}'s, three Λ_{12}'s, and three Λ_{13}'s. For all the other values of n in $0 \le n \le 24$, there is exactly one Λ_n. There are exactly 23 different Λ_{25}'s, and for $n \ge 26$ there are probably a great many.

The Leech lattice, Λ_{24}, has a profound influence on all smaller dimensions, and indeed all of the closest lattice packings of spheres that have been found to date are sections of the Leech lattice. These dense lattices are the lattices Λ_n for $1 \le n \le 24$ excepting $n = 11, 12, 13$, for which there are denser cross sections of Λ_{24}, called K_{11}, K_{12}, and K_{13} in [CS93].

It seems reasonable to consider the dimensions up to 24 separately.

DIMENSIONS UP TO 24

The values of $\delta_L(B^n)$ are known (i.e., proved) in dimensions $n \le 8$, and conjectured with modest conviction in dimensions $9 \le n \le 24$. It is a theorem due to Thue [Thu10] that $\delta_L(B^2) = \delta(B^2) = \pi/\sqrt{12}$. This is the density of the usual hexagonal packing of circles in the plane, shown in Figure 50.1.1.

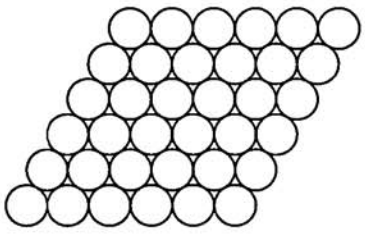


FIGURE 50.1.1
Closest packing of circles in the plane.

It may be the case, as far as is now known, that $\delta_L(B^n) = \delta(B^n)$ for all $n \le 24$ with the exception of $n = 10$, 11, 13, 20, and 22. (See Table 50.1.1.) The very recent 20-dimensional packing was obtained by Vardy [Var95]. Immediately afterward, Conway and Sloane [CS95] found that Vardy's nonlattice packing had analogues in dimension 22 (and dimensions 44 through 47), which also set new density records.

We know that $\delta_L(B^3) = \pi/\sqrt{18} = .7404....$ This is the density of the so-called "face-centered cubic" lattice, shown in Figure 50.1.2, which is generated by three equal vectors, each of which makes an angle of $\pi/3$ with the other two.

W.-Y. Hsiang's [Hsi93] claim to have proved that $\delta(B^3) = \delta_L(B^3)$ remains con-

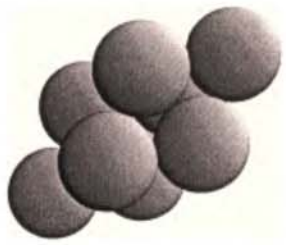


FIGURE 50.1.2
The densest lattice packing of spheres in three dimensions, which is probably also the densest packing.

troversial.[1] The Rogers bound gives $\delta(B^3) \leq .7796....$ The current record in three dimensions is due to Muder [Mud93], who found that $\delta(B^3) \leq .773055...$, improving on the old record $\delta(B^3) \leq .7784...$, due to Lindsey [Lin86].

TABLE 50.1.1 Comparison of possible values of δ and δ_L.

DIMENSION n	$\delta_L(B^n)/V_n$	$\delta(B^n)/V_n$	δ/δ_L
10	$\frac{1}{16\sqrt{3}}$	$\frac{5}{128}$	1.08523
11 and 13	$\frac{1}{18\sqrt{3}}$	$\frac{9}{256}$	1.09696
20	$\frac{1}{8}$	$\frac{7^{10}}{2^{31}}$	1.05230
22	$\frac{1}{2\sqrt{3}}$	$\frac{11^{11}}{2^{23}3^{10}\sqrt{3}}$	1.15198

The densest lattice packings known in dimensions up to $n = 10$ can be obtained from their predecessors merely by adjoining a new basis vector of the same length as all the others. Unfortunately the 11-dimensional dense lattice is not obtainable in this way; one must start from scratch.

Let $A = (a_{ij})$ be the following ten-by-ten matrix:

$$\begin{pmatrix} 4 & 2 & 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 2 & 4 & 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 \\ 2 & 2 & 4 & 2 & 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 4 & 2 & 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 2 & 2 & 4 & 2 & 0 & 0 & 0 & 1 \\ 2 & 2 & 2 & 2 & 2 & 4 & 2 & 2 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 2 & 4 & 2 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 4 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 4 & 1 \\ 0 & 0 & 0 & 1 & 1 & 2 & 1 & 1 & 1 & 4 \end{pmatrix}.$$

Our ability to build by adding fixed-length basis vectors through dimension ten is reflected algebraically in a property of this quadratic form due to Chaundy [Cha46]:

$$f(x_1, \ldots, x_{10}) = \sum_{i,j=1}^{10} a_{ij} x_i x_j$$

[1] However, no one doubts that $\delta(B^3) = \delta_L(B^3)$.

$$= 2(x_2 + x_3 + x_4 + x_5 + \tfrac{1}{2}x_6)^2 + 2(x_1 + x_2 + \tfrac{1}{2}x_6)^2 + 2(x_1 + x_3 + \tfrac{1}{2}x_6)^2$$
$$+ 2(x_4 + \tfrac{1}{2}x_6 + \tfrac{1}{2}x_{10})^2 + 2(x_5 + \tfrac{1}{2}x_6 + \tfrac{1}{2}x_{10})^2 + 2(x_7 + \tfrac{1}{2}x_6 + \tfrac{1}{2}x_{10})^2$$
$$+ 2(x_8 + \tfrac{1}{2}x_6 + \tfrac{1}{2}x_{10})^2 + 2(x_7 + x_8 + x_9 + \tfrac{1}{2}x_6)^2 + 2(x_9 + \tfrac{1}{2}x_{10})^2 + \tfrac{3}{2}x_{10}^2.$$

The property is that

$$g(x_1, \ldots, x_i) = f(x_1, \ldots, x_i, 0, \ldots, 0)$$

is an absolutely extreme quadratic form in i variables for $1 \leq i \leq 8$, and very probably is one for $i = 9, 10$ as well. Thus each new inner product matrix can be obtained from the previous one by adding a new column to the right, and its transpose to the bottom, of the matrix.

Although it is not possible to get past $n = 10$ in the manner described above, it is nonetheless possible, as stated in the previous section, to obtain all the best lattice packings known up to $n = 24$ by taking intersections of the 24-dimensional Leech lattice Λ_{24} (to be constructed in Section 50.3) with certain subspaces. Moreover, the highest center densities attained for lattice packings of B^n are symmetric about $n = 12$. Let us write $\xi(x)$ for the reciprocal of the presumedly optimal center density for a lattice in dimensions $12 \pm x$ for $0 \leq x \leq 12$, i.e.,

$$\xi(x) = \frac{V_{12+x}}{\delta_L(B^{12+x})} = \frac{V_{12-x}}{\delta_L(B^{12-x})}.$$

Its values are summarized in Table 50.1.2.

TABLE 50.1.2 Reciprocal center densities of the densest known lattice packings of B^n in dimensions up to 24.

x	12	11	10	9	8	7	6	5	4	3	2	1	0
$\xi(x)$	1	2	$2\sqrt{3}$	$4\sqrt{2}$	8	$8\sqrt{2}$	$8\sqrt{3}$	16	16	$16\sqrt{2}$	$16\sqrt{3}$	$18\sqrt{3}$	27

DIMENSIONS UP TO 2048

The best packings known in these dimensions are still fairly good, but less likely to be optimal than those of lower dimension. (See Table 50.1.3.) The success in these dimensions is due to the residual influences of combinatorial accidents like the existence of special algebraic curves, and the eight- and twenty-four-dimensional packings E_8 and Λ_{24}, the Leech lattice.

50.2 ERROR-CORRECTING CODES

We consider only codes whose symbols come from a finite field $GF(q)$, unless otherwise specified.

TABLE 50.1.3 Base 2 logarithms of center densities of some lattice packings in moderately large dimensions, in comparison with upper and lower bounds.

DIMENSION n	LOWER BOUND	ATTAINED	UPPER BOUND	SOURCE
32	−8.22	1.359	5.52	Quebbemann
36	−7.10	1.504	8.63	Kschischang-Pasupathy
48	−2.05	14.039	15.27	Thompson
54	1.27	15.88	25.86	Elkies
60	5.04	17.435	27.85	Kschischang-Pasupathy
64	7.79	24.71	31.14	Elkies
80	20.40	40.14	49.90	Shioda
104	43.38	67.01	80.20	"
128	70.28	97.40	118.6	Elkies
256	257.76	294.80	357.0	"
512	759.21	797.12	957.4	"
1024	2016.6	2018.2	2526.16	"
2048	5041.87	4891	6063.43	"

GLOSSARY

A ***code*** is a subset of $GF(q)^n$. A ***linear code*** is a subspace of $GF(q)^n$. ***Binary codes*** have $q = 2$. The elements of a code are called ***codewords***.

The ***Hamming weight*** $\|x\|$ of a codeword x is the number of nonzero symbols in it.

The ***Hamming distance*** of a code C is the minimum of $\|x - y\|$ for $x, y \in C$, $x \neq y$.

Fix q, the size of the ***symbol field***. An $[\boldsymbol{n}, \boldsymbol{k}, \boldsymbol{d}]$ ***code*** is a k-dimensional subspace of $GF(q)^n$ having Hamming distance at least d. k is called the ***dimension*** of the code, and n is called its ***block length***.

Fix q. An $(\boldsymbol{n}, \boldsymbol{M}, \boldsymbol{d})$ ***code*** is an M-element subset of $GF(q)^n$ for which the Hamming distance is at least d. $A(n, d)$ is the maximum possible value of M among (n, M, d) codes.

Fix q and n. For real x we write $\binom{x}{j} = x(x-1)(x-2)\cdots(x-j+1)/(j!)$ and define the ***Krawtchouk polynomial***

$$K_k(x) = \sum_{j=0}^{k} (-1)^j \binom{x}{j} \binom{n-x}{k-j} (q-1)^{k-j}.$$

Fix G, a compact, convex, O-symmetric body in $\mathbb{R}^n$, and fix an odd prime p. Regard $GF(p)^n$ as lying in $\mathbb{R}^n$ by making the identification

$$GF(p) = \{-(p-1)/2, \ldots, -1, 0, 1, \ldots, (p-1)/2\}.$$

Thus $GF(p)^n = \mathbb{Z}^n \cap pQ$ where Q is the unit hypercube

$$\{x \in \mathbb{R}^n \mid \max(|x_1|, \ldots, |x_n|) \leq \tfrac{1}{2}\}.$$

The ***G-norm*** of a point x of $GF(p)^n$ is

$$\|x\|_G = \inf\{\mu \geq 0 \mid x \in p\mathbb{Z}^n + \mu G\}.$$

An $[\boldsymbol{n}, \boldsymbol{k}, \boldsymbol{d}, \boldsymbol{p}, \boldsymbol{G}]$ ***code*** C is a k-dimensional subspace of $GF(p)^n$ such that $\|x - y\|_G \geq d$ whenever $x, y \in C$ with $x \neq y$.

GENERAL BOUNDS ON THE SIZE OF A CODE FOR THE HAMMING METRIC

In Table 50.2.1 we summarize the main discrete bounds on the size of a code. Asymptotic forms of the bounds we give can be found in various places. A particularly good list is in [TV91, pp. 607–612]. Surprisingly, the Gilbert-Varshamov bound is known to be nonoptimal as $n \to \infty$ for fixed symbol fields $GF(q)$ in which the prime power q is a square and at least 49, a result due to Tsfasman, Vlăduţ, and Zink [TVZ82]. See also [TV91] and [vL90].

Notation. Let us write $V_q(n, d)$ for the volume of a ball of radius d around a point $x \in GF(q)^n$ in the Hamming metric. Then

$$V_q(n, d) = \sum_{j=0}^{d} \binom{n}{j} (q-1)^j.$$

We write $\lceil x \rceil$ for the unique integer such that $x \leq \lceil x \rceil < x + 1$, and $\lfloor x \rfloor$ for the unique integer such that $\lfloor x \rfloor \leq x < \lfloor x \rfloor + 1$.

CYCLIC CODES

Here it is convenient and conventional to identify a vector $(a_0, \ldots, a_{n-1})$ in $GF(q)^n$ with a polynomial $a_0 + a_1 x^1 + \cdots + a_{n-1} x^{n-1}$.

GLOSSARY

Fix $n > 1$ and a symbol field $GF(q)$. Let $GF(q)[x]$ be the ring of polynomials in x with coefficients from $GF(q)$. Let $g(x) \in GF(q)[x]$ divide $x^n - 1$. Then the ideal $\langle g(x) \rangle$ in $GF(q)[x]/\langle x^n - 1 \rangle$ is called a ***cyclic code***. It has ***block length*** n, and ***dimension*** $k = n - \deg g$. The polynomial $g(x)$ is called the ***generator polynomial*** for the code.

Let $n = (q^m - 1)/(q - 1)$, and let β be a primitive nth root of unity in $GF(q^m)$. Also let m and $q - 1$ be relatively prime. Let $g(x) \in GF(q)[x]$ be the minimal polynomial of β. Then the ideal $\langle g(x) \rangle$ is called a ***Hamming code***. We take this nonstandard definition in order to treat Hamming codes as cyclic. The usual definition (see [MS78, p. 193]) yields a code that is equivalent to ours, in the sense that one code can be obtained from the other by applying a fixed permutation to each codeword.

Let α be a primitive nth root of unity in some extension field of $GF(q)$. Fix $t \geq 1$. Suppose that the generator polynomial $g(x)$ has roots $\alpha^t, \alpha^{t+1}, \alpha^{t+2}, \ldots, \alpha^{t+\mu-2}$ and that in fact $g(x)$ is the LCM of the minimal polynomials of those powers of α. Then $\langle g(x) \rangle$ is called a ***BCH code of designed distance μ***. If $t = 1$ it is called a ***narrow-sense BCH code***. If $n = q^m - 1$, so that α is a primitive element of $GF(q^m)$, then it is called a ***primitive BCH code***.

TABLE 50.2.1 Bounds on codes for the Hamming metric. We assume that n, k, d, and r are positive integers and that k and d do not exceed n. We write $s = (q-1)/q$ where $q \geq 2$ is the size of the symbol field. Upper bounds on $A(n,d)$ apply to linear codes too, since an $[n,k,d]$ code is an (n,M,d) code with $M = q^k$.

HYPOTHESIS	CONCLUSION	NAME OF BOUND
Bounds for nonlinear codes		
	$A(n,d) \geq q^n / V_q(n,d-1)$	Gilbert-Varshamov
	$A(n,d) \leq q^{n-d+1}$	Singleton
$d > (1-q^{-1})n$	$A(n,d) \leq d(d-(1-q^{-1})n)^{-1}$	Plotkin
$d = 3,5,7,9,11,\ldots$	$A(n,d) \leq q^n / V_q(n,(d-1)/2)$	Hamming
$sn/r \geq 1$, $r^2 - 2snr + snd > 0$	$A(n,d) \leq \dfrac{q^n snd}{V_q(n,r)(r^2-2snr+snd)}$	Elias
$B(x) = 1 + \sum_{j=1}^{n} B_j K_j(x)$, $B_1 \geq 0, \ldots, B_n \geq 0$, $B(j) \leq 0$ for $d \leq j \leq n$	$A(n,d) \leq B(0)$	linear programming
Bounds for linear codes		
$V_q(n,d-1) < q^{n+1-k}$	an $[n,k,d]$ code exists	Gilbert-Varshamov
An $[n,k,d]$ code exists	$k \leq n+1-d$	
An $[n,k,d]$ code exists	$n \geq \sum_{j=0}^{k-1} \lceil d/q^j \rceil$	Griesmer

A primitive BCH code for which $m = 1$, so that $n = q-1$, is called a ***Reed-Solomon code***.

Fix a prime block length n, and another prime number p for the size of the symbol field $GF(p)$. We require that n divides $p^{(n-1)/2} - 1$ so that p is a quadratic residue mod n. (If $p = 2$ this implies that n is of the form $8j \pm 1$.) Let Q_+ be the set of quadratic residues (i.e., squares) mod n, and let Q_- be the set of nonresidues. Let α be a primitive nth root of unity in some extension field of $GF(p)$, and let

$$q_+(x) = \prod_{j \in Q_+} (x - \alpha^j) \in GF(p)[x], \quad q_-(x) = \prod_{j \in Q_-} (x - \alpha^j) \in GF(p)[x].$$

Then $(x-1)q_+(x)q_-(x) = x^n - 1$, and there are four cyclic codes:

$$C_1 = \langle q_+(x) \rangle, \quad C_2 = \langle (x-1)q_+(x) \rangle, \quad C_3 = \langle q_-(x) \rangle, \quad C_4 = \langle (x-1)q_-(x) \rangle.$$

They are called ***quadratic residue (QR) codes***.

The ***Golay codes*** G_{23}, G_{11} are special quadratic residue codes over $GF(2)$ and $GF(3)$, respectively.

The parameters of these codes are found in Table 50.2.2.

OTHER LINEAR CODES FOR THE HAMMING METRIC

For brevity we sometimes write F_q for $GF(q)$.

GLOSSARY

The ***extended Golay codes*** G_{24}, G_{12} are obtained from the Golay codes by appending a digit, that is, an element of $GF(q)$, to each codeword to make the sum of the n digits of each codeword zero mod q. (Thus $n = 24$ and $q = 2$ for G_{24}, while $n = 12$ and $q = 3$ for G_{12}.)

The ***binary Reed-Muller code of order r***, where $0 \le r \le m$, consists of the vectors corresponding to all the polynomials of degree at most r over GF(2), in the variables $v_1, v_2, \ldots v_r$, which are Boolean functions of those binary variables.

A ***Goppa code*** is a linear code

$$C = \left\{ v = (v_1, \ldots, v_n) \in F_q^n \;\middle|\; \sum_{i=1}^{n} \frac{v_i}{z - P_i} \equiv 0 \bmod G(z) \right\}$$

where $v_i \in F_q$, $P_i \in F_{q^n}$, and $G(z)$ is a polynomial over F_{q^m} for which $G(P_i) \neq 0$, all of these holding for $1 \le i \le n$.

Recall from the previous section that a Reed-Solomon code is a BCH code over $GF(q)$ having block length $n = q - 1$.

Justesen codes may be defined as follows. Begin with a Reed-Solomon code C_0 over F_{q^n}, with parameters $[n, r, n+1-r]$, and let β be a primitive element of F_{q^m}. Let $y = (1, \beta, \beta^2, \ldots, \beta^{n-1})$. The Justesen code C is obtained by concatenating C_0 with the coordinatewise product of C_0 and y, and then replacing elements of F_{q^n} with n-tuples of elements of F_q. Thus

$$C = \{(c_1, \ldots, c_n; c_1, \beta c_2, \ldots, \beta^{n-1} c_n) \mid (c_1, \ldots, c_n) \in C_0\},$$

regarded as a vector space over F_q.

Algebraic-geometric codes are constructed as follows. Let X be a smooth, projective, algebraic curve over F_q, absolutely irreducible over F_q. Let $X(F_q) = \{P_1, \ldots P_n\}$ be the set of F_q points of X, so that $n = |X(F_q)|$. Let g be the genus of X, that is, the genus of the compact Riemann surface associated with X, in the sense of the Riemann-Roch theorem. Choose a divisor D of X, whose associated vector space $L(D)$ has dimension k over F_q. Our code $C \subset F_q^n$ is the image of $L(D)$ under the evaluation map $\mathrm{Ev} : L(D) \to F_q^n$, $\mathrm{Ev} : f \mapsto (f(P_1), \ldots, f(P_n))$.

The parameters of these codes can be found in Table 50.2.2.

CODES FOR EXOTIC METRICS

Let p be an odd prime. The author [Rus89] used $[n, k, d, p, G]$ codes and Construction A of Leech and Sloane (described in Section 50.3) with $G = B^n$ to produce packings of the sphere with density $2^{-n(1+o(1))}$ as $n \to \infty$.

Fix a compact, convex, O-symmetric body G in $\mathbb{R}^n$. For $x \in GF(p)^n$, let

$$B_{p,d}^n(x) = \{ y \in GF(p)^n \mid \|y - x\|_G \le d \},$$

TABLE 50.2.2 Comparison of parameters for certain types of codes for the Hamming metric. Distances and dimensions are at least as large as stated.

TYPE	BLOCK LENGTH n	DIMENSION k	DISTANCE d
Hamming	$\frac{q^m-1}{q-1}$	$n-m$	3
Ext. Hamming	$\frac{q^m-1}{q-1}+1$	$n-m$	4
BCH	n	$n-\deg g(x)$	$d \geq \mu$
Primitive BCH	q^m-1	$n-\deg g(x)$	$d \geq \mu$
Reed-Solomon	$q-1$	$n-\mu+1$	$d=\mu$
QR codes C_1, C_3	prime	$(n+1)/2$	$d \geq \sqrt{n}$
QR codes C_2, C_4	prime	$(n-1)/2$	$d \geq \sqrt{n}$
Golay G_{11}	11	6	5
Golay G_{23}	23	12	7
Ext. Golay G_{12}	12	6	6
Ext. Golay G_{24}	24	12	8
Reed-Muller	2^m	$1+\binom{m}{1}+\cdots+\binom{m}{r}$	2^{m-r}
Goppa	n	$n-m\deg G$	$1+\deg G$
Justesen	$2mn$	mr	$\sum_{i=1}^{t} i\binom{2m}{i}(q-1)^i$, where t is max integer such that $\sum_{i=1}^{t}\binom{2m}{i}(q-1)^i \leq n-r+1$
Algebraic -geometric	$n=$ no. of $GF(q)$-points on curve, so that $n \leq q+1+2g\sqrt{q}$	$k=\dim L(D)$	$n+1-g-k$, where $g=$ genus of curve

and let $V_{n,p,d} = |B^n_{p,d}(x)|$ be its volume. (Note that $V_{n,p,d}$ does not depend on x.) It can be shown that $V_{n,p,d} \leq |\mathbb{Z}^n \cap dG|$.

There is an analogue of the Gilbert-Varshamov bound for these codes: an $[n,k,d,p,G]$ code exists, provided $k < n+1-\log_p((p-1)V_{n,p,d}/2)$.

50.3 CONSTRUCTIONS OF PACKINGS

While we know that $\delta(B^n) \geq \delta_L(B^n) \geq 2^{-n(1+o(1))}$, we don't know explicit arrangements nearly so dense when n is large. In principle, Minkowski reduction theory makes finding the densest lattice packing of B^n a finite problem (and those imbued with a pure enough mathematical spirit may be satisfied with this) but still it is nice to have explicit arrangements. Typically, there is a tradeoff: the more explicit, or "constructive," the method is, the worse it fares as $n \to \infty$.

We mention, below, five constructions of packings from codes. Constructions A, B, and C are due to Leech and Sloane; D to Bos, Conway, and Sloane; and E to Barnes and Sloane. For more details, see [CS93].

CONSTRUCTION A

If C is a binary $[n,k,d]$ code, its Construction A lattice is $\Lambda_A(C) = 2\mathbb{Z}^n + C$. (If C is

nonlinear, this gives a periodic but nonlattice arrangement.) We have $\det \Lambda_A(C) = 2^{n-k}$, and the lattice provides a packing for spheres of radius $\min(1, \frac{1}{2}\sqrt{d})$.

If C is an $[n, k, d, p, B^n]$ code, then its Construction A lattice is $\Lambda(C) = p\mathbb{Z}^n + C$. Then $\det \Lambda(C) = p^{n-k}$, and the lattice packs spheres of radius $\frac{1}{2}\min(d, p)$. If C is an $[n, k, d, p, G]$ code, and $d \le p$, then $\Lambda(C)$ packs the body $\frac{1}{2}dG$.

CONSTRUCTION B

Let C be a binary $[n, k, d]$ code for which every codeword has even Hamming weight. The Construction B lattice of C consists of all those points $(x_1, \ldots, x_n)$ of the Construction A lattice for which $x_1 + \cdots + x_n$ is divisible by 4. We can call it $\Lambda_B(C)$. We have $\det \Lambda_B(C) = 2^{n-k+1}$, and the lattice packs spheres of radius $\frac{1}{2}\min(\sqrt{d}, \sqrt{8})$. (If C is a nonlinear even-weight code, this gives a periodic but nonlattice arrangement.)

CONSTRUCTION C

Since this produces nonlattice packings, and Construction D applied to nested linear codes produces lattice packings of equal density, we omit a description of Construction C.

CONSTRUCTION D

Let C_i be a binary $[n, k_i, d_i]$ code, with $C_{i-1} \supset C_i$ and $d_i \ge 4^i/u$ for $1 \le i \le t$, where $u \in \{1, 2\}$. Let $C_0 = GF(2)^n$, so that $k_0 = n$ and $d_0 = 1$. Let $C_{t+1} = \{(0, \ldots, 0)\}$, so that $k_{t+1} = 0$ and $d_{t+1} = \infty$. Take a row-vector basis

$$c_1 = (c_{11}, c_{12}, \ldots, c_{1n}),\ c_2 = (c_{21}, c_{22}, \ldots, c_{2n}),\ \ldots,\ c_n = (c_{n1}, c_{n2}, \ldots, c_{nn})$$

spanning $GF(2)^n$, selected so that these row vectors can be permuted with one another to produce an upper triangular matrix, and so that $c_1, c_2, \ldots, c_{k_i}$ span C_i for $0 \le i \le t$. The Construction D lattice for this nested set of codes is

$$\Lambda_D(\{C_i\}) = \left\{ x + y \;\middle|\; x \in 2\mathbb{Z}^n \text{ and } y \in \sum_{i=1}^{t} \sum_{j=1}^{k_i} b_{ij} \frac{c_{ij}}{2^{i-1}},\ \text{where each } b_{ij} \in \{0, 1\} \right\}.$$

The lattice has determinant $2^{n-(k_1+k_2+\cdots+k_t)}$ and can pack spheres of radius $1/\sqrt{u}$.

There is a similar construction, Construction D′, which uses parity checks rather than generators, and produces lattices of the same density as those of Construction D. We omit the description.

CONSTRUCTION E

This is a sort of nonbinary version of Construction D. In this subsection only, we permit codes to have nonfield symbol sets. Thus a "linear code" is merely an additive abelian group, not a vector space over the symbol field as elsewhere in this chapter.

Let $\Lambda \subset \mathbb{R}^n$ be a lattice with minimum Euclidean distance d between its points. Let D be a dilatation composed with an orthogonal transformation. Fix integers $p \geq 1$ and $r \geq 0$. Suppose $D\Lambda \subset \Lambda$, and that $pD^{-1} = a_0D^0 + a_1D^1 + \cdots + a_rD^r$ for certain integers $a_0, \ldots, a_r$. Suppose all the $p^b - 1$ nonzero congruence classes of $D^{-1}\Lambda/\Lambda$ have minimum distance from the origin at least $d/\sqrt[n]{|\det D|}$.

Let C_i be a p^{bk_i}-element subgroup of E^m, where $E = \Lambda/D\Lambda \cong (\mathbb{Z}/p\mathbb{Z})^b$, and $C_{i-1} \supset C_i$, for $1 \leq i \leq t$. Endowing these with the Hamming metric, we regard C_i as an $[m, k_i, d_i]$ code. We assume that the largest code, C_0, has parameters $[m, m, 1]$.

Let $x \in \{0, 1, 2, \ldots, p-1\}$ belong to the congruence class $\bar{x} \in \mathbb{Z}/p\mathbb{Z}$. Let $V : E \to \Lambda$ be the map

$$\bar{x}_1v_1 + \cdots + \bar{x}_bv_b \mapsto x_1v_1 + \cdots + x_bv_b,$$

and use the same symbol V to denote the map $V : E^m \to \Lambda^m$ that operates componentwise.

Let row vectors $c_1, c_2, \ldots, c_{bk_i}$ be selected,

$$c_1 = (c_{1,1}, c_{1,2}, \ldots, c_{1,bk_i}), \ \ldots, \ c_{bk_i} = (c_{n,1}, c_{n,2}, \ldots, c_{bk_i,bk_i}),$$

so that a typical codeword of C_i can be written $\bar{x}_1c_1 + \cdots + \bar{x}_{bk_i}c_{bk_i}$, where each x_j is in $\{0, 1, 2, \ldots, p-1\}$, and so that the rows $c_1, c_2, \ldots, c_{bk_i}$ can be permuted with one another to form an upper triangular matrix, for $1 \leq i \leq t$.

The Construction E lattice is the mn-dimensional lattice L_t given as follows: Let

$$M_i = \Big\{ x_1D^{-i}V(c_1) + \cdots + x_{bk_i}D^{-i}V\big(c_{bk_i}\big) \mid x_1, \ldots, x_{bk_i} \in \{0, 1, 2, \ldots, p-1\} \Big\}.$$

Let $L_0 = \Lambda^m$. For $1 \leq i \leq t$, we define

$$L_i = L_{i-1} + M_i = \{x + y \mid x \in L_{i-1}, \quad y \in M_i\}.$$

Construction E produces a lattice in $\mathbb{R}^{mn}$ whose determinant is

$$\frac{(\det \Lambda)^m}{\exp_p\big(b(k_1 + k_2 + \cdots + k_t)\big)}$$

and that lattice packs spheres of radius

$$(1/2) \min_{0 \leq j \leq t} \left(d\,|\det D|^{-j/n} \sqrt{d_j} \right).$$

E_8 AND THE LEECH LATTICE Λ_{24}

E_8 and Λ_{24} are anomalously dense and symmetrical lattice packings in $\mathbb{R}^8$ and $\mathbb{R}^{24}$, respectively. They have far more constructions than we can mention here.

Let L be the lattice $\{(x_1, \ldots, x_8) \in \mathbb{Z}^8 \mid x_1 + \cdots + x_8 \text{ is even}\}$. Then E_8 is

$$L \cup \Big(L + (\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}) \Big).$$

Alternatively, one gets E_8 by applying Construction A to the binary extended Hamming code $[8, 4, 4]$, which is the span over $GF(2)$ of the rows of this array:

$$\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

When scaled so that $\det E_8 = 1$, it packs spheres of radius $\sqrt{1/2}$. Each sphere touches 240 others, and that is known to be the maximum number possible.

Our construction of the Leech lattice will be based on the extended Golay code G_{24}, with parameters $[24, 12, 8]$, which is the span over $GF(2)$ of the rows of this array:

$$\begin{array}{cccccccccccccccccccccccc} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

Let 0 be the all-zeros vector in $\mathbb{R}^{24}$, and 1 the all-ones vector. Let v vary over G_{24}, and let x_{odd} and x_{even} vary over the points of $\mathbb{Z}^{24}$ for which $\sum_{i=1}^{24} x_i$ is odd or even, respectively. Let $T_1 = \{0 + 2v + 4x_{\text{even}}\}$, and $T_2 = \{1 + 2v + 4x_{\text{odd}}\}$. Then the ***Leech lattice*** is

$$\Lambda_{24} = (T_1 \cup T_2)/\sqrt{8}.$$

When scaled so that $\det \Lambda_{24} = 1$, it packs spheres of radius 1. Each sphere touches 196,560 others, and that is the highest possible contact number. Its automorphism group modulo reflection through the origin is the finite simple sporadic group Co_0 of order $2^{21}3^9 5^4 7^2 11 \cdot 13 \cdot 23$.

SUPERBALLS

By applying Construction A to $[n, k, d, p, G]$ codes, where G is a rather general type of body called a *superball*, it is possible to get extremely dense packings of these bodies. In fact the density is always at least $2^{-n(1+o(1))}$, like the Minkowski-Hlawka bound. Often the density is much greater. Papers containing details on these matters include [Rus93], which contains further references.

Fix k, and let n be a multiple of k.

A ***superball function*** $f : \mathbb{R}^k \to \mathbb{R}$ is a function with the following four properties:

$$f(x) > 0 \quad \text{except} \quad f(0) = 0;$$

$$f(x) = f(-x);$$

if $t > 0$ then there exists a nonsingular linear transformation A on $\mathbb{R}^k$ such that

$$tf(x) = f(Ax)$$

holds identically in x; and finally, if $0 \le \theta \le 1$ then

$$f(\theta x + (1-\theta)y) \le \theta f(x) + (1-\theta)f(y).$$

A ***superball*** is a body in $\mathbb{R}^n$ given by

$$f(x_1, \ldots, x_k) + f(x_{k+1}, \ldots, x_{2k}) + \cdots + f(x_{n-k+1}, \ldots, x_n) \le 1,$$

where f is a superball function. Let G_f denote that superball. In [Rus93] it was shown that

$$\delta_L(G_f) \ge \left(\frac{1}{2} \left(\sup_{A \in GL_n(\mathbf{R})} \frac{\int_{x \in \mathbf{R}^k} \exp(-f(Ax))dV}{\sum_{x \in \mathbf{Z}^k} \exp(-f(Ax))} \right)^{1/k} \right)^{n(1+o(1))} \quad \text{as} \quad n \to \infty.$$

The author conjectures that this holds with equality. In the case $k = 1$ and $f(x) = x^2$, the conjecture is that $\delta_L(B^n) = 2^{-n(1+o(1))}$.

50.4 SOURCES AND RELATED MATERIAL

For basic results and further references on the geometry of numbers, see [Cas59], [GL87], and [Min96]; for sphere packing, [CS93]; for packing and covering in general, [Dav64], [Fej72], and [Rog64]; for coding theory, [vL82], [vL90], [MS78], and [TV91].

RELATED CHAPTERS

Chapter 2: Packing and covering
Chapter 7: Lattice points and lattice polytopes
Chapter 51: Crystals and quasicrystals.

REFERENCES

[Bal93] K.M. Ball. A lower bound for the optimal density of lattice packings. *Internat. Math. Res. Notices*, 10:217–221, 1993.

[Cas59] J.W.S. Cassels. *An Introduction to the Geometry of Numbers.* Springer-Verlag, New York, 1959.

[Cha46] T.W. Chaundy. The arithmetic minima of positive quadratic forms I. *Quart. J. Math.*, 17:166–192, 1946.

[CS93] J.H. Conway and N.J.A. Sloane. *Sphere Packings, Lattices and Groups*, 2nd edition. Springer-Verlag, New York, 1993.

[CS95] J.H. Conway and N.J.A. Sloane. The antipode construction for sphere packings. Preprint, 1995.

[Dav64] H. Davenport. Problems of packing and covering. *Rend. Sem. Mat. Univ. Politec. Torino*, 24:41–48, 1964/1965.

[Fej72] L. Fejes Tóth. *Lagerungen in der Ebene auf der Kugel und im Raum*, 2nd edition. Volume 65 of *Grundlehren Math. Wiss.* Springer-Verlag, Berlin/New York, 1972.

[GL87] P.M. Gruber and C.G. Lekkerkerker. *Geometry of Numbers.* Elsevier, Amsterdam, 1987.

[Hil01] D. Hilbert. Mathematische Probleme. *Archiv. Math. Phys.*, 1:44-63, 1901.

[Hla43] E. Hlawka. Zur Geometrie der Zahlen. *Math. Z.*, 49:285–312, 1943.

[Hsi93] W.-Y. Hsiang. On the sphere packing problem and the proof of Kepler's conjecture. *Internat. J. Math.*, 4:739–831, 1993.

[KL78] G.A. Kabatianski and V.I. Levenshtein. Bounds for packings on a sphere and in space (in Russian). *Problemy Peredachi Informatsii*, 14:3–25, 1978; English translation in *Problems Inform. Transmission*, 14:1–17, 1978.

[Lin86] J.H. Lindsey, II. Sphere packing in R^3. *Mathematika*, 33:137–147, 1986.

[vL82] J.H. van Lint. *Introduction to Coding Theory.* Springer-Verlag, New York, 1982.

[vL90] J.H. van Lint. Algebraic geometric codes. In D. Ray-Chaudhuri, editor, *Coding Theory and Design Theory I.* Springer-Verlag, New York, 1990.

[MS78] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes.* North-Holland, Amsterdam, 1978.

[Min96] H. Minkowski. *Geometrie der Zahlen I.* Teubner, Leipzig, 1896.

[Min69] H. Minkowski. *Gesammelte Abhandlungen* (reprint). Chelsea, New York, 1969.

[Mud93] D.J. Muder. A new bound on the local density of sphere packings. *Discrete Comput. Geom.*, 10:351–375, 1993.

[Rog64] C.A. Rogers. *Packing and Covering.* Cambridge University Press, Cambridge, 1964.

[Rus89] J.A. Rush. A lower bound on packing density. *Invent. Math.*, 98:499–509, 1989.

[Rus93] J.A. Rush. A bound, and a conjecture, on the maximum lattice-packing density of a superball. *Mathematika*, 40:137–143, 1993.

[Tho] J.G. Thompson. Personal communication to N.J.A. Sloane.

[Thu10] A. Thue. Über die dichteste Zusammenstellung von kongruenten Kreisen in einer Ebene. *Skr. Vidensk-Selsk., Christ.*, 1:1–9, 1910.

[TV91] M.A. Tsfasman and S.G. Vlăduţ. *Algebraic-Geometric Codes.* Kluwer, Dordrecht, 1991.

[TVZ82] M.A. Tsfasman, S.G. Vlăduţ, and T. Zink. Modular curves, Shimura curves, and Goppa codes better than the Varshamov-Gilbert bound. *Math. Nachr.*, 109:21–28, 1982.

[Var95] A. Vardy. A new sphere packing in 20 dimensions. *Invent. Math.*, 121:119–133, 1995.

51 CRYSTALS AND QUASICRYSTALS

Marjorie Senechal

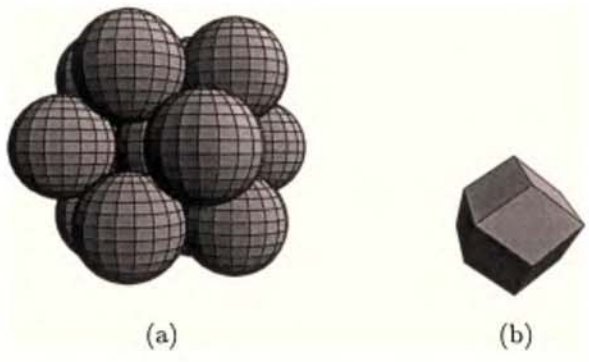
INTRODUCTION

Mathematical crystallography is the branch of discrete geometry that deals with the structure and form of crystals. For over a century the field has been a meeting ground for the theories of polytopes, lattices, tilings, and groups. Today, stimulated both by developments internal to mathematics and by the discovery of quasicrystals, the subject is broadening rapidly, and modeling the geometry of crystals requires an ever-expanding mathematical toolbag. In Section 51.1 we survey the classical foundations of the subject; in Section 51.2 we indicate how these foundations are being redesigned to encompass recent developments. *We assume that the reader is familiar with the terminology and results of Chapter 3 of this Handbook.*

51.1 PERIODIC CRYSTALS

The geometrical study of crystals began when Johann Kepler suggested that snowflakes were comprised of identical spheres arranged in what we now call cubic close-packing. Kepler also noted that if the spheres in such a packing were uniformly compressed, they would assume the forms of rhombic dodecahedra (Figure 51.1.1), and these dodecahedra would tile space. He thus demonstrated the duality between sphere-packing models and tiling models for crystal structure. This close relation between sphere packings and tilings, or more generally between point sets (the centers of the spheres in Kepler's case) and tilings, is exploited in mathematical crystallography to this day.

FIGURE 51.1.1
(a) *Cubic closest packing of spheres.* (b) *When the spheres are uniformly compressed they become space-filling rhombic dodecahedra.*



51.1.1 POINT SET MODELS

From the middle of the nineteenth century until quite recently, "regular systems of points," or unions of a finite number of them, have served as the abstract model for crystal structure. The study of crystal geometry amounted to the classification of these point sets by symmetry.

THE CLASSICAL THEORY

Let Γ be a discrete point set in $\mathbb{E}^n$.

GLOSSARY

Star (of a point $x \in \Gamma$): The configuration of line segments joining x to each of the other points of Γ.

Voronoi cell (of a point $x \in \Gamma$): The set $V(x)$ of points in $\mathbb{E}^n$ that are at least as close to x as to any other point of Γ. (See also Chapters 3 and 20.)

Voronoi tiling (associated with Γ): The tiling $\mathcal{V}_\Gamma$ whose tiles are the Voronoi cells of the points of Γ.

Regular system of points: An infinite discrete point set such that the stars of all its points are congruent; equivalently, a discrete point set that is an orbit of an infinite group of isometries.

Crystallographic group: A group of isometries that acts transitively on a regular system of points. Crystallographic groups are discrete subgroups of the nonabelian group of Euclidean motions of $\mathbb{E}^n$.

Lattice (of dimension n): A discrete subgroup of $\mathbb{R}^n$, generated by n linearly independent translations.

Crystal (classical): The union of a finite number of orbits of a crystallographic group.

Point lattice: An orbit of a lattice.

MAIN RESULTS

Table 51.1.1 gives the number of crystallographic groups in $\mathbb{E}^2$, $\mathbb{E}^3$, and $\mathbb{E}^4$, up to isomorphism. For $n \geq 5$ the number of groups is not known.

TABLE 51.1.1 Crystallographic groups.

n	2	3	4
TYPES	17	219	4783

THEOREM 51.1.1 *Bieberbach's Theorem*

A regular system of points is a finite union of translates of congruent lattices (Figure 51.1.2)*; the symmetry group G of a regular system of points is a product of a translation group T and a finite group of isometries, such that T is the maximal abelian subgroup of G.*

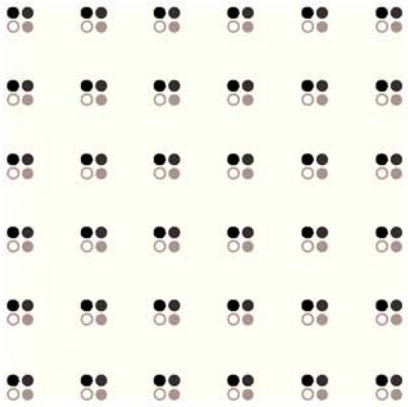


FIGURE 51.1.2
A regular system of points as a union of congruent lattices.

THEOREM 51.1.2 *The Crystallographic Restriction*

The only rotational symmetries possible for a regular system of points are those compatible with a lattice (of the same dimension).

Table 51.1.2 gives the possible orders m, $2 \leq m \leq 13$, of rotational symmetries of a regular system of points and the lowest dimension $d(m)$ in which they can occur. Five-fold rotations, as well as n-fold rotations with $n > 6$, are "forbidden" in $\mathbb{E}^2$ and $\mathbb{E}^3$.

TABLE 51.1.2 m-fold rotational symmetries.

m	$d(m)$	m	$d(m)$	m	$d(m)$	m	$d(m)$
2	1	5	4	8	4	11	12
3	2	6	2	9	6	12	4
4	2	7	6	10	4	13	12

DELONE'S REFORMULATION OF THE CLASSICAL FOUNDATIONS

In the 1930's Delone, Alexandrov, and Padurov reformulated the foundations of mathematical crystallography, replacing the regular systems of points with more general discrete point sets, which they called "(r, R)-systems."

GLOSSARY

(r,R) system: A set $\Lambda = \Lambda_{r,R}$ of points in $\mathbb{E}^n$ that is discrete and relatively dense (r is the infimum of the distances between pairs of points of Λ, and every sphere of radius $\geq R$ contains at least one point of Λ).

Delone (or ***Delaunay***) ***set:*** The modern term for an (r, R) system.

c-star (of a point x in a Delone set): The configuration of line segments joining x to the points of the Delone set that lie in $B(x, c)$, the ball with center x and radius c.

MAIN RESULTS

The Voronoi cell of any point x of a Delone set $\Lambda_{r,R}$ is contained in the ball $B(x, R)$; thus the cell is completely determined by $\Lambda \cap B(x, 2R)$.

If an orbit of a group of isometries of $\mathbb{E}^n$ is a Delone set, then the group is crystallographic and the Delone set is a regular system of points.

THEOREM 51.1.3 *The Local Theorem (for point sets)*

There is a real number k such that if all the $2Rk$-stars of a Delone set $\Lambda_{r,R}$ are congruent, then Λ is a regular system of points. (See also Section 3.2.)

PROBLEM 51.1.4

Does the constant k in the Local Theorem depend only on the dimension n?

51.1.2 TILING MODELS

Crystal growth is modular: beginning with a relatively tiny cluster of atoms, a crystal grows by the accretion of modules (atoms, molecules) to this "seed." The position that a module assumes on the growing crystal is assumed to be determined by local forces, as are subsequent rearrangements that may be required to minimize surface energy. In models of crystal structure consistent with this process, the modules are sometimes represented as spheres, but more commonly as space-filling polyhedra. In particular, it is convenient to think of a crystal as a tiling of space by congruent tiles. The tiles may be the crystallographer's "unit cells," the Voronoi cells of the crystal lattice, or stereohedra.

GLOSSARY

Unit cell (of a lattice in $\mathbb{E}^n$): The Minkowski sum of a set of n generating vectors of the lattice.

Zonotope: The Minkowski sum of an arbitary number of line segments (or vectors).

n-parallelotope: A convex n-polytope that tiles $\mathbb{E}^n$ by translation. (See Section 3.2.) Unit cells and Voronoi cells (of lattice points) are parallelotopes.

Stereohedron: The Voronoi cell of a point of a regular system of points. (A stereohedron is not necessarily a parallelotope.)

MAIN RESULTS

Table 51.1.3 gives the number of combinatorial types of n-parallelotopes in $\mathbb{E}^2$, $\mathbb{E}^3$, and $\mathbb{E}^4$.

TABLE 51.1.3 n-parallelotopes.

n	2	3	4
TYPES	2	5	52

A 2-parallelotope is combinatorially equivalent to a quadrilateral or a hexagon. The 3-parallelotopes are, combinatorially, cubes, hexagonal prisms, truncated octahedra, rhombic dodecahedra, and the "elongated" rhombic dodecahedra (which have four hexagonal and eight rhombic faces). The 2-parallelotopes and 3-parallelotopes are zonotopes, but this is not generally true in higher dimensions.

Every 2-, 3-, and 4-parallelotope is an affine image of the Voronoi cell of a lattice in $\mathbb{E}^2$, $\mathbb{E}^3$, and $\mathbb{E}^4$, respectively; this is also true in $\mathbb{E}^n$ if the parallelotope has $2(2^n - 1)$ faces (the maximal number in that dimension) [MRS95].

The number of combinatorial types of stereohedra in $\mathbb{E}^n$ is bounded (see Section 3.2).

PROBLEM 51.1.5

Is every n-parallelotope an affine image of the Voronoi cell of some full rank lattice point in $\mathbb{R}^n$? Voronoi proved that the answer is "yes" if exactly $n+1$ Voronoi cells meet at every vertex of the Voronoi tiling.

51.1.3 MODELING X-RAY DIFFRACTION

In 1912 the German physicist Max von Laue demonstrated the light-like nature of X-rays and the plausibility of a lattice structure for crystals by showing that crystals can serve as diffraction gratings for X-rays (this experiment also established the existence of atoms). X-ray diffraction turned out to be the Rosetta stone that unlocked the solid state. Synthetic pharmaceuticals, electronics, and medical imaging are only three of the many fields of application that have resulted from this discovery.

X-ray diffraction is far-field (Fraunhofer) diffraction. This means that the distances from the X-ray source to the crystal and from the crystal to the photographic plate on which scattered intensities are recorded are sufficiently far that the scattering can be modeled by Fourier transformation [Cow86].

GLOSSARY

Dual lattice: If L is a lattice in $\mathbb{E}^n$, its dual lattice L^* is the group of vectors $\vec{y} \in \mathbb{E}^n$ such that $\vec{y} \cdot \vec{x} \in \mathbb{Z}$ for every $\vec{x} \in L$; here $\cdot$ denotes the usual scalar product.

Dirac delta "function" at x: Intuitively, the generalized function δ_x that assigns unit mass to the point $x \in \mathbb{E}^n$ and vanishes at all other points.

MAIN RESULTS

Assume, for simplicity, that our regular system of points is a point lattice. We

associate to the corresponding lattice L the generalized function

$$\rho(x) = \sum_{x_n \in L} \delta_{x_n}(x); \tag{51.1.1}$$

its Fourier transform is the generalized function

$$\hat{\rho}(s) = \sum_{x_n \in L} \exp(-2\pi i x_n \cdot s), \tag{51.1.2}$$

where $s \in \mathbb{E}^n$. The diffraction pattern that we observe (on a photographic plate) when X-rays are passed through this "crystal" is a density map of the crystal's "intensity function" (the Fourier transform of the autocorrelation $\rho(x) * \overline{\rho(-x)}$). The "Wiener diagram" below, in which $\updownarrow$ denotes Fourier transformation, describes the relationship between the crystal and the observed intensities.

$$\begin{array}{ccc} \rho(x) & \overset{\text{autocorrelation}}{\longrightarrow} & \rho(x) * \overline{\rho(-x)} \\ \updownarrow & & \updownarrow \\ \hat{\rho}(s) & \overset{\text{squaring}}{\longrightarrow} & |\hat{\rho}(s)|^2 \end{array}$$

The task of the crystallographer is to deduce $\rho(x)$ from the intensity function, a task greatly complicated by the fact that the intensity is real while $\hat{\rho}(s)$ is complex.

This diagram is widely used in crystallography for heuristic purposes, although it is not valid (in any theory of generalized functions) because convolution and multiplication are not defined for infinite sums of deltas. Nevertheless, there is a sense in which it gives correct information [Hof95]. In particular, sharp bright spots in the diffraction pattern correspond to delta functions in the Fourier transform in the case of periodic point sets (and also in the case of model sets, discussed in Section 51.2 below). The Poisson summation formula states, in effect, that the diffraction pattern of a point lattice is a set of sharp bright spots at the points of its dual point lattice.

THEOREM 51.1.6 *Poisson Summation Formula*
Let L and L^* be dual lattices, and let $\rho(x)$ be as in (51.1.1) *above. Then* (51.1.2) *can be written in the form*

$$\hat{\rho}(s) = \sum_{s_n \in L^*} \delta_{s_n}(s). \tag{51.1.3}$$

51.2 GENERALIZED CRYSTALS AND QUASICRYSTALS

After the discovery of X-ray diffraction in 1912, it was unquestioningly assumed that a crystal is a solid with a periodic atomic structure. Only a periodic structure, it was reasoned, could produce diffraction patterns with sharp bright spots, because—roughly speaking—the spots indicate the repetition, throughout the crystal's atomic pattern, of congruent c-stars for all $c > 0$. The "long-range order" created by this repetition, it was assumed, must be periodic. But this classical model began to be questioned in the 1970's when it was found that the structures of so-called

modulated crystals could not be accounted for by three-dimensional periodicity. The paradigm that had reigned since Laue's experiment collapsed completely with the discovery, in the early 1980's, of crystals with "forbidden" icosahedral symmetry. Today it is widely agreed that both periodic and nonperiodic crystals exist. But the structure of nonperiodic crystals is still not well understood. Rather than repeat the mistake of the past by again defining a crystal in terms of some a priori concept of its structure, the Commission on Aperiodic Crystals of the International Union of Crystallography has proposed as a working definition: *a crystal is a solid with an essentially discrete diffraction pattern.*

To put this into mathematical language, we follow the periodic model by associating a sum of Dirac deltas to a Delone set Λ, one delta at each point, and computing the Fourier transform of the autocorrelation (when the autocorrelation exists). This transform is a measure, called the spectrum of Λ; the spectrum can be uniquely decomposed into a sum of discrete and continuous measures. The discrete component of the spectrum is itself a countable sum of weighted Dirac deltas, located at a set of points that we will call Λ_d (when Λ is a lattice, $\Lambda_d = \Lambda^*$). We always have $0 \in \Lambda_d$; if $\Lambda_d \neq \{0\}$, it is said to be nontrivial. (The set Λ_d need not be discrete as a point set; in general it will be everywhere dense.)

Definition: A ***(generalized) crystal*** is a Delone set Λ with nontrivial Λ_d.

We will outline the implications of this definition in Section 51.2.4 below. First, we describe some generalizations of the notions of regular systems of points and stereohedra. It must be emphasized that at this early stage, all definitions are subject to change, and very few theorems have been proved in satisfactory generality (if they have been proved at all).

51.2.1 POINT SET MODELS

A point set model for a generalized crystal is a suitable generalization of a regular system of points, but so far there is no agreement on what "suitable" should mean. However, most models assume that the point set is a Delone set satisfying additional conditions, for example as in the definition above. Classification by symmetry group is replaced by local isomorphism classes.

GLOSSARY

c-atlas: The set of congruence classes of c-stars of the points of a Delone set Λ.

Repetitive point set: A Delone set Λ such that the stars of every c-atlas are relatively dense in Λ (i.e., for each such star s there is an $R_s > 0$ such that every ball of radius R_s contains a copy of s).

Local isomorphism class: Two Delone sets in $\mathbb{R}^n$ belong to the same local isomorphism class if every bounded configuration of each is relatively dense in the other.

X-ray diffraction patterns of real crystals suggest that point set models for generalized crystals in $\mathbb{E}^k$ should be subsets of $\mathbb{Z}$-modules of rank $m \geq k$. A $\mathbb{Z}$-module whose rank m is greater than its dimension k may be everywhere dense. One way to select the points of the crystal is by means of an "acceptance domain" or "window." A crystal obtained in this way is called a *model set* [Mey95].

Model set: Let L be a lattice of rank $m = k + n$ in $\mathbb{E}^m$; let $p_\parallel$ and $p_\perp$ be the orthogonal projections into a k-dimensional subspace $\mathcal{E} = \mathbb{E}^k$ and its orthogonal complement $\mathcal{E}^\perp = \mathbb{E}^n$, respectively. Assume that $p_\parallel$, restricted to L, is one-one and that $p_\perp(L)$ is everywhere dense in $\mathbb{E}^n$, and let Ω be a bounded subset of $\mathbb{E}^n$ with nonempy interior. Then the set

$$\Lambda(\Omega) = \{p_\parallel(x) \,|\, x \in L,\ p_\perp(x) \in \Omega\} \tag{51.2.1}$$

is called a model set. When Ω is a translate of the projection of the Voronoi cell of the lattice into the orthogonal space, the window is said to be ***canonical.***

The ingredients for a one-dimensional model set are shown in Figure 51.2.1, in which $m = 2$, $n = k = 1$, and L is a square lattice. The subspace $\mathcal{E}$ is the solid line of positive slope; the window Ω is the thick line segment in $\mathcal{E}^\perp$. A lattice point x is projected into $\mathcal{E}$ if and only if $p_\perp(x) \in \Omega$ (alternatively, if and only if x lies in the cylinder bounded by the dotted lines). Note that the window in Figure 51.2.1 is *not* canonical.

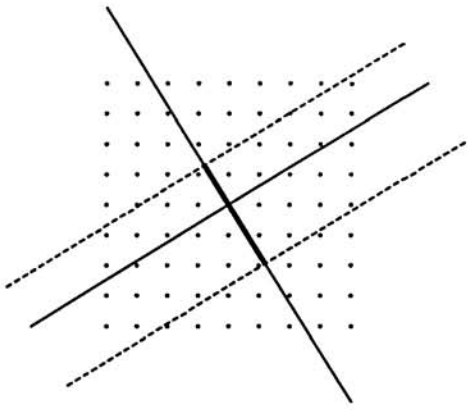


FIGURE 51.2.1
Ingredients for a one-dimensional model set. The subspace $\mathcal{E}$ is the solid line; the window Ω is the thick line segment in $\mathcal{E}^\perp$.

Meyer set: A Meyer set is a Delone set Λ such that $\Lambda - \Lambda$ is also a Delone set.

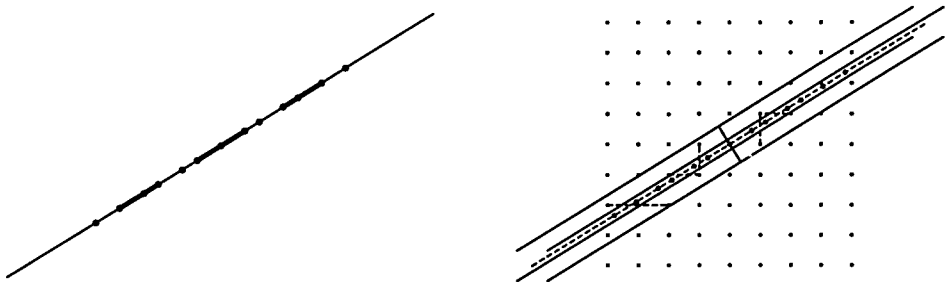
Pisot-Varaghavan (PV) number: An algebraic integer greater than one whose algebraic conjugates are less than one in absolute value.

MAIN RESULTS

Model sets are repetitive Delone sets. By translating Ω we get an infinite family of model sets in the same local isomorphism class. If the subspace $\mathcal{E}$ contains no points of the dual lattice L^*, then the model set is nonperiodic, i.e., it is not invariant under any translation.

When the window Ω is a translate of the projection, into $\mathcal{E}^\perp$, of the Voronoi cell of the lattice, the relative frequencies of the r-stars of $\Lambda(\Omega)$ are determined by the location of the corresponding points $p_\perp(x)$ in the window (see Figure 51.2.2).

FIGURE 51.2.2
Every point in a one-dimensional model set is the second point in a three-point configuration or star. (a) *In this example, there are three translations classes of such stars.* (b) *Each star is characterized by the interval, in the window Ω, into which the lattice point corresponding to its "center" projects. ($\mathcal{E}$ is the dotted line; the triples of lattice points projecting to the stars are also indicated by dotted lines.)*



Every model set is a Meyer set; conversely, every Meyer set is a subset of a set of the form $\Lambda(\Omega) + F$, where $\Lambda(\Omega)$ is a model set and F is finite [Mey95].

Λ is a Meyer set if and only if $\Lambda - \Lambda \subseteq \Lambda + F$, where F is finite [Lag].

OPEN PROBLEMS

PROBLEM 51.2.1

The most important open problem is the one posed by the discovery of nonperiodic crystals: What are necessary and sufficient conditions for a discrete point set to be a crystal according to the new definition? (It is not necessary that the set be Delone!)

PROBLEM 51.2.2

Is there an analogue of the Local Theorem for generalized crystals?

CONJECTURE 51.2.3

A Delone set Γ with inflation symmetry, i.e., $\lambda\Gamma \subset \Gamma$ for some $\lambda > 1$, is a Meyer set if and only if λ is a PV number.

51.2.2 TILING MODELS

Tiling models are useful in the theory of generalized crystals for precisely the same reasons they are useful in the classical periodic case: they give us a clearer picture of how space is partitioned than point set models do, and they may help us to understand the growth of crystals.

Which tilings are appropriate models for generalized crystals? High-resolution electron micrographs of many crystal structures can be interpreted meaningfully as tilings; often these tilings appear to be hierarchical (see Section 3.4). The hi-

erarchical structure may play a role in crystal formation or stability. Some of the tilings can be derived by the projection method (see below), for which there does not appear to be a physical interpretation. However, the projection method is of great theoretical value.

CANONICALLY PROJECTED TILINGS

Canonically projected tilings are closely related to the model sets defined in Section 51.2.1.

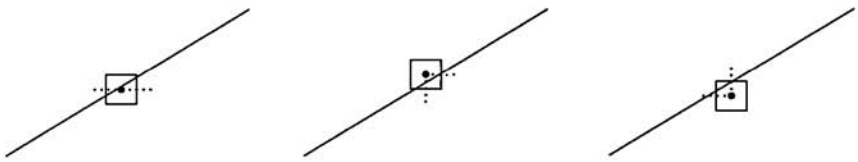
GLOSSARY

Canonical projection method for tilings: Let L be a lattice in $\mathbb{E}^m$, $\mathcal{E}$ a k-dimensional subspace, and $\vec{\gamma} \in \mathbb{E}^m$. Let $\mathcal{V}$ be the Voronoi tiling associated with L, and $\mathcal{D}$ the dual Delone tiling (see Section 3.1). The canonical projection method for tilings projects, onto $\mathcal{E}$, the $(n-k)$-dimensional faces of $\mathcal{D}$ that correspond, under duality, to the k-dimensional faces of $\mathcal{V}$ that are cut by $\mathcal{E} + \vec{\gamma}$ (Figure 51.2.3). Thus, if $\mathcal{E}$ meets the interior of a Voronoi cell $V(x)$ (dimension n), we project x onto $\mathcal{E}$: x is the vertex (dimension 0) of the Delone tiling that corresponds to $V(x)$ in the duality. The vector $\vec{\gamma}$ is the ***shift vector*** for the projection.

Canonically projected tiling: A tiling that can be constructed by the canonical projection method.

Note: Some authors require "canonical" to mean, in addition to the above, that L is the standard integer lattice.

FIGURE 51.2.3
The line $\mathcal{E}$ cuts a subset of the one- and two-dimensional faces of the Voronoi tiling (Voronoi cells are indicated by squares); we project the corresponding one- and zero-dimensional faces of the Delone tiling (dotted line segments and their endpoints).



MAIN RESULTS

Let $\mathcal{S}$ be a family of face-to-face tilings in $\mathbb{E}^k$ whose protoset is finite and admits tilings of exactly one local isomorphism class. Assume also that $\mathcal{S}$ is closed, in the sense that if $P_1 \subset P_2 \subset \cdots$ is a sequence of patches of $\mathcal{S}$ with unbounded in-radii, then $\cup_{i=1}^{\infty} P_i \in \mathcal{S}$. Then either $|S| = 1$ and that unique tiling is periodic, or $|S| = 2^{\aleph_0}$ [Dol95, Dan].

Let $\mathcal{E} + \vec{\gamma}$ be a translate of a k-dimensional subspace of $\mathbb{E}^n$, let L be an n-dimensional lattice with Voronoi tiling $\mathcal{V}$, and let $*$ be the dual map. Denote the

set of faces of $\mathcal{V}$ that have nonempty intersection with $\mathcal{E}$ by $\mathcal{V} \wedge \mathcal{E}_\gamma$, and let $p_\parallel$ be as above. Then

$$(\mathcal{V} \cap \mathcal{E}_\gamma)^* = p_\parallel((\mathcal{V} \wedge \mathcal{E}_\gamma)^*). \tag{51.2.2}$$

The set of vertices of a canonically projected tiling is a model set with a canonical window.

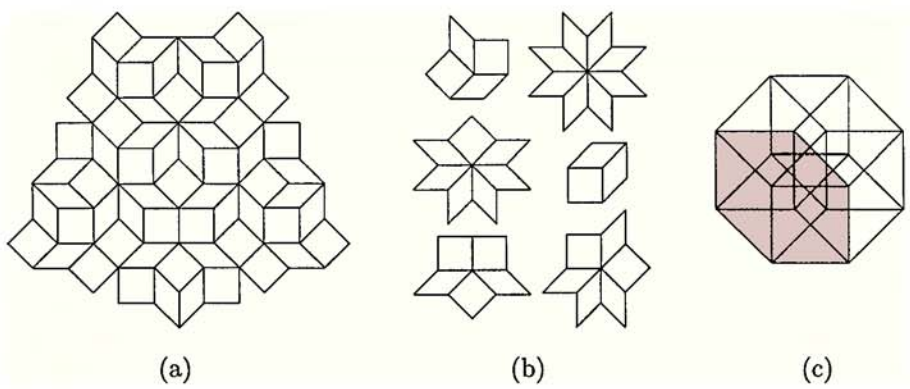
Some of the best known projected nonperiodic tilings are listed in Table 51.2.1. In all four cases, the lattice L is the standard integer lattice and the window Ω is a projection of a hypercube (the Voronoi cell of L). The vector $\vec{\gamma}$ is chosen so that $\mathcal{E} + \vec{\gamma}$ does not intersect any faces of $\mathcal{V}$ of dimension less than $n - k$ (thus only a subset of the faces of the Delone tiling $\mathcal{D}$ of dimensions $0, 1, \ldots, k$ will be projected). The first three tilings are hierarchical; by an unpublished "folk theorem," the fourth is too. $\mathcal{E}$ is a subspace that is stable under some finite rotation group; the rotational symmetry reappears in some of the bounded configurations of the tiling, and also in its diffraction pattern.

TABLE 51.2.1 Canonically projected nonperiodic tilings.

TILING FAMILY	L	$\mathcal{E}$
Fibonacci tiling	I_2	line with slope $1/\tau$ $\quad(\tau = (1+\sqrt{5})/2)$
Ammann tiling	I_4	plane stable under 8-fold rotation
Generalized Penrose tiling	I_5	plane stable under 5-fold rotation
Penrose 3D tiling	I_6	3-space stable under icosahedral rotation group

The famous Penrose tilings of $\mathbb{E}^2$ (by rhombs) are precisely those generalized Penrose tilings defined by the flats $\mathcal{E} + \vec{\gamma}$, $\vec{\gamma} \in \mathbb{E}^5$ such that $\vec{\gamma} \cdot \vec{w} \equiv \frac{1}{2} \pmod 1$, where $\vec{w} = (1, 1, 1, 1, 1)$.

FIGURE 51.2.4
The Ammann tiling. (a) *A portion of the tiling.* (b) *The six vertex stars.* (c) *The projected faces of the 4-cube are hexagons that decompose the window Ω into cells. There are six congruence classes of cells, corresponding to the six classes of stars. (A star of j tiles, $j = 3, \ldots, 8$, corresponds to the intersection of the projections of j hexagons.)*



The relative frequencies of the vertex stars of a canonically projected tiling are determined by the window: they are the ratios of volumes of the intersections of the projected faces of $V(0)$ (see Figures 51.2.2 and 51.2.4, and also [Sen95]).

THE MULTIGRID METHOD

The multigrid method is an important variant of the canonical projection method for tilings. In this version, the tiling is constructed as a dual of an n-grid, which is a superposition of n grids. For special choices of the grids and grid star, the n-grid is precisely the intersection $\mathcal{E}_\gamma \cap \mathcal{V}$ in (51.2.2); in these cases the multigrid and the canonical projection methods produce the same families of tilings. The multigrid method is more general but less studied [Sen].

GLOSSARY

Grid: A countably infinite family of equispaced parallel ($(k-1)$-dimensional) hyperplanes in $\mathbb{E}^k$.

Grid vector: A vector orthogonal to the grid whose length is the distance between adjacent hyperplanes of the grid.

n-grid (also ***multigrid***): A union of n grids (in $\mathbb{E}^k$).

Grid star: The set of grid vectors of an n-grid.

Shift vector (for n-grids in $\mathbb{E}^k$): We think of the n grids as initially passing through the origin, and then shift them so that at most k grids pass through a single point. The shift vector $\vec{\gamma}$ is the n-tuple of the shifts away from the origin; if at most k hyperplanes of the n-grid meet in any point, $\vec{\gamma}$ is said to be *regular*.

Note: We use the same symbol, $\vec{\gamma}$, for shift vectors and for translations of $\mathcal{E}$ to emphasize the fact that they play precisely the same role in the theory.

Pentagrid: A 5-grid in $\mathbb{E}^2$ whose star consists of unit vectors pointing from the center to the vertices of a regular pentagon (Figure 51.2.5). A pentagrid is said to be regular if its shift vector is regular.

n-grid dual: A tiling dual to an n-grid whose edges are parallel to the vectors of the grid star.

MAIN RESULTS

All of the main results in the subsection on Canonically Projected Tilings above can be reinterpreted in the language of multigrids and thus derived by the multigrid method.

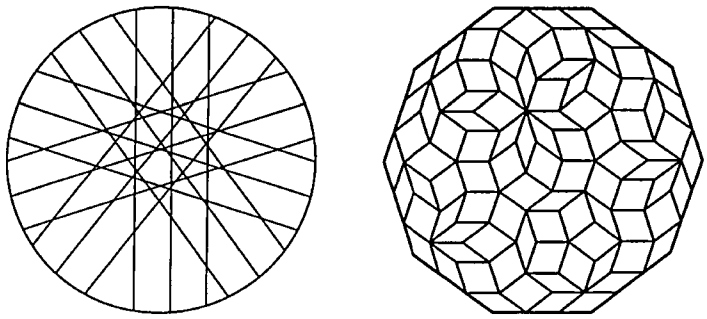
HIERARCHICAL TILINGS

These tilings are discussed in Section 3.4.

OPEN PROBLEMS

There is a large class of open problems concerned with the generality of these tiling construction methods and the relations among them.

FIGURE 51.2.5
A portion of a pentagrid (a) *and the corresponding patch of a generalized Penrose tiling* (b).



(a) (b)

PROBLEM 51.2.4

Which hierarchical tilings are projected tilings, and vice versa?

PROBLEM 51.2.5

Every tiling of $\mathbb{E}^k$ *by zonotopes has a pseudogrid dual (a grid in which the hyperplanes are replace by pseudohyperplanes—see Chapter* 6*); which of these pseudogrids are stretchable, and how is this related to the question of whether or not the tiling is a crystal?*

PROBLEM 51.2.6

Which tilings can be lifted to a surface (not necessarily contained in a cylinder) of faces of a lattice Delone complex in some higher-dimensional space?

PROBLEM 51.2.7

Does every repetitive nonperiodic tiling belong to a mutually locally derivable family of hierarchical tilings?

51.2.3 MODELING CRYSTAL "GROWTH"

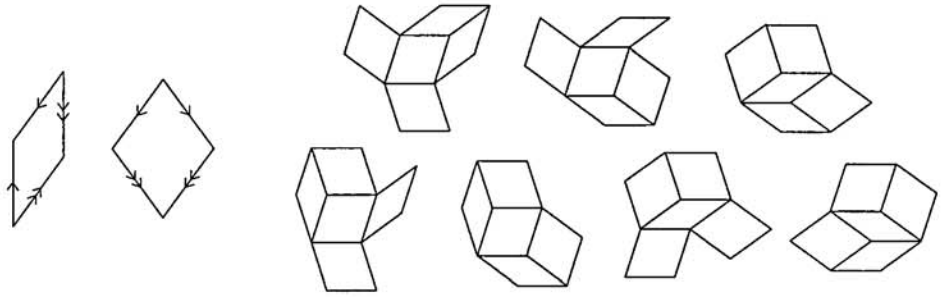
The classical notion of modeling crystal growth by tilings can be carried over to the more general setting, but now we want the tilings to be nonperiodic. Nonperiodic tilings can be constructed hierarchically (see Section 3.4), by projection, and by many other methods. However, if we require that the nonperiodicity be *forced* by "matching rules" of some sort, then there are fewer possibilities. There has been a great deal of progress in matching rule theory during the past decade, but many important problems are still open.

GLOSSARY

Aperiodic prototile set: A set of prototiles that admits only nonperiodic tilings (see Chapter 3). Some prototiles of aperiodic sets are simple shapes (with or without markings on them), but others can be quite complicated.

Matching rule (for a set of prototiles): A finite atlas, of some finite radius, of allowed configurations of marked or unmarked prototiles, such as face pairs, vertex stars, or face coronas (for definitions, see Chapter 3).

FIGURE 51.2.6
The matching rules for the Penrose tilings. (a) *The tiles can be marked, as shown here; the rule is that both the type (double or single) and direction of the arrows must match.* (b) *The configurations shown here constitute another (equivalent) matching rule for the Penrose tiles: each tile must be matched to four other tiles in one of these ways.*



Definition: A matching rule (for an aperiodic set of prototiles) is:
perfect, if it enforces nonperiodicity and repetitivity, and defines a single local isomorphism class;
strong, if it enforces nonperiodicity and repetitivity, but admits more than one local isomorphism class;
weak, if it enforces nonperiodicity but not repetitivity.

To prove that a prototile set is aperiodic, one must exhibit a matching rule that is at least weak.

There does not seem to be a name for matching rules that force periodic structures, but such rules do exist for certain prototile sets. For example, a tiling by squares will be periodic if we insist on a single vertex star, four squares meeting at a vertex. See also the classification of isohedral tilings in [GS87].

In cases when it is useful to distinguish between the prototiles and their shapes (for example, between marked and unmarked prototiles), it is helpful to make the following further distinctions: a matching rule, if it exists, is said to be ***local*** if a finite atlas of configurations of unmarked tiles suffices to characterize the matching rule (as for the Penrose tiles in Figure 51.2.6 above), and ***nonlocal*** if the tiles of the atlas must be decorated to characterize the rule. The matching rules for the Ammann tiling (see [GS87] and Figure 51.2.4) are nonlocal.

MAIN RESULTS

Given any aperiodic prototile set in $\mathbb{E}^2$, it is possible to construct a region homeomorphic to an annulus whose interior cannot be tiled with those prototiles [DS95]. (This property, well-known empirically to anyone who has ever played with Penrose tiles, is thus completely general; it follows that untileable "holes" in Penrose and other aperiodic tilings cannot be avoided by strengthening their matching rules.)

The proofs of the following results rely heavily on various theorems in discrete geometry, for example theorems of Helly type.

The atlas of face coronas of the Penrose tilings (Figure 51.2.6(b)) is a perfect local matching rule [deB]; the Ammann "octagonal" tiling does not have a local matching rule of any radius.

Those generalized Penrose tilings for which $\vec{\gamma} \cdot \vec{w} \in \frac{1}{2} + \mathbb{Z}[\tau]$ are in the same mutually locally derivable class as the Penrose tilings, and hence are self-similar and have perfect, local, matching rules [Le].

Nonlocal rules exist for all canonically projected tilings for which L is the integer lattice, $k = n$, and $\mathcal{E}$ is quadratic [LP93], and for all generalized Penrose tilings such that $\vec{\gamma} \cdot \vec{w} \in Q[\tau]$ [Le95].

OPEN PROBLEMS

Matching rules have been found for a large number of hierarchical tilings; in most cases, they are nonlocal. Do matching rules exist for all hierarchical tilings? Which are local and which are not?

Which tilings constructed by projection can be equipped with matching rules? Which are local and which are not?

51.2.4 X-RAY DIFFRACTION AND GENERALIZED CRYSTALS

As we noted in Section 51.2.1, a central question for generalized crystallography is: Which Delone sets are crystals? This question can also be posed in the context of tilings, since given a tiling, we can associate to it many different Delone sets, such as the vertex set of the tiling, or the barycenters of the tiles. Almost every aspect of this question is still open.

GLOSSARY

Quasicrystal: A generalized crystal whose intensity function is invariant under a rotational symmetry forbidden by the Crystallographic Restriction.

Symmetry group (of a quasicrystal): The group of isometries under which the intensity function is invariant.

Poisson comb: A crystal with purely discrete spectrum.

ϵ-dual (of a Delone set Λ): $\Lambda_\epsilon^* = \{y \in \mathbb{E}^n \,|\, |\exp(2\pi i y \cdot \lambda) - 1| \leq \epsilon, \forall \lambda \in \Lambda\}$.

MAIN RESULTS

A Delone set Λ is a Meyer set if and only if for every $\epsilon > 0$, Λ_ϵ^* is relatively dense in $\mathbb{E}^n$.

Every model set with suitably tame window is a Poisson comb.

OPEN PROBLEMS

PROBLEM 51.2.8

Which Meyer sets are crystals? (To answer this, it will first be necessary to clarify the relation between Λ_ϵ^* and Λ_d.)

51.3 SOURCES AND RELATED MATERIAL

SURVEYS

All results not given an explicit reference above may be traced in the following surveys.

[DS91]: Essays, mostly by physicists, on various aspects of quasicrystals and quasicrystal models.

[Jar89]: A collection of introductory essays on tiling models for quasicrystals.

[AG95]: Proceedings of the conference "Beyond Quasicrystals" held in Les Houches, France, March, 1994.

[Sen95]: A monograph devoted to the geometry of quasicrystal models.

[MP]: Proceedings of the NATO Advanced Study Institute "Mathematics of Aperiodic Order," held in Waterloo, Canada, August, 1995. This volume is 1996 state-of-the art.

RELATED CHAPTERS

Chapter 3: Tilings
Chapter 13: Basic properties of convex polytopes
Chapter 16: Symmetry of polytopes and polyhedra
Chapter 20: Voronoi diagrams and Delaunay triangulations
Chapter 50: Sphere packing and coding theory

REFERENCES

[AG95] F. Axel and D. Gratias, editors. *Beyond Quasicrystals.* Collection du Centre de Physique des Houches, Editions de Physique, Springer-Verlag, Berlin, 1995.

[Cow86] J.M. Cowley. *Diffraction Physics.* North Holland, Amsterdam, 1986.

[Dan] L. Danzer. Personal communication.

[deB] N.G. de Bruijn. Remarks on Penrose tilings. To appear.

[DS91] D. DiVincenzo and P.J. Steinhardt. *Quasicrystals: the State of the Art.* World Scientific, Singapore, 1991.

[Dol95] N. Dolbilin. The countability of a tiling family and the periodicity of a tiling. *Discrete Comput. Geom.,* 13:405–414, 1995.

[DS95] S. Dworkin and J.I. Shieh. Deceptions in quasicrystal growth. *Comm. Math. Phys.,* 168:337-352, 1995.

[GS87] B. Grünbaum and G.C. Shephard. *Tilings and Patterns.* Freeman, New York, 1987.

[Hof95] A. Hof. On diffraction by aperiodic structures. *Comm. Math. Phys.,* 169:25–43, 1995.

[Jar89] M.V. Jaric, editor. *Introduction to the Mathematics of Quasicrystals.* Academic Press, San Diego, 1989.

[Lag] J. Lagarias. Meyer's concept of quasicrystal and quasiregular sets. *Comm. Math. Phys.,* to appear.

[Le95] T.Q.T. Le. Local rules for pentagonal quasicrystals. *Discrete Comput. Geom.,* 14:31-70, 1995.

[Le] T.Q.T. Le. Local rules for quasiperiodic tilings. In [MP].

[LP93] T.Q.T. Le and S. Piunikhin. Local rules for multidimensional quasicrystals. *Diff. Geom. Appl.,* 5:13-31, 1993.

[Mey95] Y. Meyer. Quasicrystals, Diophantine approximation, and algebraic numbers. In F. Axel and D. Gratias, editors, *Beyond Quasicrystals.*, pages 3–16. Collection du Centre de Physique des Houches, Editions de Physique, Springer-Verlag, Berlin, 1995.

[Moo95] R. Moody. Meyer sets and the finite generation of quasicrystals. In B. Gruber, editor, *Symmetries in Science.* Plenum, New York, 1995.

[Mos95] R. Mosseri. Random tilings. In F. Axel and D. Gratias, editors, *Beyond Quasicrystals*, pages 335–354. Collection du Centre de Physique des Houches, Editions de Physique, Springer-Verlag, Berlin, 1995.

[MP] R. Moody and G. Patera. *The Mathematics of Aperiodic Order.* NATO Series, Kluwer, Dordrecht, to appear.

[MRS95] L. Michel, S.S. Ryshkov, and M. Senechal. An extension of Voronoï's theorem on primitive parallelotopes. *Europ. J. Combin.,* 16:59-63, 1995.

[Rad94] C. Radin. The pinwheel tilings of the plane. *Ann. of Math.,* 139:661-702, 1994.

[Sen95] M. Senechal. *Quasicrystals and Geometry.* Cambridge University Press, 1995.

[Sen] M. Senechal. A critique of the projection method. In [MP].

52 COMPUTATIONAL GEOMETRY SOFTWARE

Nina Amenta

INTRODUCTION

This chapter describes the current state of computational geometry software, which is improving rapidly. Many internet addresses are given in the references, but these are subject to change. An up-to-date on-line *Directory of Computational Geometry Software* is maintained at The Geometry Center [Ame]. The CGAL project [CGA], which proposes to develop an integrated library of computational geometry software, may be one future focus of change.

The software listed here varies from robust, well-documented, production quality programs to experimental, exploratory code. One important way in which the programs vary is in the treatment of numerical error and the presence of degeneracies in the input; computational geometry algorithms are sensitive and often crash outright when implemented naively. Robust implementation of geometric predicates is an active research area (see Section 35.1), and some publicly distributed implementations are listed below. Degeneracies can be handled as special cases, or eliminated by symbolic perturbation [Sei97] or by breaking ties lexicographically, which is less general but often adequate; see Section 35.5. We use the following definitions for different approaches to arithmetic.

GLOSSARY

Floating point: Normal floating-point arithmetic, with rounding.

Integer or rational: Fixed precision exact arithmetic.

Exact arithmetic: Extend precision of numbers as necessary to compute exact results.

Floating-point filters: Do arithmetic comparisons in floating point, invoke exact arithmetic when result is close to equality.

Adaptive precision: Do arithmetic comparisons only to the precision required to determine the sign.

Robust floating point: Algorithms designed to work robustly with inexact arithmetic.

52.1 TOOLS

We list available libraries for robust computation of geometric predicates (Table 52.1.1), libraries of data structures (Table 52.1.2), and programming environments for planar computational geometry (Table 52.1.3). The latter are systems

built around a graphical interface that displays input and output, allows user interaction, and supports animation. They each come with a large library of algorithms, including most of the two-dimensional problems covered elsewhere in this chapter.

TABLE 52.1.1 Arithmetic tools.

TOOL	FUNCTION/COMMENTS	LANG	ARITHMETIC
Robust predicates [Shea]	orientation and in-circle tests	C	adaptive float
`Real/Expr` [YDO]	exact arithmetic	C++	exact int, rational, float
	Efficient implementation; does $+, -, *, /, \sqrt{}$.		
Determinant sign [ABD$^+$]	2×2 and 3×3 determinants	C++	adaptive integer
	Uses floating-point arithmetic for integer computation [ABD$^+$95].		
`LN` [FV93]	floating-point filter compiler	C++	
Simulation of simplicity [Müc]	symbolic perturbation	C	exact integer
Toolkit [Reg]	multivariate polynml intersection	C	float, int, mixed arith
	Resultants, Sturm-Habicht sequences.		
Polynomial solver [Emib]	multivariate polynml intersection	C	floating point
	Resultants.		

TABLE 52.1.2 Objects and data structures.

TOOL	FUNCTION/COMMENTS	LANG	ARITHMETIC
`LEDA` [N$^+$]	objects, data structs, algs	C++	floating point, exact int or rational
	Library of data structures, planar geometric objects and primitives; graph algorithms. Numerical tools include floating-point filter, `REAL` data type. Large user community.		
`PLAGEO/SPAGEO` [Gie]	objects	C++	floating point
	Intersections and distances in two and three dimensions.		

TABLE 52.1.3 Environments.

PROGRAM	PLATFORM	COMMENTS
XYZGeobench [Schb]	Mac, ObjectPascal	Robust, well documented.
Geolab [dRJGM]	Sun, C++	Requires the Sun C++ compiler.
Workbench for CG [EKK$^+$]	Mac, Smalltalk	

Several of the tables throughout this chapter mention *Geomview*. Geomview is an interactive three-dimensional geometry viewing program written at the Geometry Center [Ame] available on several platforms. Its sophistication and generality have induced several geometry software developers to include Geomview output as an option.

52.2 CONVEX HULL AND RELATED PROBLEMS

Convex hull programs (Table 52.2.1) are versatile. The computation of a Delaunay triangulation, a Voronoi diagram, or a farthest-point Voronoi diagram in dimension d can be formulated as a $(d+1)$-dimensional convex hull computation (see Section 20.1). For polytopes containing the origin, there is a duality transformation between the convex hull of a set of points and the intersection of a set of linear halfspaces. It is easy to translate a set of points so that their convex hull contains the origin, but finding a translation of a set of halfspaces so that their intersection contains the origin might require linear programming. So it is generally easier to use a program that computes halfspace intersections to find a convex hull than vice versa. For an experimental comparison of `cdd`, `qrs`, and `qhull` on certain bad examples, see Table 19.3.1 and [AB95, ABS97].

TABLE 52.2.1 Arbitrary-dimensional convex hull programs.

PROGRAM	ALGORITHM/COMMENTS	ARITHMETIC
`hull` [Clab]	incremental	adaptive float
	Resolves degeneracies lexicographically. Optional randomization. Outputs include Voronoi diagrams, Delaunay triangulations, volume of Voronoi faces, and alpha shapes. Library interface, Geomview output. For arithmetic see [Cla92].	
`qhull` [BH]	incremental	floating point
	Computes approximate convex hulls to user-specified tolerances, thus avoiding issues of numerical accuracy and degeneracy. Computes Voronoi diagrams and Delaunay triangulations, farthest-point Voronoi diagrams and Delaunay triangulations, volume, and Voronoi cell volumes. Library interface, Geomview output.	
`cdd` [Fuk]	incremental	floating point or exact integer
	Halfspace intersection using Fourier-Motzkin elimination. C (floating point) or C++ (exact integer). Does not handle degeneracies. An auxiliary program computes projections to lower-dimensional subspaces.	
`lrs` [Avi]	pivoting	integer
	Halfspace intersection, using a deterministic transversal that minimizes storage. Can resolve degeneracies lexicographically. Computes volume.	
`chD` [Emia]	incremental	exact integer
	Uses a strong form of symbolic perturbation. Computes volume. Geomview output.	
`porta` [CL]	incremental	integer
	Halfspace intersection or convex hull using Fourier-Motzkin elimination. Also projections of polytopes, hyperplane-polytope intersection test, and counting integer points in a polytope. See Section 13.1.	

In dimensions two (Table 52.2.3) and three (Table 52.2.2), most attention has been paid to the Delaunay triangulation. Experimental comparison of Delaunay triangulation algorithms in two dimensions may be found in [SD95], [She96], and [For95].

TABLE 52.2.2 3D Voronoi diagram, Delaunay triangulation, and convex hull.

PROGRAM	ALGORITHM/COMMENTS	ARITHMETIC
`Detri` [Müc]	flipping	exact integer
	Robust 3D Delaunay. Uses symbolic perturbation for degeneracies.	
`tess` [Haz]	giftwrapping	floating point
	3D Delaunay with attention to numerical stability of predicates.	
`DeWall,InCoDe` [PCS]	div & conq, incremental	floating point
	3D Delaunay, speed-up through bucketing.	
`Delaunay tree` [Dev]	incremental	floating point
	Semi-dynamic (points are efficiently added or deleted if operations are in random order) two and three-dimensional Delaunay triangulation. In C++.	
Comput. Geom. in C [O'R94]	incremental	integer
	Three-dimensional convex hull.	

TABLE 52.2.3 2D Voronoi diagram, Delaunay triangulation, and convex hull.

PROBLEM	PROGRAM	ALGORITHM	ARITHMETIC
2D Voronoi/Delaunay	`voronoi` [For]	sweepline	floating point
2D Voronoi/Delaunay	Triangle [Sheb]	div & conq, incr'tal	adaptive float
2D Voronoi/Delaunay	[Lisa], Graphics Gem [Hec94]	incremental	floating point
2D Voronoi/Delaunay	Comput. Geom. in C [O'R94]	brute-force	integer
2D Voronoi/Delaunay	LEDA [N$^+$]	incremental	floating point
2D convex hull	2dch [Clad]	horizontal scan	floating point
2D convex hull	Comput. Geom. in C [O'R94]	Graham's scan	integer
2D convex hull	LEDA [N$^+$]	incremental	floating point

52.3 TRIANGULATION AND MESH GENERATION

Triangulation of simple polygons and of polygonal domains in the plane are viewed as theoretically solved problems. In terms of software, there are a few implementations of a randomized near-linear-time algorithm for a simple polygon. There are also several programs for constrained Delaunay triangulation and for the arbitrary triangulation or trapezoidation of a polygonal region. See Table 52.3.1. Finally, there are several programs which generate a mesh on a polygonal region, using Steiner points (Table 52.3.2).

The three-dimensional problems are active and important research areas, relevant to a developing market for commercial mesh generators. Until recently commercial systems constructed mainly structured hexahedral meshes, and required significant user interaction. Changes in solver software, however, now allow for the use of tetrahedral meshes, which require more computational geometry to con-

struct. Companies such as ICEM-CFD, SDRC, and Ansys produce tetrahedral meshing software. We list a few experimental automatic three-dimensional meshing programs in Table 52.3.3. For Web references to both academic and commercial three-dimensional meshing programs, see the list at [Scha], the survey [WVHC], and the comparison at [Hin].

TABLE 52.3.1 Two-dimensional triangulation and trapezoidation.

PROBLEM	PROGRAM/COMMENTS	LANG	ARITHMETIC
Constr Delaunay triang	`Triangle` [Sheb]	C	adaptive float
	Flipping algorithm. X or Postscript graphics, great documentation.		
Constr Delaunay triang	`CDT` [Lisb]	C	floating point
Triangulation	[NM], Graphics Gem [Pae95]	C	floating point
	Seidel's $O(n \log^* n)$ algorithm for simple polygons, extension for polygons with holes.		
Simple polygon trapez	`trapdec` [Gun]	C	floating point
	Seidel's algorithm for simple polygons.		
Arrangement trapez	`arrange` [Gol]	C	floating point or int
	Randomized incremental trapez of arrangement of polygons in the plane or on the sphere.		
Terrain triang	`Scape` [Gar]	C	floating point
	Greedy incremental algorithm for approximation of dense point data by a triangulated terrain.		

TABLE 52.3.2 Two-dimensional mesh generation.

PROGRAM	ALGORITHM/COMMENTS	LANG	ARITHMETIC
`Triangle` [Sheb]	Delaunay refinement	C	adaptive float
	Ruppert's algorithm. Specify min angle, element size.		
`GEOMPACK` [Joe]	convex decomposition, refinement	Fortran	floating point
	Specify number of elements.		
`tripoint` [Mit]	quad-tree	C++	floating point
	Mitchell's algorithm for meshes with no small angle. MATLAB interface.		
`SimLab` [Che]	Delaunay refinement	LISP	exact integer
	Chew's algorithm. Handles curved boundaries. Built on algebraic and topological computation system.		

52.4 OTHER PROGRAMS

Programs to compute intersections of polygons (Sections 23.3 and 33.4), or to test whether a point lies inside a polygon, are used in computer graphics and have mostly been written in that community. Another common operation in graphics is the computation of BSP trees; see [Wad] and Figure 47.1.4.

TABLE 52.3.3 Three-dimensional mesh generation.

PROGRAM	ALGORITHM	LANG	ARITHMETIC
`GEOMPACK` [Joe]	convex decomp, refinement	Fortran	floating point
`QMG` [Vav]	quad-tree	C++ and MATLAB	floating point

Computation of the medial axis of a polygon (Section 43.4 and Figure 47.1.5) or set of polylines is difficult, and the programs are comparatively recent. See Table 52.4.1.

Linear programming software is again mostly commercial; see [Gre] and Chapter 38 of this Handbook for more pointers. We list some computational geometry programs for low-dimensional linear programming and related LP-type problems in Table 52.4.2.

For point sets (Table 52.4.3), there are programs to compute alpha shapes (Section 43.4), a family of shapes based on the Delaunay triangulation, and a program to find a center point, which is a d-dimensional analogue of the median.

Graph drawing software is another area of both research and commercial software development. See the survey [DETT94], the on-line directory [San], and Chapter 44 of this Handbook. An experimental comparison appears in [DGL+95].

TABLE 52.4.1 Problems on polygons.

PROBLEM	PROGRAM/COMMENTS	LANG	ARITHMETIC
Point in polygon test	[Hai], Graphics Gem [Hec94]	C	floating point
Point in polygon test	[Wei], Graphics Gem [Hec94]	C	floating point
Point in polygon test	Comput. Geom. in C [O'R94]	C	floating point
Planar point location	`ploc` [Sei]	C	floating point
Boolean operations	`clippoly` [Schc]	C++	floating point
Medial axis	`PLVOR` [Ima]	Fortran	robust floating point
Approx medial axis	skeletonization software [Ogn]	C	floating point
	Package for binary images and complicated polygons. Great interface.		
Line segment inter	`LEDA` [N+]	C++	floating point
Visibility graph	`VisPak` [WPJ]	C++	floating point
	Collection including visibility graphs, visibility polygons and programs for axis-aligned polygons.		

RELATED CHAPTERS

Chapter 19: Convex hull computations
Chapter 20: Voronoi diagrams and Delaunay triangulations
Chapter 22: Triangulations
Chapter 23: Polygons
Chapter 35: Robust geometric computation

TABLE 52.4.2 LP-type problems.

PROGRAM	LANG/COMMENTS	ARITHMETIC
`linprog` [Hoh]	C	floating point
	Seidel's algorithm. Allows linear constraints and rational objective functions, so with ingenuity you can compute smallest enclosing ball, polytope separation distance, linear programming on a ball, etc.	
`ball` [Whi]	C++	floating point
	Sharir and Welzl's algorithm, applied to the problem of finding the smallest ball containing a family of balls.	
`lp` [Clac]	C	floating point
	Short implementations of Seidel's and Clarkson's algorithms.	

TABLE 52.4.3 Point set problems.

PROBLEM	PROGRAM	ARITHMETIC
Alpha shapes	Alpha shapes software [FEM+]	exact integer
	Package to generate, display, and compute volume and surface area of weighted or unweighted two and three-dimensional alpha shapes. Finds and measures holes, pockets, and voids. Robust, handles degeneracy. Well documented.	
Alpha shapes	`hull` [Clab]	adaptive float
	Computes alpha shapes in any dimension. Robust, handles degeneracy.	
Approx center point	center point program [Claa]	floating point
	Iterative Radon point algorithm of Clarkson, Eppstein, Miller, Sturtivant, and Teng.	
k-nearest neighbors	`Ranger` [Ski]	floating point
	Several space decomposition algorithms. Also does orthogonal range queries.	

Chapter 38: Linear programming in low dimensions
Chapter 43: Pattern recognition
Chapter 44: Graph drawing

REFERENCES

[AB95] D. Avis and D. Bremner. How good are convex hull algorithms? In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 20–28, 1995.

[ABD+] F. Avnaim, J.-D. Boissonnat, O. Devillers, F.P. Preparata, and M. Yvinec. Determinant sign software. `http://www.inria.fr/prisme/personnel/devillers/anglais/determinant.html`.

[ABD+95] F. Avnaim, J.-D. Boissonnat, O. Devillers, F.P. Preparata, and M. Yvinec. Evaluating signs of determinants using single precision arithmatic. Tech. Rep. 2306, INRIA, Sophia-Antipolis, 1995.

[ABS97] D. Avis, D. Bremner, and R. Seidel. How good are convex hull algorithms? *Comput. Geom. Theory Appl.*, 1997, to appear.

[Ame] N. Amenta. **Directory of Computational Geometry Software.** http://www.geom.umn.edu/locate/cglist.

[Avi] D. Avis. **lrs,qrs** and **rs.** ftp://mutt.cs.mcgill.ca/pub/C.

[BH] C.B. Barber and H. Huhdanpaa. **qhull.** http://www.geom.umn.edu/locate/qhull.

[CGA] CGAL project homepage. http://www.cs.ruu.nl/CGAL/.

[Che] L.P. Chew. Meshing code in **simlab.** http://www.cs.cornell.edu/Info/Projects/SimLab/releases/release-1-0.html.

[CL] T. Christof and A. Löbel. **Porta.** ftp://elib.zib-berlin.de/pub/mathprog/polyth/porta.

[Claa] K.L. Clarkson. Center point program. http://cm.bell-labs.com/who/clarkson/center.html.

[Clab] K.L. Clarkson. **Hull.** http://netlib.bell-labs.com/netlib/voronoi/hull.html.

[Clac] K.L. Clarkson. Linear programming programs. http://cm.bell-labs.com/who/clarkson/lp2.html.

[Clad] K.L. Clarkson. Two-dimensional convex hull program. http://cm.bell-labs.com/who/clarkson/2dch.c.

[Cla92] K.L. Clarkson. Safe and effective determinant evaluation. In *33d Annu. IEEE Sympos. Found. Comput. Sci.*, pages 387–395, 1992.

[DETT94] G. Di Battista, P. Eades, R. Tamassia, and I.G. Tollis. Algorithms for drawing graphs: An annotated bibliography. *Comput. Geom. Theory Appl.*, 4:235–282, 1994.

[Dev] O. Devillers. Delaunay tree program. http://www.inria.fr/prisme/logiciel/del-tree.html .

[DGL+95] G. Di Battista, A. Garg, G. Liotta, R. Tamassia, E. Tassinari, and F. Vargiu. An experimental comparison of three graph drawing algorithms. In *11th Annu. ACM Sympos. Comput. Geom.*, pages 306–315, 1995.

[dRJGM] P.J. de Rezende, W.R. Jacometti, C.N. Gon, and L.F. Morgado. **geolab.** ftp://geom.umn.edu/pub/contrib/comp_geom/geolab.

[EKK+] P. Epstein, J. Kavanagh, A. Knight, J. May, T. Nguyen, and J.-R. Sack. **Workbench for Computational Geometry.** ftp://ftp.ccs.carleton.ca/pub/workbench.

[Emia] I.Z. Emiris. **chD.** ftp://robotics.eecs.Berkeley.edu/pub/ConvexHull.

[Emib] I.Z. Emiris. Sparse resultant solver. http://www.inria.fr/safir/SAM/emiris/logiciels.alg-eng.html.

[FEM+] P. Fu, H. Edelsbrunner, E. Mücke, M. Facello, G. Bourhis, D. Setterdahl, D. Guoy, N. Akkiraju, J. Qian, and D. Nekhayev. Alpha shapes software. http://fiaker.ncsa.uiuc.edu/alpha.

[For] S. Fortune. **voronoi.** http://netlib.bell-labs.com/netlib/voronoi/index.html.

[For95] S. Fortune, *Voronoi diagrams and Delaunay triangulations*, In F.K. Hwang and D.-Z. Du, editors, *Computing in Euclidean Geometry*, 2nd edition, pages 225–265. World Scientific, Singapore, 1995.

[Fuk] K. Fukuda. `cdd`.
`ftp://ifor13.ethz.ch/pub/fukuda/cdd`.

[FV93] S. Fortune and C.J. Van Wyk. Efficient exact arithmatic for computational geometry. In *Proc. 9th Annu. ACM Sympos. Comput Geom.*, pages 163–172, 1993. *Software not released by AT&T.*

[Gar] M. Garland. `scape`.
`http://www.cs.cmu.edu/~garland/scape`.

[Gie] G.-J. Giezeman. `PLAGEO,SPAGEO`.
`ftp://archive.cs.ruu.nl/pub/SGI/GEO`.

[Gol] M. Goldwasser. `arrange`.
`ftp://flamingo.stanford.edu/pub/wass/arrangement`.

[Gre] J.W. Gregory. Linear programming FAQ.
`ftp://rtfm.mit.edu/pub/usenet/sci.answers/linear-programming-faq`.

[Gun] D. Gunopulos. `trapdec`.
`ftp://geom.umn.edu/pub/contrib/comp_geom/trapdec.tar.Z`.

[Hai] E. Haines. Code from "Point in polygon strategies". In [Hec94].

[Haz] C. Hazlewood. `tess`.
`ftp://geom.umn.edu/pub/contrib/comp_geom/tess.tar.Z`.

[Hec94] P. Heckbert, editor. *Graphics Gems IV*. Academic Press, Boston, 1994.
`ftp://ftp-graphics.stanford.edu/pub/Graphics/GraphicsGems`.

[Hin] A. Hines. Mesh generation survey.
`http://www-users.informatik.rwth-aachen.de/~roberts/hines.html`.

[Hoh] M. Hohmeyer. `linprog`.
`ftp://icemcfd.com/pub/linprog.a`.

[Ima] T. Imai. `PLVOR`.
`http://simplex.t.u-tokyo.ac.jp/~imai/software.html`.

[Joe] B. Joe. `GEOMPACK`.
`ftp://menaik.cs.ualberta.ca/pub/geompack`.

[Lisa] D. Lischinski. Code from "Incremental Delaunay triangulation". In [Hec94].

[Lisb] D. Lischinski. Constrained Delaunay triangulation program.
`http://www.cs.washington.edu/homes/danix/code/cdt.shar.Z`.

[Mit] S. Mitchell. `tripoint`.
`ftp://cs.cornell.edu/pub/scott/tri`.

[Müc] E. Mücke. `Detri`.
`http://www.geom.umn.edu/locate/cglist/GeomDir`.

[N+] S. Näher et al. `LEDA`.
`http://www.mpi-sb.mpg.de/LEDA/leda.html`.

[NM] A. Narkhede and D. Manocha. Code from "Fast polygon triangulation". In [Pae95].

[Ogn] R.L. Ogniewisz. Skeletonization software.
`http://hrl.harvard.edu/people/postdocs/rlo/rlo.dir/rlo-soft.html`.

[O'R94] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, 1994.
`ftp://grendel.csc.smith.edu/pub/compgeom`.

[Pae95] A. Paeth, editor. *Graphics Gems V*. Academic Press, Boston, 1995. ftp://ftp-graphics.stanford.edu/pub/Graphics/GraphicsGems.

[PCS] C. Montani, P. Cignoni, and R. Scopigno. `DeWall` and `InCoDe`. http://miles.cnuce.cnr.it/cg/swOnTheWeb.html.

[Reg] A. Rege. Toolkit for algebra and geometry. ftp://robotics.eecs.berkeley.edu/pub/rege.

[San] G. Sander. List of graph drawing tools. http://www.cs.uni-sb.de/RW/users/sander/html/gstools.html.

[Scha] R. Schneiders. Finite element mesh generation. http://www-users.informatik.rwth-aachen.de/~roberts/meshgeneration.-html#Software.

[Schb] P. Schorn. `XYZGeobench`. ftp://ftp.inf.ethz.ch/pub/xyz.

[Schc] K. Schutte. `clippoly`. http://www.ph.tn.tudelft.nl/People/klamer/clippoly_entry.html.

[SD95] P. Su and R.L.S. Drysdale. A comparison of sequential Delaunay triangulation algorithms. In *11th Annu. ACM Sympos. Comput. Geom.*, pages 61–70, 1995.

[Sei] R. Seidel. `ploc`. ftp://geom.umn.edu/pub/contrib/comp_geom/ploc.c.

[Sei97] R. Seidel. The nature and meaning of perturbations in geometric computing. *Discrete Comput. Geom.*, 1997, to appear.

[Shea] J.R. Shewchuck. Robust predicates. http://www.cs.cmu.edu/~quake/robust.html.

[Sheb] J.R. Shewchuck. `Triangle`. http://www.cs.cmu.edu/~quake/triangle.html.

[She96] J.R. Shewchuck. Triangle: Engineering a 2d quality mesh generator and Delaunay triangulator. In *Proc. 1st ACM Workshop on Applied Comput. Geom.*, volume 1148 of *Lecture Notes in Comput. Sci.*, pages 124–133. Springer-Verlag, Berlin, 1996.

[Ski] S. Skiena. `Ranger`. http://www.cs.sunysb.edu/~algorith/implement/ranger/implement.shtml.

[Vav] S. Vavasis. `QMG`. http://www.cs.cornell.edu/Info/People/vavasis/qmg-home.html.

[Wad] B. Wade. BSP tree frequently asked questions (FAQ). http://www.qualia.com/bspfaq/.

[Wei] K. Weiler. Code from "An incremental angle point in polygon test". In [Hec94].

[Whi] D. White. `ball`. http://vision.ucsd.edu/~dwhite/ball.html.

[WPJ] S. Wismath, H. Pinto, and L. Jackson. `VisPak`. http://www.cs.uleth.ca/~wismath/vis.html.

[WVHC] A. Woo, J. Volakis, G. Hulbert, and J. Case. Survey of volumetric grid generators. http://lovelace.nasa.gov/Parallel/People/woo/papers/gridgen/gridgen.html.

[YDO] C.K. Yap, T. Dube, and K. Ouchi. `REAL/EXPR`. http://simulation.nyu.edu/projects/exact.