INCLUDES CD WITH VBA CODE
AND EXCEL SPREADSHEETS

SECOND EDITION

THE COMPLETE GUIDE TO Option Pricing Formulas

ESPEN GAARDER HAUG, PhD

THE COMPLETE GUIDE TO

Option Pricing Formulas

SECOND EDITION

ESPEN GAARDER HAUG

McGraw-Hill
New York Chicago San Francisco
Lisbon London Madrid
Mexico City Milan Seoul New Delhi San Juan
Singapore Sydney Toronto

This book is for my mother and father, who gave me all I needed to write this book—a pen and a mind—the rest was discipline, hard work, and patience. We are all standing on the shoulders of our parents. With this book, hopefully I can give back a fraction of all the knowledge I have received from this miraculous journey called life. Life is full of options. I was given the option to write this book, and I exercised that option. Whether it was optimal for me to exercise it, only time can tell. You now have the option to read this book. Whether it is optimal for you to exercise that option is hard to say without a good option formula. This book is full of option formulas, so what are you waiting for?

Espen Gaarder Haug

CONTENTS

INTRODUCTION

Some people collect stamps; others collect coins, matchboxes, butterflies, or cars. I collect option pricing formulas. The book you have before you is a copy of this collection. As opposed to cars, one can easily share a collection of option pricing formulas with others. A collection like this would naturally not have been possible if it weren't for all the excellent researchers both in academia and in the industry who willingly share their knowledge in various publications.

Persons who collect stamps usually arrange their stamps under some kind of system—their issue year, what country they come from, and so on. I have organized my collection of option pricing formulas in a similar fashion. Each formula is given a reference and the year when it was first published.[1] With a few exceptions, I have also included a numerical example or a table with values for each option pricing formula. This should make it easier to understand the various option pricing models, as well as be of value to anyone who wants to check his or her computer implementation of an option pricing formula.

To better illustrate the use and implementation of option pricing formulas, I have included examples of programming codes for several of them. Programming codes for most of the formulas, together with ready-to-use spreadsheets, are included on the accompanying CD. Most of the formulas implemented also contains 3-D charts to illustrate the value or the various risk parameters of the option. By using this computer code in combination with the book, you should no longer view options pricing as a black box. The book differs from other texts on options pricing in the way I have tried to cut accompanying text to the bone. Text is included to illuminate the essence of implementing and applying the option pricing formulas. This should make it easy and efficient to find the formula you need, whether it is to close a multimillion-dollar options contract (without being ripped

[1] To the best given by my knowledge, that naturally is incomplete.

off) or to see if someone has already solved your problem of finding the value of some exotic options.

This collection of option pricing formulas is not intended as a textbook in option pricing theory but rather as a reference book for those who are already familiar with basic finance theory. However, if you think that a collection of option pricing formulas is useful only to theoreticians, you are wrong. The collection does not contain lengthy deductions of option pricing formulas[2] but rather the essence of options pricing. Most of these formulas are used daily by some of the best talent on Wall Street and by traders in financial centers worldwide. The collection of option pricing formulas is an ideal supplement for quants, quant traders, financial engineers, students taking courses in option pricing theory, or anyone else working with financial options.

The collection came out of my many years of work in option trading and derivatives research at the Den norske Bank, Chemical Bank, Tempus Financial Engineering, Paloma Partners, Amaranth Advisors, and J.P. Morgan Chase. Over many years, I collected everything I came across on the valuation of options. My collection of articles and books on the subject has increased every year. In order to see the large picture of the various option pricing models, as well as to avoid carrying around a heap of books and papers, I decided to compile the most central option pricing formulas into a book: *The Complete Guide to Option Pricing Formulas.*

Few, if any, financial markets have seen such explosive growth and new developments as the options markets. Continuously, new products are under development. With a few exceptions, I have chosen to collect option pricing models that can be used by practitioners. In a collection of formulas such as this, errors are particularly destructive. A great deal of effort has been put into minimizing typing errors. I hope that readers who find any remaining errors will call them to my attention so that they may be purged from any future editions. You can find my e-mail at www.espenhaug.com.

A table of all the option pricing formulas is included following the table of contents, giving an overview for easy reference. The table gives a short description of the key characteristics of all the option pricing formulas included in this book. If you are working with an option pricing problem, this table should be a natural starting point. Definitions of symbols are naturally important in a collection such as this. I have tried to define symbols in accordance with the modern literature. Currently, use of symbols in the theory of options pricing is far from standardized. Following the Option Pricing Overview is a Glossary of Notations, which you will find useful when using this collection.

---

[2] All material contains references to the originators, in case you wish to have all the details.

ACKNOWLEDGMENTS

Several individuals have helped improve the quality and completeness of the book. I appreciate the discussions, suggestions, and help contributed by Alexander Adamchuk, Ferdinando Ametrano, Grant Armstrong, Rainer Baule, Hans-Peter Bermin, Petter Bjerksund, Jeremy Bradley, Aaron Brown, Christine Brown, Don Chance, Tony Corso, Daniel Duffy, Christopher W. Edge, Steinar Ekern, Stein Eric Fleten, Stein Frydenberg, Teniel Gordon, Jørgen Haug, Alex Huang, Cho H. Hui, P. Kearney, Marisa Khan, Simon Launonen, Morten Lindeman, Lisa Majmin, Kristian Miltersen, Maurizio Mondello, Hicharn Mouline, Michael Ross, Fabrice Douglas Rouah, Gunnar Stensland, Erik Stettler, Svein Stokke, Robert Thoren, Jürgen Topper, Igor Tsatskis, Dan Tudball, Thijs van den Berg, Nico van der Wijst, Tobias Voigt, Sjur Westgaard, Zvi Wiener, W^2, Yuxing Yan—as well as, all the great people at the Wilmott forum.

A special thanks to some of the people who gave me considerable inputs to the second edition: David Bates, Jørgen Haug, Alireza Javaheri, Alan Lewis, William Margrabe, Silvian Mayer, Paul Wilmott, and Graeme West. I also had some fruitful discussions and received inspiration from the masters Bruno Dupire, Emanuel Derman, Nepse Guah, Nassim Taleb, and Peter Carr.

I would also like to thank some of the people in the derivatives industry with whom I worked closely together since the first edition and who kept inspiring me in my trading as well as my academic work, the original golden boys team at Paloma Partners Lauren Rose, Dr. Wu, Todor Delev, Stan Metelits (the Wizard), Patrick Morris Suzuki, and Tom Ku, Robert Jones and Nicolas Maounis. In J.P. Morgan, New York, I have been lucky to work as a trader in one of the most successful proprietary trading groups that at its last peak was best known as the dream team: Alex Gurevich, Ali Feroze, Andy Fell, Boutros Klink, Dean Williams, Even Berntsen, Gabriel Sod-Hoffs, Gregg Parr, John Stevenson, Joseph Choi, Lucky Sunder, Richard Tchenberdjian, Nanette Yang, Vinay Pande, and Vee Lung Phan. I would also like to thank all the people I have forgotten to thank.

It has been a pleasure to work with the people at McGraw-Hill. In particular, I'm grateful for many helpful suggestions and assistance from Stephen Isaacs and Kevin Thornton.

Enjoy the world of option formulas and option trading; must the most powerful formulas be on your side in good and bad times!

Espen Gaarder Haug

WHAT IS NEW IN THE SECOND EDITION?

If you already have the first edition, you probably wonder what is new in the second edition. Even before the first edition was published, I started working on the second edition. First, my plan was to make a few improvements and add a couple of formulas. But over the years the number of new formulas and improvements just took off. As you will see, the second edition contains more than double the amount of information and formulas as the first edition. In fact, writing the second edition involved more work than writing the first edition itself. In particular, the second edition contains much more coverage on option Greeks (option sensitivities), while the first edition just touched upon this topic, as have most other option textbooks. You'll find the second edition goes far beyond anything you have seen published on this topic. The number of new exotic options and derivatives that I have added to the second edition is astonishing. Just to mention a few: power options, powered options, log options, reset options, Margrabe barrier options, double barrier binary options, and dual double barrier options. The book also contains descriptions and implementations of stochastic volatility models, variance and volatility swaps, generalized jump diffusion models, and skewness and kurtosis models, among other nonstandard models and products.

With regard to numerical methods, I have added a lot of information in how to use tree models—for example, to value a lot of different complex exotic options. The second edition also covers finite difference methods and Monte Carlo simulation. The second edition also contains many new formulas and implementations for calculating volatility and risk parameters.

The accompanying CD-ROM contains much, much more than that provided with the first edition. In this version you can calculate almost any Greek for almost any option, as well as graph the value or the various option sensitivities in 3-D surface charts. It also contains sophisticated stochastic volatility models, as well as advanced Monte Carlo methods.

OPTIONS PRICING FORMULAS OVERVIEW

Politics is for the present, but an equation is for eternity.

Albert Einstein

The table on the next few pages offers an overview of the options pricing formulas presented in this book. For easy reference, each formula in the table is accompanied by a set of letters signifying key characteristics.

Type of formula:

- C: Closed-form solution.
- P: Closed-form approximation.
- N: Numerical method.
- E: European option.
- A: American option.

Type of underlying asset:

- S: Stock.
- I: Stock/index paying a dividend yield.
- F: Futures or forward contract.
- C: Currency.
- R: Interest rate or debt.

In the column "Computer code," a bullet (•) indicates that computer code for the formula is included in the book or on the accompanying CD-ROM.

Option Pricing Formula	Type of Formula	Type of Underlying Asset	Distribution of Underlying Asset	Computer Code	Short Description
Black–Scholes–Merton: Chapter 1					
Black–Scholes (1973)	C, E	S	Lognormal	•	The breakthrough in option pricing.
Merton (1973)	C, E	S, I	Lognormal	•	Extension of Black–Scholes formula including a dividend yield.
Black (1976)	C, E	F	Lognormal	•	Modified Black–Scholes for options on forward or futures.
Asay (1982)	C, E	F	Lognormal	•	Modified Black-76 for options that are fully margined.
Garman and Kohlhagen (1983) and Grabbe (1983)	C, E	C	Lognormal	•	Modified Black–Scholes for options on currencies.
Generalized Black–Scholes and Merton	C, E	S, I, F, C	Lognormal	•	Combines all the models above into one formula.
Black–Scholes–Merton on variance form	C, E	S, I, F, C	Lognormal	•	Use variance as input instead of standard deviation.
Bachelier (1900)	C, E	S	Normal	•	The first breakthrough in option valuation.
Sprenkle (1964)	C, E	S	Lognormal	•	Precursor to Black–Scholes–Merton.
Boness (1964)	C, E	S	Lognormal	•	Precursor to Black–Scholes–Merton.
Samuelson (1965)	C, E	S	Lognormal	•	Precursor to Black–Scholes–Merton, takes into account that the expected return for call option is greater than for stock.

Analytical Approximations for American Options: Chapter 3					
Barone-Adesi and Whaley (1987)	P, A	S, I, F, C	Lognormal	•	Approximation much used in practice.
Bjerksund and Stensland (1993)	P, A	S, I, F, C	Lognormal	•	Extremely computer efficient.
Bjerksund and Stensland (2002)	P, A	S, I, F, C	Lognormal	•	Accurate and efficient.
American Perpetual	C, A	S, I, F, C	Lognormal	•	Infinite time to maturity.
Exotic Options—Single Asset: Chapter 4					
Variable Purchase Options	C, E	S, I, F, C	Lognormal	•	Number of underlying shares deterministic function of the asset price.
Executive stock option	C, E	S, I	Lognormal	•	Take into account the probability that the executive will stay with the firm until the option expires.
Moneyness option	C, E	S, I, F, C	Lognormal	•	Premium in percent of forward, strike in percentage in- or out-of-the money.
Power contract	C, E	S, I, F, C	Lognormal	•	Contract where payoff is powered.
Standard power option (Asymmetric power option)	C, E	S, I, F, C	Lognormal	•	Option where asset is powered, gives high leverage.
Capped power option (Capped asymmetric option)	C, E	S, I, F, C	Lognormal	•	Option where asset is powered, but maximum payoff is capped.
Powered option (Symmetric power option)	C, E	S, I, F, C	Lognormal	•	Option where payoff is powered, gives high leverage.
Log contract	C, E	S, I, F, C	Lognormal	•	Contract where payoff is natural logarithm of asset. Building block in var and vol swaps.
Log option	C, E	S, I, F, C	Lognormal	•	Option where payoff is natural logarithm of asset.

(*continued*)

Option Pricing Formula	Type of Formula	Type of Underlying Asset	Distribution of Underlying Asset	Computer Code	Short Description
Forward start option	C, E	S, I, F, C	Lognormal	•	Starts at-the-money or proportionally in- or out-of-the-money after a known elapsed time into the future.
Fade-in option	C, E	S, I, F, C	Lognormal	•	Payoff weighted by how many fixings the asset inside a predefined range.
Ratchet option (Cliquet option)	C, E	S, I, F, C	Lognormal		A series of forward starting options.
Reset strike option	C, E	S, I, F, C	Lognormal	•	Strike is reset to the asset price at a predetermined future time.
Discrete time-switch option	C, E	S, I, F, C	Lognormal	•	Accumulates cash for every time unit the option is in-the-money.
Simple chooser option (as-you-like-it option)	C, E	S, I, F, C	Lognormal	•	Gives the right to choose between a call and put option.
Complex chooser option	C, E	S, I, F, C	Lognormal	•	Offers more flexibility than a simple chooser option.
Options on options (compound options)	C, E	S, I, F, C	Lognormal	•	Option on a plain vanilla option: call on call, call on put, put on call, and put on put.
Buyer-extendible option	C, E	S, I, F, C	Lognormal		Option that can be extended by the option holder.
Writer-extendible option	C, E	S, I, F, C	Lognormal	•	Option that will be extended by the writer if the option is out-of-the-money.
Floating strike lookback option (no-regrets option)	C, E	S, I, F, C	Lognormal	•	Options to sell at maximum or buy at minimum observed price.
Fixed strike lookback option (hindsight option)	C, E	S, I, C	Lognormal	•	An observed maximum or minimum asset price against a fixed strike.

Partial-time floating strike lookback option	C, E	S, I, C	Lognormal	•	Same as floating strike lookback except lookback monitoring only in parts of the option's lifetime.
Partial-time fixed strike lookback option	C, E	S, I, C	Lognormal	•	Same as fixed strike lookback except lookback monitoring only in parts of the option's lifetime.
Extreme spread option	C, E	S, I, C	Lognormal	•	Option on the difference between the observed maximum or minimum from two different time periods.
Mirror option	C, E	S, I, F, C	Lognormal	•	Options where holder can choose to mirror the path of the underlying asset.
Standard barrier option (inside barrier option)	C, E	S, I, F, C	Lognormal	•	Options where existence is dependent whenever the asset price hits a barrier level before expiration.
American barrier option	C, A	S, I, F, C	Lognormal		Same as above, but American style.
Double barrier option	C, E	S, I, F, C	Lognormal	•	Options with two barriers, one above and one below the current asset price.
Partial-time single-asset barrier option	C, E	S, I, F, C	Lognormal	•	Barrier hits are only monitored in a part of the options' lifetime.
Discrete barrier option	C, E	S, I, F, C	Lognormal	•	Adjustment that can be used for pricing barrier options with discrete barrier monitoring.
Look-barrier option	C, E	S, I, C	Lognormal	•	Combination of a partial time barrier option and a forward start fixed strike lookback option.
Soft-barrier option	C, E	S, I, C	Lognormal	•	The option has a barrier range and is knocked in or out partially.
First-then-barrier options	C, A	F	Lognormal		Dependent on lower and upper barrier.

(continued)

Option Pricing Formula	Type of Formula	Type of Underlying Asset	Distribution of Underlying Asset	Computer Code	Short Description
Double barrier option using symmetry	C, E	F	Lognormal		Options with two barriers, one above and one below the current asset price.
Dual double barrier option using symmetry	C, E	F	Lognormal		Gives call if hitting upper barrier, and put if lower barrier or vice versa.
Gap option (pay-later option)	C, E	S, I, F, C	Lognormal	•	One strike decides if the option is in or out-of-the-money; another strike decides the size of the payoff.
Cash-or-nothing option	C, E	S, I, F, C	Lognormal	•	Pays out cash if in-the-money and zero if out-of-the-money.
Asset-or-nothing option	C, E	S, I, F, C	Lognormal	•	Pays out asset if in-the-money; otherwise pays zero.
Supershare option	C, E	S, I, F, C	Lognormal	•	Pays out (Asset/Low strike) if the asset falls between a lower and higher strike.
Binary barrier options (digital option)	C, E	S, I, F, C	Lognormal	•	Can price 28 different binary barrier options.
Double Barrier Binary Options	C, E	S, I, F, C	Lognormal	•	Binary option with two barriers, one above and one below the current asset price.
Geometric average option (Asian option)	C, E	S, I, F, C	Lognormal	•	Option on a geometric average: $(x_1 \cdots x_n)^{1/n}$.
Arithmetic average option (Asian option)	P, A	S, I, F, C	Lognormal	•	Options on an arithmetic average: $(x_1 + \cdots + x_n)/n$.
Exotic Options on Two Assets: Chapter 5					
Relative outperformance option	C, E	S, I, F, C	Lognormal	•	Option on the relative performance of two assets.
Product option	C, E	S, I, F, C	Lognormal	•	Option on the product of two assets.
Two-asset correlation option	C, E	S, I, F, C	Lognormal	•	One asset decides if the option is in or out-of-the-money. Another asset with its own strike decides the payoff.

Exchange one asset for another option	C, E, A	S, I, F, C	Lognormal	•	Option to exchange one asset for another.
Exchange option on exchange option	C, E	S, I, F, C	Lognormal	•	Can be used to value sequential exchange opportunities.
Option on the maximum or minimum of two assets	C, E	S, I, F, C	Lognormal	•	Call or put options on the maximum or minimum of two assets.
Spread option	P, E	S, I, F, C	Lognormal	•	Option on the difference between two assets.
Two-asset barrier option (outside barrier option)	C, E	S, I, F, C	Lognormal	•	One asset decides barrier hits; the other asset decides payoff.
Partial-time two-asset barrier option	C, E	S, I, F, C	Lognormal	•	Barrier hits are only monitored in a part of the option's lifetime.
Margrabe barrier option	C, E	S, I, F, C	Lognormal		Barrier option on ratio of two assets.
Two-asset cash-or-nothing option	C, E	S, I, F, C	Lognormal	•	Two assets and two strikes decide if the option pays out a cash amount or nothing.
Best or worst cash-or-nothing option	C, E	S, I, F, C	Lognormal	•	Pays predefined cash amount depending on two asset.
Option on the minimum and maximum of two averages	C, E	S, I, F, C	Lognormal	•	Max-min option, but on two averages.
Foreign equity option stuck in domestic currency	C, E	S and C	Lognormal	•	Options on foreign equity in domestic currency.
Fixed exchange rate foreign equity option (Quantos)	C, E	S and C	Lognormal	•	Foreign equity option with fixed exchange rate.
Equity linked foreign exchange option	C, E	S and C	Lognormal	•	FX option where quantity depends on foreign equity price.
Takeover foreign exchange option	C, E	S and C	Lognormal	•	FX option that only can be exercised if takeover is successful.

(continued)

Option Pricing Formula	Type of Formula	Type of Underlying Asset	Distribution of Underlying Asset	Computer Code	Short Description
Chapter 6					
Settlement adjusted BSD	C, E	S, I, F, C	Lognormal	•	Black–Scholes adjusted for non-instant settlement.
French (1984) trading day adjusted	C, E	S, I, F, C	Lognormal	•	Black–Scholes adjusted for trading day volatility.
Wilmott (2000) discrete hedging	C, E	S, I, F, C	Lognormal		Black–Scholes adjusted for discrete hedging.
Leland (1985) transaction costs	C, E	S, I, F, C	Lognormal		Black–Scholes adjusted for hedging with transaction cost.
Lo and Wang (1995) trending markets	C, E	S, I, F, C	Lognormal	•	Black–Scholes adjusted for trending markets.
Hagan and Woodward (1999) (Cox and Ross (1976)) CEV	P, E	F	Constant elasticity of variance	•	Black–Scholes adjusted for constant elasticity of variance.
Jarrow and Rudd (1982) skewness and kurtosis	P, E	S, I, F, C	Various Non-specified	•	Black–Scholes adjusted for skewness and kurtosis.
Corrado and Su (1996) skewness and kurtosis	P, E	S, I, F, C	Various Non-specified	•	Black–Scholes adjusted for skewness and kurtosis.
Ray (1993) Pascal distribution	P, E	F	Pascal		Black–Scholes type model but with Pascal distribution.
Merton (1976) Jump Diffusion	C, E	S	Jump diffusion[1]	•	First jump-diffusion process model.
Bates (1991) Jump Diffusion	C, E	S	Jump diffusion[1]	•	Generalized jump-diffusion model.
Hull and White (1987) stochastic volatility	P, E	S, I, F, C	Stochastic volatility	•	Stochastic volatility model based on Taylor series.
Hull and White (1988) stochastic volatility	P, E	S, I, F, C	Stochastic volatility	•	Stochastic volatility model based on Taylor series.

SABR model stochastic volatility	P, E	F	Stochastic volatility	•	Practical and promising stochastic volatility model.
Variance swap	C		Stochastic volatility	•	Variance swap based on static hedging.
Volatility swap	C		Garch(1,1)	•	Volatility swap based on Garch(1,1) model.
Trees and Finite Difference Methods: Chapter 7					
Binomial trees	N, E, and A	S, I, F, C	Lognormal	•	Can be used to value most types of single asset options.
Barrier option in binomial trees	N, E, and A	S, I, F, C	Lognormal	•	A "standard" binomial tree where the number of time steps is adjusted so the barrier falls on the nodes.
Convertible bonds in binomial trees	N, E, and A	Stock and bond	Lognormal Stock	•	Convertible bond valuation with variable credit adjusted discount rate.
Trinomial trees	N, E, and A	S, I, F, C	Lognormal	•	More computer efficient and gives better flexibility than binomial trees.
Three-dimensional binomial trees	N, E, and A	S, I, F, C	Lognormal	•	Can be used for valuation of most options on two correlated assets.
Implied binomial trees	N, E, and A	S, I, F, C	Implied distribution from market data		Especially useful for valuation of exotic options consistent with more liquid plain vanilla European options.
Implied trinomial trees	N, E, and A	S, I, F, C	Implied distribution from market data	•	Offers more flexibility than implied binomial trees.
Explicit finite difference	N, E, and A	S, I, F, C	Lognormal	•	Can be used to value most types of single asset options.
Implicit finite difference	N, E, and A	S, I, F, C	Lognormal	•	Can be used to value most types of single asset options.
Crank-Nicholson method	N, E, and A	S, I, F, C	Lognormal	•	Can be used to value most types of single asset options.

(*continued*)

Option Pricing Formula	Type of Formula	Type of Underlying Asset	Distribution of Underlying Asset	Computer Code	Short Description
Monte Carlo Simulation: Chapter 8					
Standard Monte Carlo simulation	N, E	S, I, F, C, R	Dependent on the simulated process	•	Very flexible but relatively slow in computer time.
Antithetic Monte Carlo simulation	N, E	S, I, F, C, R	Dependent on the simulated process	•	Very flexible and more accurate than standard MC.
IQMC (Importance sampling)	N, E	S, I, F, C, R	Dependent on the simulated process	•	Very flexible and much faster than standard MC.
Quasi Random Monte Carlo simulation	N, E	S, I, F, C, R	Dependent on the simulated process	•	Very flexible and much faster than standard MC.
American Monte Carlo simulation	N, E, A	S, I, F, C, R	Dependent on the simulated process	•	Can be used for American options, but very slow in computer time.
Options on Stocks that Pay Discrete Dividends: Chapter 9					
Simple vol adjustment Chriss (1997)	C, E	S	Lognormal	•	The model is flawed and leads to arbitrage opportunities.
Haug and Haug (1998) Beneder and Vorst (2001) volatility adjustment.	C, E	S	Lognormal	•	The approximation is good for most cases, but can be inaccurate in some cases.
Bos, Gairat, and Shepeleva (2003) volatility adjustment.	C, E	S	Lognormal	•	The approximation is good for most cases, but can be inaccurate in some cases.

Bos and Vandermark (2002) volatility adjustment.	C, E	S	Lognormal	•	The approximation is one of the best but can be inaccurate in some special cases.
Roll–Geske–Whaley American call	C, A	S	"Lognormal"	•	The model is flawed and leads to arbitrage opportunities.
Non-recombining tree	N, E, A	S	Lognormal	•	Robust and accurate but slow and theoretically not very sound.
Haug, Haug, and Lewis (2003)	C, E, A	S	Various		Robust and accurate and theoretically sound; should be benchmark model.
Villiger (2005) discrete dividend yield	C, E	S	Lognormal	•	Closed form, computer efficient, and theoretically sound.
Recombining tree discrete dividend yield	N, E, A	S	Lognormal	•	Robust and accurate, computer efficient, and theoretically sound.
Commodity and Energy Options: Chapter 10					
Black-1976F adjusted	C, E	Forward	Lognormal	•	Black-76 adjusted for options on forwards expiring after the option.
Energy swaption	C, E	Swap(Forward)	Lognormal	•	Black-76 adjusted for options on commodity/energy swaps.
Miltersen and Schwartz (1998) commodity option model.	C, E	Forward	Lognormal[1]	•	Three-factor model with stochastic term structures of convenience yields and forward interest rates.
Pilipović (1997) seasonal	N, E, and A	S, I, C, F	Any	•	Seasonality adjustment that can be applied to Monte Carlo.

Option Pricing Formula	Type of Formula	Type of Underlying Asset	Distribution of Underlying Asset	Computer Code	Short Description
Interest Rate Derivatives: Chapter 11					
Black-76 for options on money market futures	C, E	Implied forward rates	Lognormal forward rates	•	Value call on futures as put on implied yield and vice versa.
Black-76 cap and floor model	C, E	Implied forward rates	Lognormal forward rates		A whole series of options on implied forward rates.
Modified Black-76 swaption model	C, E	Swap rate	Lognormal swap rate	•	Options on interest-rate swaps: payer and receiver swaptions.
Black-76 for options on bonds	C, E	Forward price of bond	Lognormal bond forward price	•	Often used when time to maturity on the option is short relative to the time to maturity on the underlying bond.
Schaefer and Schwartz (1987) adjusted Black–Scholes model	C, E	Bond price	Lognormal		Allows the price volatility of the bond to be a function of the bond duration.
Rendleman and Barter (1980)	N, E, and A	R	Lognormal interest rate		No arbitrage-free equilibrium model.
Vasicek (1977)	C, E, N, A	R	Normal interest rate mean reversion	•	No arbitrage-free equilibrium model.
Ho and Lee (1986)	C, E, N, A	R	Normal interest rate		Arbitrage-free with respect to underlying zero coupon rates.
Hull and White (1990)	C, E, N, A	R	Normal interest rate mean reversion		Arbitrage-free with respect to underlying zero coupon rates.
Black, Derman, and Toy (1990)	N, E, and A	R	Lognormal interest rate	•	Arbitrage-free with respect to underlying zero coupon rates.

[1] Lognormal futures price and normal distributed convenience yields and interest rates.

GLOSSARY OF NOTATIONS

The following list is in alphabetical order, ending with non-Latin symbols.

A	Accumulated amount in time-switch options.
b	Cost of carry rate (i.e., the cost of interest plus any additional costs). In every formula, it is continuously compounded.
BSM	Generalized Black-Scholes-Merton formula described in Chapter 1.
c	Price of European call option.
C	Price of American call option.
c_{BSM}	Call option value using the generalized Black-Scholes-Merton formula.
CEV	Constant elasticity of variance model.
CRR	Cox-Ross-Rubinstein binomial tree described in Chapter 7.
d	The size of the downward movement of the underlying asset in a binomial or trinomial tree.
D	Cash dividend.
E	Spot exchange rate of a currency.
f	Derivatives value, for example, an option.
F	Forward price or futures price.
GBM	Geometric Brownian motion.
H	Barrier (only used for barrier options).
K	Predetermined cash payoff.
L	Lower barrier in a barrier option.
$M(a, b; \rho)$	The cumulative bivariate normal distribution function described in Chapter 13.
n	Number of time steps in lattice or tree model.

$n(x)$	The standardized normal density function described in Chapter 13.
$N(x)$	The cumulative normal distribution function described in Chapter 13.
p_{BSM}	Put option value using the generalized Black-Scholes-Merton formula.
p	Price of European put option. Up probability in tree or lattice models.
P	Price of American put option, also used as bond price.
PDE	Partial differential equation.
q	Instantaneous proportional dividend yield rate of the underlying asset. Down probability in implied trinomial tree.
Q	Fixed quantity of asset.
r	Risk-free interest rate. In general, this is a continuously compounded rate. An exception is the Black–Derman–Toy model in Chapter 11 and some of the formulas in the section on interest rates in Chapter 11.
r_f	Foreign risk-free interest rate.
S	Price of underlying asset.
T	Time to expiration of an option or other derivative security in number of years.
u	The size of the up movement of the underlying asset in a binomial or trinomial tree.
U	Upper barrier in barrier option.
w	Value of European option.
W	Value of American option.
X	Strike price of option.
y	Bond or swap yield.
γ	Percentage of the total volatility explained by the jump in the jump-diffusion model.
Γ	Gamma of option.
δ	Discrete dividend yield.
Δ	Delta of option.
Δt	Size of time step in a tree model.
θ	Mean reversion level.
Θ	Theta of option.
κ	Speed of mean reversion ("gravity").
λ	Arrow-Debreu prices in the implied tree model. Expected number of jumps per year in the jump-diffusion model.

Λ	Elasticity of a plain vanilla European option (options sensitivity in percent with respect to a percent movement in the underlying asset).
μ	Drift of underlying asset (also used in other contexts).
ξ	Volatility of volatility in most stochastic volatility models.
π	The constant Pi ≈ 3.14159265359.
ρ	Correlation coefficient.
σ	Volatility of the relative price change of the underlying asset.
Φ	Phi of option.

CHAPTER

1

BLACK-SCHOLES-MERTON

Everything that can be counted does not necessarily count; everything that counts cannot necessarily be counted.

Albert Einstein

The first part of this chapter covers the Black-Scholes-Merton (BSM) formula and its close relatives. The last part offers a quick look at some of the most important precursors to the BSM model.

The BSM formula and its binomial counterpart may easily be the most used "probability model/tool" in everyday use — even if we consider all other scientific disciplines. Literally tens of thousands of people, including traders, market makers, and salespeople, use option formulas several times a day. Hardly any other area has seen such dramatic growth as the options and derivatives businesses. In this chapter we look at the various versions of the basic option formula. In 1997 Myron Scholes and Robert Merton were awarded the Nobel Prize (The Bank of Sweden Prize in Economic Sciences in Memory of Alfred Nobel). Unfortunately, Fischer Black died of cancer in 1995 before he also would have received the prize.

It is worth mentioning that it was not the option formula itself that Myron Scholes and Robert Merton were awarded the Nobel Prize for, the formula was actually already invented, but rather for the way they derived it — the replicating portfolio argument, continuous-time dynamic delta hedging, as well as making the formula consistent with the capital asset pricing model (CAPM). The continuous dynamic replication argument is unfortunately far from robust. The popularity among traders for using option formulas heavily relies on hedging options with options and on the top of this dynamic delta hedging, see Higgins (1902), Nelson (1904), Mello and Neuhaus (1998), Derman and Taleb (2005), as well as Haug (2006) for more details on this topic. In any case, this book is about option formulas and not so much about how to derive them.

1.1 BLACK-SCHOLES-MERTON

Provided here are the various versions of the Black-Scholes-Merton formula presented in the literature. All formulas in this section are originally derived based on the underlying asset S follows a geometric Brownian motion

$$dS = \mu S dt + \sigma S dz,$$

where μ is the expected instantaneous rate of return on the underlying asset, σ is the instantaneous volatility of the rate of return, and dz is a Wiener process.

1.1.1 The Black-Scholes Option Pricing Formula

The formula derived by Black and Scholes (1973) can be used to value a European option on a stock that does not pay dividends before the option's expiration date.[1] Letting c and p denote the price of European call and put options, respectively, the formula states that

$$c = SN(d_1) - Xe^{-rT}N(d_2) \tag{1.1}$$

$$p = Xe^{-rT}N(-d_2) - SN(-d_1), \tag{1.2}$$

where

$$d_1 = \frac{\ln(S/X) + (r + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = \frac{\ln(S/X) + (r - \sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T}$$

S = Stock price.
X = Strike price of option.
r = Risk-free interest rate.
T = Time to expiration in years.
σ = Volatility of the relative price change of the underlying stock price.
$N(x)$ = The cumulative normal distribution function, described in Chapter 13.

[1]The Black-Scholes formula can also be used to price American call options on a nondividend-paying stock, since it will never be optimal to exercise the option before expiration.

Example
Consider a European call option with three months to expiry. The stock price is 60, the strike price is 65, the risk-free interest rate is 8% per year, and the volatility is 30% per annum. Thus, $S = 60$, $X = 65$, $T = 0.25$, $r = 0.08$, $\sigma = 0.3$,

$$d_1 = \frac{\ln(60/65) + (0.08 + 0.3^2/2)0.25}{0.3\sqrt{0.25}} = -0.3253$$

$$d_2 = d_1 - 0.3\sqrt{0.25} = -0.4753$$

The value of the cumulative normal distribution $N(\cdot)$ can be found using the approximation function in Chapter 13:

$$N(d_1) = N(-0.3253) = 0.3725 \qquad N(d_2) = N(-0.4753) = 0.3173$$

$$c = 60N(d_1) - 65e^{-0.08\times 0.25}N(d_2) = 2.1334$$

Computer algorithm
The *BlackScholes*$(\cdot)$ function returns the call price if the *CallPutFlag* is set equal to "*c*" or the put price when set equal to "*p*." In the computer code $v = \sigma$.

```
Function BlackScholes(CallPutFlag As String, S As Double, X _
              As Double, T As Double, r As Double, v As Double) As Double

    Dim d1 As Double, d2 As Double

    d1 = (Log(S / X) + (r + v^2 / 2) * T) / (v * Sqr(T))
    d2 = d1 - v * Sqr(T)
    If CallPutFlag = "c" Then
        BlackScholes = S * CND(d1) - X * Exp(-r * T) * CND(d2)
    ElseIf CallPutFlag = "p" Then
        BlackScholes = X * Exp(-r * T) * CND(-d2) - S * CND(-d1)
    End If

End Function
```

where *CND*$(\cdot)$ is the cumulative normal distribution function described in Chapter 13. Example: *BlackScholes*("*c*", 60, 65, 0.25, 0.08, 0.3) will return a call price of 2.1334 as in the numerical example above.

Black-Scholes PDE
An alternative way to find the Black-Scholes option value is to solve the Black-Scholes partial differential equation (PDE). This can be done numerically using several different methods and is covered in Chapter 7. The PDE is given by

$$\left[\frac{\partial c}{\partial t} + \frac{1}{2}\frac{\partial^2 c}{\partial S^2}\sigma^2 S^2 + r\frac{\partial c}{\partial S}S\right]dt = rc$$

The Black-Scholes formula is a closed-form solution to this PDE given the payoff function (boundary condition) of a plain vanilla option.

1.1.2 Options on Stock Indexes

Merton (1973) extended the Black-Scholes model to allow for a continuous dividend yield, in addition to several other extensions. The model can be used to price European call and put options on a stock or stock index paying a known dividend yield equal to q:

$$c = Se^{-qT}N(d_1) - Xe^{-rT}N(d_2) \tag{1.3}$$

$$p = Xe^{-rT}N(-d_2) - Se^{-qT}N(-d_1), \tag{1.4}$$

where

$$d_1 = \frac{\ln(S/X) + (r - q + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = \frac{\ln(S/X) + (r - q - \sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T}$$

Example
Consider a European put option with six months to expiration. The stock index is 100, the strike price is 95, the risk-free interest rate is 10% per year, the dividend yield is 5% per year, and the volatility is 20% per year. $S = 100$, $X = 95$, $T = 0.5$, $r = 0.1$, $q = 0.05$, and $\sigma = 0.2$:

$$d_1 = \frac{\ln(100/95) + (0.1 - 0.05 + 0.2^2/2)0.5}{0.2\sqrt{0.5}} = 0.6102$$

$$d_2 = d_1 - 0.2\sqrt{0.5} = 0.4688$$

$$N(d_1) = N(0.6102) = 0.7291 \qquad N(d_2) = N(0.4688) = 0.6804$$

$$N(-d_1) = N(-0.6102) = 0.2709 \qquad N(-d_2) = N(-0.4688) = 0.3196$$

$$p = 95e^{-0.1\times0.5}N(-d_2) - 100e^{-0.05\times0.5}N(-d_1) = 2.4648$$

Merton PDE
The PDE behind the Merton formula is

$$\left[\frac{\partial c}{\partial t} + \frac{1}{2}\frac{\partial^2 c}{\partial S^2}\sigma^2 S^2 + (r - q)\frac{\partial c}{\partial S}S\right]dt = rc$$

1.1.3 Options on Futures

The formula of Black (1976) gives the price of European options when the underlying security is a forward or futures contract with initial price F:

$$c = e^{-rT}[FN(d_1) - XN(d_2)] \tag{1.5}$$

$$p = e^{-rT}[XN(-d_2) - FN(-d_1)], \tag{1.6}$$

where

$$d_1 = \frac{\ln(F/X) + (\sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = \frac{\ln(F/X) - (\sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T}$$

Example
Consider a European option on the Brent Blend futures with nine months to expiration. The futures price is USD 19, the strike price is USD 19, the risk-free interest rate is 10% per year, and the volatility is 28% per year. $F = 19$, $X = 19$, $T = 0.75$, $r = 0.1$, and $\sigma = 0.28$:

$$d_1 = \frac{\ln(19/19) + (0.28^2/2)0.75}{0.28\sqrt{0.75}} = 0.1212$$

$$d_2 = d_1 - 0.28\sqrt{0.75} = -0.1212$$

$$N(d_1) = N(0.1212) = 0.5483 \qquad N(d_2) = N(-0.1212) = 0.4517$$

$$N(-d_1) = N(-0.1212) = 0.4517 \qquad N(-d_2) = N(0.1212) = 0.5483$$

$$c = e^{-0.1\times 0.75}[19N(d_1) - 19N(d_2)] = 1.7011$$

$$p = e^{-0.1\times 0.75}[19N(-d_2) - 19N(-d_1)] = 1.7011$$

Black-76 PDE
The PDE behind the Black-76 formula is

$$\left[\frac{\partial c}{\partial t} + \frac{1}{2}\frac{\partial^2 c}{\partial F^2}\sigma^2 F^2\right] dt = rc$$

See Appendix A at the end of this chapter for more information on how to derive this PDE.

1.1.4 Margined Options on Futures

Asay (1982) has modified the Black-76 model for options on futures contracts where the premium is fully margined; see also Lieu (1990). The option premium is then paid into a margin account which accrues interest while the option is alive. Such contracts trade on, for example, the Sydney Futures Exchange. The Asay formula is like the Black-76 formula, but without the interest rate terms:

$$c = FN(d_1) - XN(d_2) \tag{1.7}$$

$$p = XN(-d_2) - FN(-d_1), \tag{1.8}$$

where

$$d_1 = \frac{\ln(F/X) + (\sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = \frac{\ln(F/X) - (\sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T}$$

Example
Consider a nine-month put option on a futures contract with price 4200 and volatility 15%, with strike 3800. What is the option value when the option premium is fully margined? $S = 4200$, $X = 3800$, $T = 0.75$, $\sigma = 0.15$:

$$d_1 = \frac{\ln(4200/3800) + (0.15^2/2)0.75}{0.15\sqrt{0.75}} = 0.8354$$

$$d_2 = d_1 - 0.15\sqrt{0.75} = 0.7055$$

$$N(-d_1) = N(-0.8354) = 0.2017, \qquad N(-d_2) = N(-0.7055) = 0.2403$$

$$p = 3800N(-d_2) - 4200N(-d_1) = 65.6185$$

Asay PDE
The PDE behind the Asay formula is

$$\left[\frac{\partial c}{\partial t} + \frac{1}{2}\frac{\partial^2 c}{\partial F^2}\sigma^2 F^2\right] dt = c$$

1.1.5 Currency Options

The Garman and Kohlhagen (1983) modified Black-Scholes model can be used to price European currency options; see also Grabbe (1983). The model is mathematically equivalent to the Merton (1973) model presented earlier. The only difference is that the dividend yield is replaced by the risk-free rate of the foreign currency r_f:

$$c = Se^{-r_f T} N(d_1) - Xe^{-rT} N(d_2) \tag{1.9}$$

$$p = Xe^{-rT} N(-d_2) - Se^{-r_f T} N(-d_1), \tag{1.10}$$

where

$$d_1 = \frac{\ln(S/X) + (r - r_f + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = \frac{\ln(S/X) + (r - r_f - \sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T}$$

For more information on currency options, see DeRosa (2000).

Example
Consider a European USD-call/EUR-put option with six months to expiration. The USD/EUR exchange rate is 1.56, the strike is 1.6, the domestic risk-free interest rate in EUR is 8% per year, the foreign risk-free interest rate in the United States is 6% per year, and the volatility is 12% per year. $S = 1.56, X = 1.6, T = 0.5, r = 0.06, r_f = 0.08, \sigma = 0.12$.

$$d_1 = \frac{\ln(1.56/1.6) + (0.06 - 0.08 + 0.12^2/2)0.5}{0.12\sqrt{0.5}} = -0.3738$$

$$d_2 = d_1 - 0.12\sqrt{0.5} = -0.4587$$

$$N(d_1) = N(-0.3738) = 0.3543 \qquad N(d_2) = N(-0.4587) = 0.3232$$

$$c = 1.56e^{-0.08\times 0.5}N(d_1) - 1.6e^{-0.06\times 0.5}N(d_2) = 0.0291$$

The option premium is thus 0.0291 USD per EUR. Alternatively, the premium can be quoted in EUR per USD $0.0291/1.56^2 = 0.0120$—or as percent of spot, $0.0291/1.56 = 0.0186538$, or 1.8654% of EUR (or the spot price). Hence, if the option has a notional of 100 million EUR, the total option premium is 1,865,384.62 EUR, or $1{,}865{,}384.62 \times 1.56 = 2{,}910{,}000.00$ in USD.

Currency Option PDE
The partial differential equation behind the Garman and Kohlhagen formula is

$$\left[\frac{\partial c}{\partial t} + \frac{1}{2}\frac{\partial^2 c}{\partial S^2}\sigma^2 S^2 + (r - r_f)\frac{\partial c}{\partial S}S\right]dt = rc$$

1.1.6 The Generalized Black-Scholes-Merton Option Pricing Formula

The Black-Scholes-Merton model can be "generalized" by incorporating a cost-of-carry rate b. This model can be used to price European options on stocks, stocks paying a continuous dividend yield, options on futures, and currency options:

$$c_{BSM} = Se^{(b-r)T}N(d_1) - Xe^{-rT}N(d_2) \tag{1.11}$$

$$p_{BSM} = Xe^{-rT}N(-d_2) - Se^{(b-r)T}N(-d_1), \tag{1.12}$$

where

$$d_1 = \frac{\ln(S/X) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

$b = r$	gives the Black and Scholes (1973) stock option model.
$b = r - q$	gives the Merton (1973) stock option model with continuous dividend yield q.
$b = 0$	gives the Black (1976) futures option model.
$b = 0$ and $r = 0$	gives the Asay (1982) margined futures option model.
$b = r - r_f$	gives the Garman and Kohlhagen (1983) currency option model.

Computer algorithm

The *GBlackScholes*(·) function returns the call price if the *CallPutFlag* is set equal to "*c*" or the put price when set equal to "*p*."

```
Function GBlackScholes(CallPutFlag As String, S As Double, X As Double, _
                T As Double, r As Double, b As Double, v As Double) As Do

   Dim d1 As Double, d2 As Double
   d1 = (Log(S / X) + (b + v^2 / 2) * T) / (v * Sqr(T))
   d2 = d1 - v * Sqr(T)

   If CallPutFlag = "c" Then
       GBlackScholes = S * Exp((b - r) * T) * CND(d1) - X * Exp(-r * T) _
       * CND(d2)
   ElseIf CallPutFlag = "p" Then
       GBlackScholes = X * Exp(-r * T) * CND(-d2) - S * Exp((b - r) * T) _
       * CND(-d1)
   End If

End Function
```

where *CND*(·) is the cumulative normal distribution function described in Chapter 13. Example: *GBlackScholes*("*p*," 75, 70, 0.5, 0.1, 0.05, 0.35) returns a put price of 4.0870.

Generalized Black-Scholes-Merton PDE

The Black-Scholes-Merton (BSM) PDE in terms of asset price S

$$\left[\frac{\partial c}{\partial t} + \frac{1}{2}\frac{\partial^2 c}{\partial S^2}\sigma^2 S^2 + b\frac{\partial c}{\partial S}S\right]dt = rc$$

Solving the PDE with the appropriate boundary condition yields the BSM formula. An alternative is to solve the PDE numerically. This method is slower but more flexible. It is covered in Chapter 7. See Appendix A at the end of this chapter for more information about how to derive this PDE.

One can alternatively rewrite this PDE in terms of $\ln(S)$. Letting $x = \ln(S)$ yields

$$\frac{\partial c}{\partial t} + \frac{1}{2}\sigma^2\frac{\partial^2 c}{\partial x^2} + (b - \sigma^2/2)\frac{\partial c}{\partial x} = rc$$

In Chapter 7 we will also look at how to solve this PDE using finite difference methods.

1.2 PARITIES AND SYMMETRIES

This section presents several useful parity and symmetry relationships for the BSM formula.

1.2.1 Put-Call Parity for European Options

The put-call parity described in detail by Higgins (1902) and Nelson (1904) gives the value of a put option with the same strike price, and vice versa. An arbitrage opportunity exists if the parity does not hold. This is based on several assumptions—for instance, that we can easily short the underlying asset, no bid-ask spreads, and no transaction costs. It does not, however, rely on any assumptions about the distribution of the price of the underlying security.

Stock Options

$$c = p + S - Xe^{-rT}, \qquad p = c - S + Xe^{-rT} \tag{1.13}$$

Example

Consider a European call option on a nondividend-paying stock with a time to maturity of six months. The stock price is 100, the strike price is 105, the risk-free rate is 10% per year, and the call value is 8.5. What is the value of a put with the same parameters? $S = 100$, $X = 105$, $T = 0.5$, $r = 0.1$, $c = 8.5$.

$$p = 8.5 - 100 + 105e^{-0.1\times 0.5} = 8.3791$$

Options on a Stock Paying Continuous Dividend Yield

$$c = p + Se^{-qT} - Xe^{-rT}, \qquad p = c - Se^{-qT} + Xe^{-rT} \tag{1.14}$$

Option on Futures

$$c = p + (F - X)e^{-rT}, \qquad p = c - (F - X)e^{-rT} \tag{1.15}$$

Fully Margined Options on Futures

$$c = p + F - X, \qquad p = c - F + X \tag{1.16}$$

Currency Option

$$c = p + Se^{-r_f T} - Xe^{-rT} \qquad p = c - Se^{-r_f T} + Xe^{-rT} \tag{1.17}$$

The Put-Call Parity for the Generalized Black-Scholes Formula

$$c = p + Se^{(b-r)T} - Xe^{-rT} \qquad p = c - Se^{(b-r)T} + Xe^{-rT} \tag{1.18}$$

where b is the cost-of-carry of the underlying security.

$b = r$ = Cost-of-carry on a nondividend-paying stock.
$b = r - q$ = Cost-of-carry on a stock that pays a continuous dividend yield equal to q.
$b = 0$ = Cost-of-carry on a future contract.
$b = 0$ and $r = 0$ Gives the Asay (1982) margined futures option model.
$b = r - r_f$ = Cost-of-carry on a currency position.

1.2.2 At-the-Money Forward Value Symmetry

Put and call options will have value symmetry (identical values) when they are at-the-money forward Nelson (1904), defined as

$$Se^{bT} = X \qquad \text{or} \qquad S = Xe^{-bT}$$

At this strike, put and call options will also have rho and theta symmetry, but not delta symmetry (see next chapter). The result is naturally based on using the same volatility for call and put options. Put-call parity will normally ensure this symmetry, but it may not hold in markets where there are restrictions on short selling, or other market imperfections.

1.2.3 Put-Call Symmetry

There is also a put-call value symmetry for puts and calls with different strikes, first described by Bates (1991), and in more detail by Carr (1994) and Carr and Bowie (1994)

$$c(S, X, T, r, b, \sigma) = \frac{X}{Se^{bT}} p\left(S, \frac{(Se^{bT})^2}{X}, T, r, b, \sigma\right) \tag{1.19}$$

A call with strike X is thus equivalent to $\frac{X}{Se^{bT}}$ puts with strike $\frac{(Se^{bT})^2}{X}$. This symmetry is useful for hedging and pricing barrier options, as shown in Chapter 4.

1.2.4 Put-Call Supersymmetry

Another useful symmetry between call and put options is given by

$$c(S, X, T, r, b, \sigma) = p(-S, -X, T, r, b, -\sigma)$$

—that is, inputting negative asset price, strike, and volatility for a put option yields the same value as a call option. Consider next the following state space transformation:

$$k \times c(S, X, T, r, b, \sigma) = c(k \times S, k \times X, T, r, b, \sigma),$$

which together with the above symmetry implies the put-call supersymmetry

$$c(S, X, T, r, b, \sigma) = -p(S, X, T, r, b, -\sigma), \qquad (1.20)$$

and naturally

$$p(S, X, T, r, b, \sigma) = -c(S, X, T, r, b, -\sigma) \qquad (1.21)$$

The value of a call is thus equal to the value of minus a put with negative volatility, and vice versa. The result simplifies coding and implementation of many option calculations. There is no longer a need to develop or to code separate formulas for put and call options.

The put-call supersymmetry can be extended to many exotic options and holds also for American options. See Adamchuk (1998), Peskir and Shiryaev (2001), Haug (2002), and Aase (2004) for more details on supersymmetry as well as a discussion on negative volatility.

1.2.5 Black-Scholes-Merton on Variance Form

In some circumstances, it is useful to rewrite the BSM formula using variance as input instead of volatility, $V = \sigma^2$:

$$c = Se^{(b-r)T}N(d_1) - Xe^{-rT}N(d_2) \qquad (1.22)$$

$$p = Xe^{-rT}N(-d_2) - Se^{(b-r)T}N(-d_1), \qquad (1.23)$$

where

$$d_1 = \frac{\ln(S/X) + (b + V/2)T}{\sqrt{VT}}$$

$$d_2 = d_1 - \sqrt{VT}$$

BSM on variance form clearly gives the same price as when written on volatility form. The variance form is used indirectly in terms of its partial derivatives in some stochastic variance models, as well as for hedging of variance swaps, described in Chapter 6. The BSM on variance form moreover admits an interesting symmetry between put and call options as discussed by Adamchuk and Haug (2005) at www.wilmott.com.

$$c(S, X, T, r, b, V) = -c(-S, -X, -T, -r, -b, -V) \quad (1.24)$$

and

$$p(S, X, T, r, b, V) = -p(-S, -X, -T, -r, -b, -V) \quad (1.25)$$

It is possible to find several similar symmetries if we introduce imaginary numbers.

1.3 BEFORE BLACK-SCHOLES-MERTON

The curious reader may be asking how people priced options before the BSM breakthrough was published in 1973. This section offers a quick overview of some of the most important precursors to the BSM model.

1.3.1 The Bachelier Model

As early as 1900, Louis Bachelier published his now famous work on option pricing. In contrast to Black, Scholes, and Merton, Bachelier assumed a normal distribution for the asset price—in other words, an arithmetic Brownian motion process

$$dS = \sigma dz$$

where S is the asset price and dz is a Winer process. This implies a positive probability for observing a negative asset price—a feature that is not popular for stocks and any other asset with limited liability features.

The current call price is the expected price at expiration. This argument yields

$$c = (S - X)N(d_1) + \sigma\sqrt{T}n(d_1), \quad (1.26)$$

and for a put option we get

$$p = (X - S)N(-d_1) + \sigma\sqrt{T}n(d_1), \tag{1.27}$$

where

$$d_1 = \frac{S - X}{\sigma\sqrt{T}}$$

S = Stock price.
X = Strike price of option.
T = Time to expiration in years.
σ = Volatility of the underlying asset price
$N(x)$ = The cumulative normal distribution function, described in Chapter 13.
$n(x)$ = The standard normal density function.

At-the-Money Approximation
In case the option is at-the-money, $S = X$, a good approximation for the Bachelier formula is given by

$$c = p \approx \sigma\sqrt{\frac{T}{2\pi}} \approx \sigma 0.4\sqrt{T}$$

Modified Bachelier Model
By using the arguments of BSM but now with arithmetic Brownian motion (normal distributed stock price), we can easily correct the Bachelier model to take into account the time value of money in a risk-neutral world. This yields

$$c = SN(d_1) - Xe^{-rT}N(d_1) + \sigma\sqrt{T}n(d_1) \tag{1.28}$$

$$p = Xe^{-rT}N(-d_1) - SN(-d_1) + \sigma\sqrt{T}n(d_1) \tag{1.29}$$

$$d_1 = \frac{S - X}{\sigma\sqrt{T}}$$

1.3.2 The Sprenkle Model

Sprenkle (1964) assumed the stock price was lognormally distributed and thus that the asset price followed a geometric Brownian motion, just as in the Black and Scholes (1973) analysis. In this way he ruled out the possibility of negative stock prices, consistent with limited liability. Sprenkle moreover allowed for a drift in the asset price, thus allowing positive interest rates and risk aversion (Smith, 1976). Sprenkle assumed today's value was equal to the expected value at maturity.

$$c = Se^{\rho T} N(d_1) - (1-k) X N(d_2) \tag{1.30}$$
$$d_1 = \frac{\ln(S/X) + (\rho + \sigma^2/2)T}{\sigma\sqrt{T}}$$
$$d_2 = d_1 - \sigma\sqrt{T},$$

where ρ is the average rate of growth of the asset price and k is the adjustment for the degree of market risk aversion.

1.3.3 The Boness Model

Boness (1964) assumed a lognormal asset price. Boness derives the following value for a call option:

$$c = SN(d_1) - Xe^{-\rho T} N(d_2) \tag{1.31}$$
$$d_1 = \frac{\ln(S/X) + (\rho + \sigma^2/2)T}{\sigma\sqrt{T}}$$
$$d_2 = d_1 - \sigma\sqrt{T},$$

where ρ is the expected rate of return to the asset.

1.3.4 The Samuelson Model

Samuelson (1965; see also Smith, 1976) assumed the asset price follows a geometric Brownian motion with positive drift, ρ. In this way he allowed for positive interest rates and a risk premium.

$$c = Se^{(\rho - w)T} N(d_1) - Xe^{-wT} N(d_2) \tag{1.32}$$
$$d_1 = \frac{\ln(S/X) + (\rho + \sigma^2/2)T}{\sigma\sqrt{T}}$$
$$d_2 = d_1 - \sigma\sqrt{T},$$

where ρ is the average rate of growth of the share price and w is the average rate of growth in the value of the call. This is different from the Boness model in that the Samuelson model can take into account that the expected return from the option is larger than that of the underlying asset $w > \rho$.

1.4 APPENDIX A: THE BLACK-SCHOLES-MERTON PDE

1.4.1 Ito's Lemma

Ito's lemma basically states that if an asset price follows the Ito process

$$dS = \mu(S,t)dt + \sigma(S,t)dz,$$

then a derivative security f that is a function of S and time t must follow the Ito process

$$df = \left(\frac{\partial f}{\partial S}\mu + \frac{\partial f}{\partial t} + \frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2\right) + \frac{\partial f}{\partial S}\sigma dz$$

This Ito process has a drift equal to $\frac{\partial f}{\partial S}\mu + \frac{\partial f}{\partial t} + \frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2$ and a dispersion equal to $\frac{\partial f}{\partial S}\sigma dz$. Ito's lemma is a powerful tool to value derivative securities, for example, by helping us find a BSM partial differential equation (PDE) that can solved for the price, with the appropriate boundary conditions.

Ito's Lemma and the Black-Scholes-Merton Option Model
A call must be a function of the process of the underlying asset price S. Assuming that the asset price follows a geometric Brownian motion,

$$dS = \mu S dt + \sigma S dz \tag{1.33}$$

$$dc = \left[\frac{\partial c}{\partial t} + \frac{\partial c}{\partial S}\mu S + \frac{1}{2}\frac{\partial^2 c}{\partial S^2}\sigma^2 S^2\right]dt + \frac{\partial c}{\partial S}\sigma S dz \tag{1.34}$$

The objective is to eliminate all risk with an combination of $\frac{\partial c}{\partial S}$ in the asset and short one option.

The value of the portfolio:

$$V = -c + \frac{\partial c}{\partial S}S$$

The change in the portfolio value becomes:

$$dV = -dc + \frac{\partial c}{\partial S}dS \tag{1.35}$$

Substituting (1.33) and (1.34) into (1.35):

$$\begin{aligned} dV &= -\left[\frac{\partial c}{\partial t} + \frac{\partial c}{\partial S}\mu S + \frac{1}{2}\frac{\partial^2 c}{\partial S^2}\sigma^2 S^2\right] dt - \frac{\partial c}{\partial S}\sigma S dz \\ &\quad + \frac{\partial c}{\partial S}(\mu S dt + \sigma S dz) \\ &= -\frac{\partial c}{\partial t}dt - \frac{\partial c}{\partial S}\mu S dt - \frac{1}{2}\frac{\partial^2 c}{\partial S^2}\sigma^2 S^2 dt - \frac{\partial c}{\partial S}\sigma S dz \\ &\quad + \frac{\partial c}{\partial S}\mu S dt + \frac{\partial c}{\partial S}\sigma S dz \\ &= -\frac{\partial c}{\partial t}dt - \frac{1}{2}\frac{\partial^2 c}{\partial S^2}\sigma^2 S^2 dt \end{aligned}$$

This equation does not include the stochastic Winer process dz. In time dt, the portfolio earns capital gains dV and income from the underlying assets S equal to

$$(r - b)\frac{\partial c}{\partial S} S dt$$

Change in the wealth of the portfolio dW in time dt is

$$dW = -\frac{\partial c}{\partial t}dt - \frac{1}{2}\frac{\partial^2 c}{\partial S^2}\sigma^2 S^2 dt + (r - b)\frac{\partial c}{\partial S} S dt$$

thus dW is risk-free over dt:

$$dW = rV dt$$

$$-\frac{\partial c}{\partial t}dt - \frac{1}{2}\frac{\partial^2 c}{\partial S^2}\sigma^2 S^2 dt + (r - b)\frac{\partial c}{\partial S} S dt = r\left(-c + \frac{\partial c}{\partial S} S\right) dt$$

$$\frac{\partial c}{\partial t}dt + \frac{1}{2}\frac{\partial^2 c}{\partial S^2}\sigma^2 S^2 dt + b\frac{\partial c}{\partial S} S dt = rc$$

$$\left[\frac{\partial c}{\partial t} + \frac{1}{2}\frac{\partial^2 c}{\partial S^2}\sigma^2 S^2 + b\frac{\partial c}{\partial S} S\right] dt = rc$$

The BSM call option pricing formula is found by using the boundary condition at option expiration:

$$c = \max(S - X; 0).$$

1.4.2 Dynamic Hedging

The idea of market neutral static delta hedging goes all the way back to Higgins (1902) and Nelson (1904). The idea was later extended by Thorp and Kaussof (1967) and Thorp (1969) and was further extended to continuous time dynamic delta hedging by Black and Scholes (1973) and Merton (1973). Continuous time delta hedging as just described is removing all risk all the time under some strict theoretical assumptions. Unfortunately continuous dynamic delta hedging is far from robust in practice. Dynamic delta hedging removes a lot of risk compared to not hedging or even static delta hedging. Unfortunately, options are extremely risky instruments, and even after removing a lot of risk there is more than plenty of risk left. That is, in practice dynamic delta hedging alone cannot be used as an argument for risk-neutral valuation. See Haug (2006) for more references and detailed discussion on this topic. We will here just briefly discuss a more robust alternative to the dynamic hedging argument, namely, the Derman-Taleb method.

Derman-Taleb Method

Derman and Taleb (2005) describes an interesting way to come up with the Black-Scholes-Merton formula, but without relaying on dynamic delta hedging and neither on the Capital Asset Pricing Model (CAPM). They are starting out by the valuation methods used before Black-Scholes-Merton, simply by discounting the expected pay off from an option based on an assumption on the distribution of the underlying asset at maturity. Assuming the underlying stock is log-normal distributed we get

$$\begin{aligned} c &= e^{-RT}(E[S-X]_+) \\ &= e^{-RT}[Se^{\mu T}N(d_1) - XN(d_2)], \end{aligned} \tag{1.36}$$

$$d_1 = \frac{\ln(S/X) + (\mu + \sigma^2/2)T}{\sigma\sqrt{T}}, d_2 = d_1 - \sigma\sqrt{T},$$

and for a put we have

$$\begin{aligned} p &= e^{-RT}(E[X-S]_+) \\ &= e^{-RT}[XN(-d_2) - Se^{\mu T}N(-d_1)], \end{aligned} \tag{1.37}$$

where μ is the average rate of growth of the share price over the lifetime of the option. R is the discount rate including a unknown risk premium.

Next assuming the forward price is strictly based on arbitrage pricing. Based on this we know the forward price of a stock not is dependent on the real expected drift in the stock, but simply on the risk-free rate and naturally dividend, but here we for simplicity skip dividend even if the conclusions would be the same. The current value of the forward price with delivery price X must then be

$$F_V = S - Xe^{-rT}$$

where r is the risk-free zero coupon rate covering the period from now until the expiration of the forward contract T. Second Derman-Taleb takes advantage of the put-call parity, see Nelson (1904):

$$F_V = c - p$$

That is a forward expiring at the same time as the option must be equal to a long European call, c, plus a short put p. Combining equation 1.36 with 1.37 we get

$$F_V = c - p = e^{-RT}[Se^{\mu T} - X]$$

To avoid arbitrage opportunities both μ and R must be set equal to the risk free rate r. Another way to think about this is that we now have removed all risk in the option and that we, therefore, can use risk-neutral valuation of the options, we end up with

$$c = SN(d_1) - Xe^{-rT}N(d_2),$$

$$d_1 = \frac{\ln(S/X) + (r + \sigma^2/2)T}{\sigma\sqrt{T}},$$

$$d_2 = d_1 - \sigma\sqrt{T},$$

This is identical to the Black and Scholes (1973) and Merton 1973 formula. The formula is no longer dependent on dynamic delta hedging, neither directly on the CAPM formula. The method is based on a pure arbitrage argument and is extremely robust and is fully consistent with continuous-time as well as discrete-time trading.

This method is also the simplest method consistent with the volatility smile, see Haug (2006). The Derman-Taleb method is in this respect not a model that directly describes the stochastic process of the underlying asset or the dynamics of the volatility. It is a pure arbitrage argument as well as relative value arbitrage argument that actually is extremely robust and basically is how options traders operate by hedging away most option risk with other options, and on top of this are using dynamic delta hedging. A consequence of this argument as discussed by Haug (2006) is that option valuation also must be dependent on supply and demand of options. There is still a link towards the underlying asset, but this link is only of the weak form, and not of the strong form assumed by Black, Scholes and Merton where all derivatives can be created synthetically without taking risk. In other words, option traders need to take into account both the dynamic process of the underlying asset as well as the supply and demand for options when valuing options. This is actually the way option trader's use option formulas.

Forward Price

The forward price of a financial asset and also many commodities are purely based on arbitrage and not on expectations, for an early indication on this see Keynes (1924) and Blau (1944–1945). The fair initial contract price F that makes the initial value of a forward contract zero is

$$F = Se^{bT}$$

The fair value of a forward contract just at the time of initiating is zero, but then after initiated the forward contract the value is now naturally related to the difference between the current spot price and the agreed upon forward price F_A. F_A was initially equal to F, the fair arbitrage value of a forward contract after initialized must be

$$F_V = e^{-rT}[Se^{bT} - F_A]$$

That is simply the discounted difference between the current spot price and the agreed upon forward price, but taking into account the cost-of-carry of holding the underlying asset.

CHAPTER

2

BLACK-SCHOLES-MERTON GREEKS

The next step was automatic for a trained scientist: analyze the relation between the price of the warrant and the price of its associated common stock. Find the rules, or 'laws,' connecting the two prices.

Edward Thorp

The options sensitivities (Greeks) below are the partial derivatives of the Black-Scholes-Merton (BSM) formula introduced in Chapter 1. The partial derivative is a measure of the sensitivity of the option price to a small change in a parameter of the formula. Appendix B at the end of this chapter contains more details on how to derive the partial derivatives. The second part of this chapter covers how to numerically compute option sensitivities. A large part of this chapter is based on Haug (2003).

For a quick reference, many of the option sensitivities described in this chapter are listed in Table 2-1.

2.1 DELTA GREEKS

2.1.1 Delta

Delta is the option's sensitivity to small changes in the underlying asset price.

Call

$$\Delta_{\text{call}} = \frac{\partial c}{\partial S} = e^{(b-r)T} N(d_1) > 0 \tag{2.1}$$

Put

$$\Delta_{\text{put}} = \frac{\partial p}{\partial S} = e^{(b-r)T} [N(d_1) - 1] < 0 \tag{2.2}$$

Figure 1 illustrates the delta of a call option for varying asset prices and times to maturity. Figure 2 similarly illustrates the delta of a put option.

TABLE 2-1

Black-Scholes-Merton Option Greeks (Partial Derivatives) Summary

Name	Symbol	Derivative	Other name	Formula
Delta call	Δ_{call}	$\frac{\partial c}{\partial S}$	Spot delta	$e^{(b-r)T}N(d_1)$
Delta put	Δ_{put}	$\frac{\partial p}{\partial S}$	Spot delta	$e^{(b-r)T}[N(d_1)-1]$
DdeltaDvol		$\frac{\partial^2 c}{\partial S \partial \sigma} = \frac{\partial^2 p}{\partial S \partial \sigma}$	Vanna, DvegaDspot	$\frac{-e^{(b-r)T}d_2}{\sigma} n(d_1)$
DvannaDvol		$\frac{\partial^3 c}{\partial S \partial \sigma^2} = \frac{\partial^3 p}{\partial S \partial \sigma^2}$		$\text{Vanna}\left(\frac{1}{\sigma}\right)\left(d_1 d_2 - \frac{d_1}{d_2} - 1\right)$
DdeltaDtime call		$-\frac{\partial^2 c}{\partial S \partial T}$	Charm, delta bleed	$-e^{(b-r)T}\left[n(d_1)\left(\frac{b}{\sigma\sqrt{T}} - \frac{d_2}{2T}\right) + (b-r)N(d_1)\right]$
DdeltaDtime put		$-\frac{\partial^2 p}{\partial S \partial T}$	Charm, delta bleed	$-e^{(b-r)T}\left[n(d_1)\left(\frac{b}{\sigma\sqrt{T}} - \frac{d_2}{2T}\right) - (b-r)N(-d_1)\right]$
Elasticity call	Λ_{call}	$\frac{\partial c}{\partial S}\frac{S}{\text{call}}$	Lambda, leverage	$e^{(b-r)T}N(d_1)\frac{S}{\text{call}}$
Elasticity put	Λ_{put}	$\frac{\partial p}{\partial S}\frac{S}{\text{put}}$	Lambda, leverage	$e^{(b-r)T}[N(d_1)-1]\frac{S}{\text{put}}$
Gamma	Γ	$\frac{\partial^2 c}{\partial S^2} = \frac{\partial^2 p}{\partial S^2}$	Convexity	$\frac{n(d_1)e^{(b-r)T}}{S\sigma\sqrt{T}}$
GammaP	Γ_P	$\frac{S}{100}\frac{\partial^2 c}{\partial S^2} = \frac{S}{100}\frac{\partial^2 p}{\partial S^2}$	Gamma percent	$\frac{S\Gamma}{100}$
DgammaDvol		$\frac{\partial^3 c}{\partial S^2 \partial \sigma} = \frac{\partial^3 p}{\partial S^2 \partial \sigma}$	Zomma	$\Gamma\left(\frac{d_1 d_2 - 1}{\sigma}\right)$

DgammaPDvol	$\frac{\partial^3 c}{\partial S^2 \partial \sigma} = \frac{\partial^3 p}{\partial S^2 \partial \sigma}$	ZommaP	$\Gamma_P \left(\frac{d_1 d_2 - 1}{\sigma} \right)$
DgammaDspot	$\frac{\partial^3 c}{\partial S^3} = \frac{\partial^3 p}{\partial S^3}$	Speed	$-\frac{\Gamma \left(1 + \frac{d_1}{\sigma \sqrt{T}}\right)}{S}$
DgammaPDspot	$\frac{\Gamma_P}{\partial S}$	Speed percent	$-\Gamma \frac{d_1}{100 \sigma \sqrt{T}}$
DgammaDtime	$-\frac{\partial^3 c}{\partial S^2 \partial T} = -\frac{\partial^3 p}{\partial S^2 \partial T}$	Colour, gamma bleed	$\Gamma \left(r - b + \frac{b d_1}{\sigma \sqrt{T}} + \frac{1 - d_1 d_2}{2T} \right)$
DgammaPDtime	$-\frac{\Gamma_P}{\partial T}$	Gamma percent bleed	$\Gamma_P \left(r - b + \frac{b d_1}{\sigma \sqrt{T}} + \frac{1 - d_1 d_2}{2T} \right)$
Vega	$\frac{\partial c}{\partial \sigma} = \frac{\partial p}{\partial \sigma}$	Zeta	$S e^{(b-r)T} n(d_1) \sqrt{T}$
VegaP	$\frac{\sigma}{10} \frac{\partial c}{\partial \sigma} = \frac{\sigma}{10} \frac{\partial p}{\partial \sigma}$	Percentage vega	$\frac{\sigma}{10} S e^{(b-r)T} n(d_1) \sqrt{T}$
DvegaDvol	$\frac{\partial^2 c}{\partial \sigma^2} = \frac{\partial^2 p}{\partial \sigma^2}$	Vomma, volga	$\text{Vega} \left(\frac{d_1 d_2}{\sigma} \right)$
DvegaPDvol	$\frac{\sigma}{10} \frac{\partial^2 c}{\partial \sigma^2} = \frac{\sigma}{10} \frac{\partial^2 p}{\partial \sigma^2}$	VommaP, volgaP	$\text{VegaP} \left(\frac{d_1 d_2}{\sigma} \right)$
DvommaDvol	$\frac{\partial^3 c}{\partial \sigma^3} = \frac{\partial^3 p}{\partial \sigma^3}$	Ultima	$\text{Vomma} \left(\frac{1}{\sigma} \right) \left(d_1 d_2 - \frac{d_1}{d_2} - \frac{d_2}{d_1} - 1 \right)$
DVegaDTime	$-\frac{\partial^2 c}{\partial \sigma \partial T} = -\frac{\partial^2 p}{\partial \sigma \partial T}$	Vega bleed	$\text{Vega} \left(r - b + \frac{b d_1}{\sigma \sqrt{T}} - \frac{1 + d_1 d_2}{2T} \right)$
Variance vega	$\frac{\partial c}{\partial V} = \frac{\partial p}{\partial V}$		$S e^{(b-r)T} n(d_1) \frac{\sqrt{T}}{2\sigma}$

(Continued)

TABLE 2-1 *(Continued)*

Name	Symbol	Derivative	Other name	Formula
DdeltaDvar		$\frac{\partial^2 c}{\partial S \partial V} = \frac{\partial^2 p}{\partial S \partial V}$		$-Se^{(b-r)T} n(d_1) \frac{d_2}{2\sigma^2}$
Variance vomma		$\frac{\partial^2 c}{\partial V^2} = \frac{\partial^2 p}{\partial V^2}$		$\frac{Se^{(b-r)T}\sqrt{T}}{4\sigma^3} n(d_1)(d_1 d_2 - 1)$
Variance ultima		$\frac{\partial^3 c}{\partial V^3} = \frac{\partial^3 p}{\partial V^3}$		$\frac{Se^{(b-r)T}\sqrt{T}}{8\sigma^5} n(d_1)[(d_1 d_2 - 1)(d_1 d_2 - 3) - (d_1^2 + d_2^2)]$
Theta call	Θ_{call}	$-\frac{\partial c}{\partial T}$	Expected bleed	$-\frac{Se^{(b-r)T} n(d_1)\sigma}{2\sqrt{T}} - (b-r)Se^{(b-r)T} N(d_1) - rXe^{-rT} N(d_2)$
Theta put	Θ_{put}	$-\frac{\partial p}{\partial T}$	Expected bleed	$-\frac{Se^{(b-r)T} n(d_1)\sigma}{2\sqrt{T}} + (b-r)Se^{(b-r)T} N(-d_1) + rXe^{-rT} N(-d_2)$
Driftless theta	θ	$-\frac{\partial c}{\partial T} = -\frac{\partial p}{\partial T}$	Pure bleed ($b = 0, r = 0$)	$-\frac{Sn(d_1)\sigma}{2\sqrt{T}}$
Rho call	ρ_{call}	$\frac{\partial c}{\partial r}$		$TXe^{-rT} N(d_2)$
Rho call futures option	ρ_{call}	$\frac{\partial c}{\partial r}$		$-Tc$
Rho put	ρ_{put}	$\frac{\partial p}{\partial r}$		$-TXe^{-rT} N(-d_2)$
Rho put futures option	ρ_{put}	$\frac{\partial p}{\partial r}$		$-Tp$

Phi call	Φ_{call}	$\frac{\partial c}{\partial q}$	Rho-2	$-TSe^{(b-r)T}N(d_1)$
Phi put	Φput	$\frac{\partial p}{\partial q}$	Rho-2	$TSe^{(b-r)T}N(-d_1)$
Carry rho call		$\frac{\partial c}{\partial b}$		$TSe^{(b-r)T}N(d_1)$
Carry rho put		$\frac{\partial p}{\partial b}$		$-TSe^{(b-r)T}N(-d_1)$
Zeta call	ζ_{call}		In-the-money prob.	$N(d_2)$
Zeta put	ζ_{put}		In-the-money prob.	$N(-d_2)$
DzetaDvol call		$\frac{\partial \zeta_{\text{call}}}{\partial \sigma}$		$-n(d_2)\left(\frac{d_1}{\sigma}\right)$
DZetaDVol put		$\frac{\partial \zeta_{\text{put}}}{\partial \sigma}$		$n(d_2)\left(\frac{d_1}{\sigma}\right)$
DZetaDTime call		$-\frac{\partial \zeta_{\text{call}}}{\partial T}$		$n(d_2)\left(\frac{b}{\sigma\sqrt{T}}-\frac{d_1}{2T}\right)$
DZetaDTime put		$-\frac{\partial \zeta_{\text{put}}}{\partial T}$		$-n(d_2)\left(\frac{b}{\sigma\sqrt{T}}-\frac{d_1}{2T}\right)$
Strike delta call		$\frac{\partial c}{\partial X}$	Discounted probability	$-e^{-rT}N(d_2)$
Strike delta put		$\frac{\partial p}{\partial X}$	Discounted probability	$e^{-rT}N(-d_2)$
Strike gamma		$\frac{\partial^2 c}{\partial X^2}=\frac{\partial^2 p}{\partial X^2}$	RND	$\frac{n(d_2)e^{-rT}}{X\sigma\sqrt{T}}$

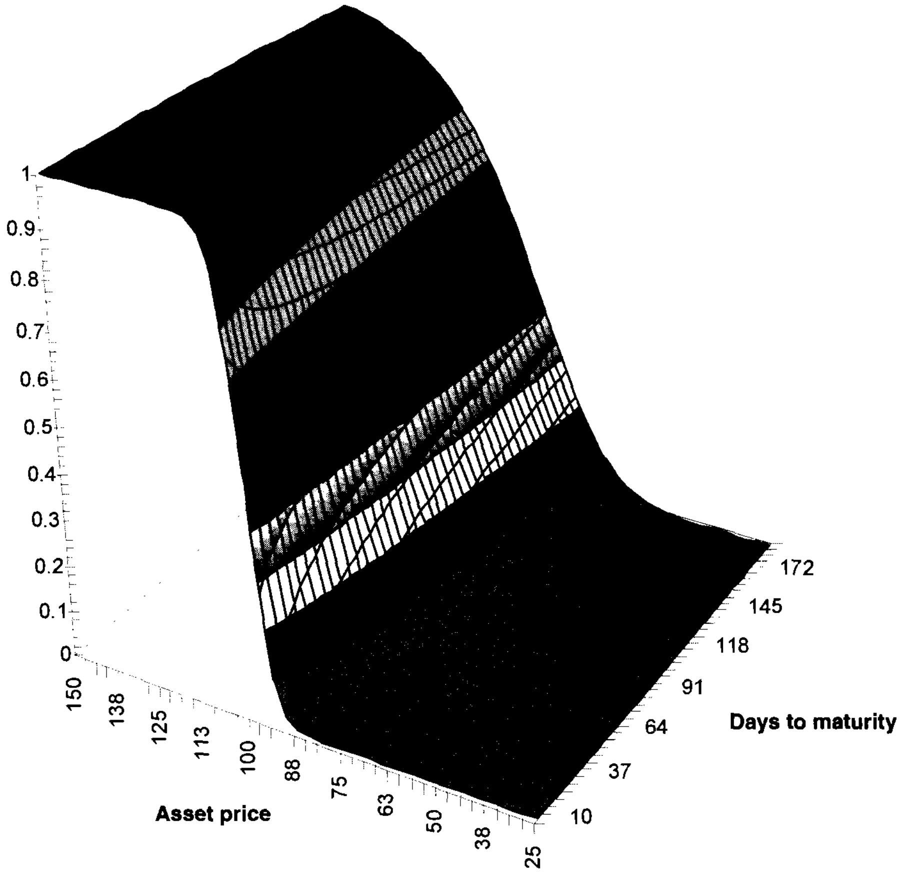


FIGURE 1 Spot delta call: $X = 100$, $r = 7\%$, $b = 4\%$, $\sigma = 30\%$.

Example

Consider a futures option with six months to expiration. The futures price is 105, the strike price is 100, the risk-free interest rate is 10% per year, and the volatility is 36% per year. Thus, $S = 105$, $X = 100$, $T = 0.5$, $r = 0.1$, $b = 0$, and $\sigma = 0.36$.

$$d_1 = \frac{\ln(105/100) + (0 + 0.36^2/2)0.5}{0.36\sqrt{0.5}} = 0.3189$$

$$N(d_1) = N(0.3189) = 0.6251$$

$$\Delta_{\text{call}} = e^{(0-0.1)0.5} N(d_1) = 0.5946$$

$$\Delta_{\text{put}} = e^{(0-0.1)0.5}[N(d_1) - 1] = -0.3566$$

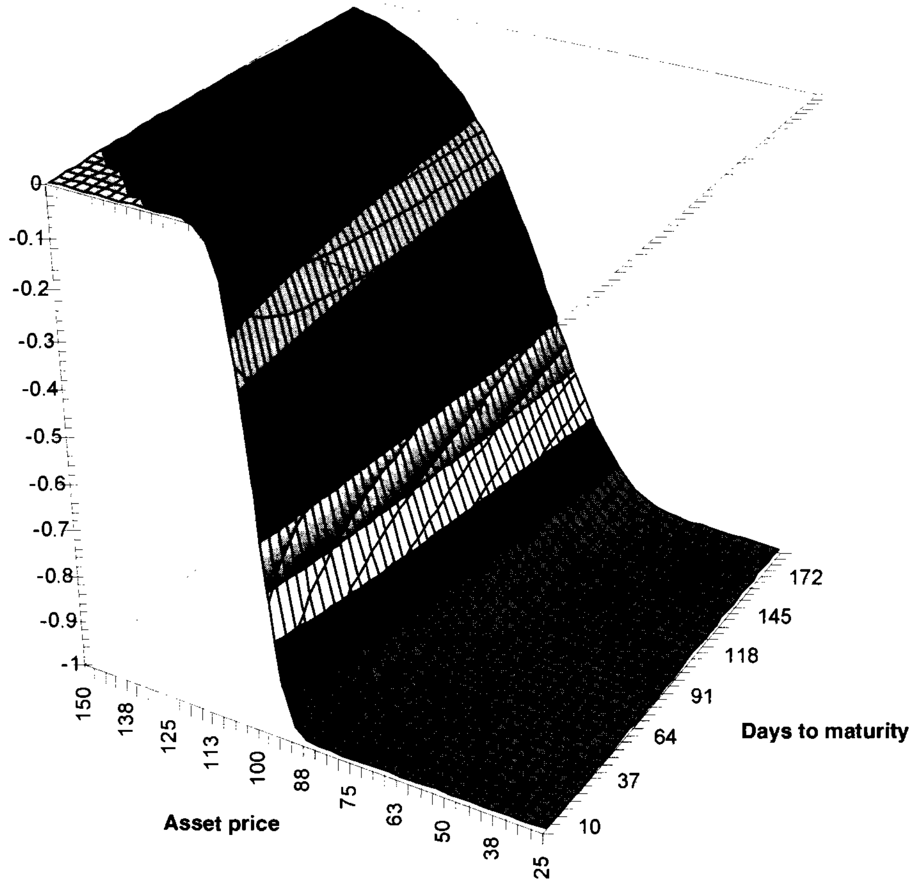


FIGURE 2 Spot delta put: $X = 100$, $r = 7\%$, $b = 4\%$, $\sigma = 30\%$.

Computer algorithm

```
Function GDelta(CallPutFlag As String, S As Double, X As Double, T As Double, _
  r As Double, b As Double, v As Double) As Double

    Dim d1 As Double

    d1 = (Log(S / X) + (b + v^2 / 2) * T) / (v * Sqr(T))
    If CallPutFlag = "c" Then
        GDelta = Exp((b - r) * T) * CND(d1)
    Else
        GDelta = -Exp((b - r) * T) * CND(-d1)
    End If

End Function
```

The Behavior of Delta

As a call option gets deep-in-the-money, $N(d_1)$ approaches 1, but it never exceeds 1 (since it's a cumulative distribution function). For a European call option on a nondividend-paying stock, $N(d_1)$ is moreover equal to the option's delta. Delta can therefore never exceed 1 for this option. For general European call options, delta is given by $e^{(b-r)T}N(d_1)$. If the term $e^{(b-r)T}$ is larger than 1 and the option is deep-in-the-money, the delta can thus become considerably larger than 1. This occurs if the cost-of-carry is larger than the interest rate, or if interest rates are negative. Figure 3 illustrates the delta of a call option. As expected the delta exceeds 1 when time to maturity is large and the option is deep-in-the-money.

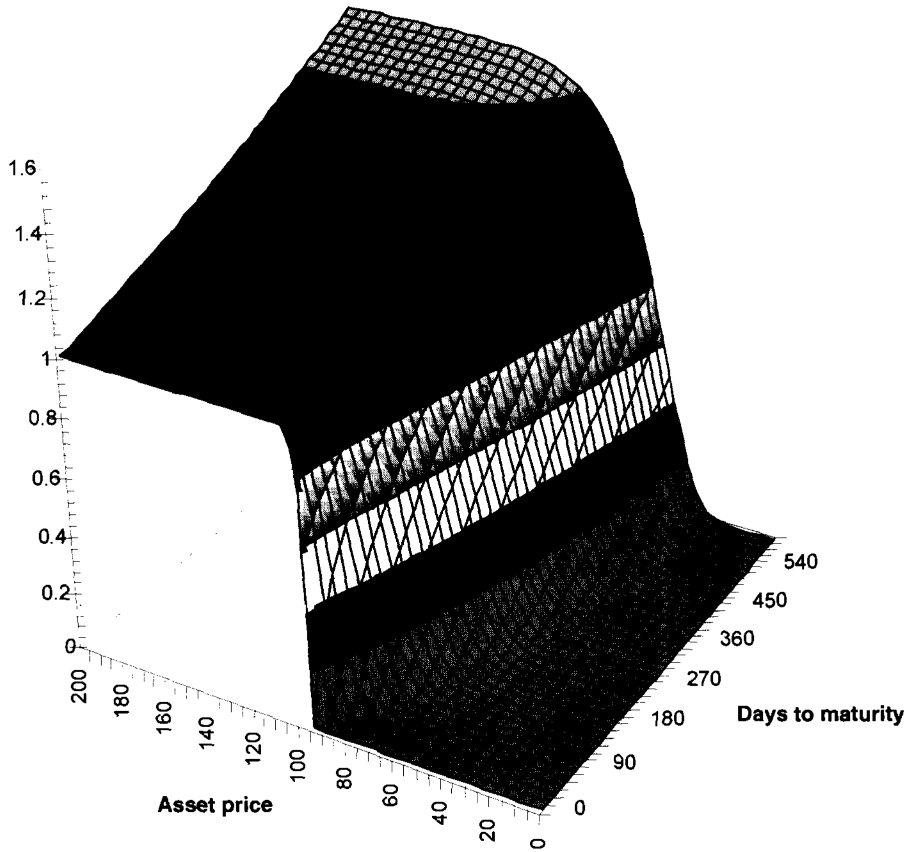


FIGURE 3 Spot delta call: $X = 100$, $r = 5\%$, $b = 30\%$, $\sigma = 25\%$.

Example
Consider a commodity option with two years to expiration. The commodity price is 90, the strike price is 40, the risk-free interest rate is 3% per year, the cost-of-carry is 9% per year, and the volatility is 20%. What is the delta of a call option? $S = 90$, $X = 40$, $T = 2$, $r = 0.03$, $b = 0.09$, and $\sigma = 0.2$.

$$d_1 = \frac{\ln(90/40) + (0.09 + 0.2^2/2)2}{0.2\sqrt{2}} = 3.6449$$

$$N(d_1) = N(3.6449) = 0.9999$$

$$\Delta_{\text{call}} = e^{(0.09-0.03)2}N(3.6449) = 1.1273$$

The delta of this option is about 112.73%, which implies that the call price will increase (decrease) with about 1.13 dollars if the spot price increases (decreases) by one dollar.

2.1.2 Delta Mirror Strikes and Assets

The following strikes equalize the absolute values of deltas of put and call options:

$$X_{\text{put}} = \frac{S^2}{X_{\text{call}}}e^{(2b+\sigma^2)T} \qquad X_{\text{call}} = \frac{S^2}{X_{\text{put}}}e^{(2b+\sigma^2)T}$$

That is

$$\Delta_{\text{call}}(S, X_{\text{call}}, T, r, b, \sigma) = -\Delta_{\text{put}}\left(S, \frac{S^2}{X_{\text{call}}}e^{(2b+\sigma^2)T}, T, r, b, \sigma\right) \tag{2.3}$$

These relationships are useful to determine strikes for delta-neutral option strategies, especially for strangles, straddles, and butterflies. The weakness of this approach is that it works only for symmetric volatility smiles. In practice, you still often need only an approximately delta neutral strangle. Moreover, volatility smiles are often more or less symmetric in the currency market.

In the special case of a straddle-symmetric-delta strike, described by Wystrup (1999), the formulas above simplify to

$$X_{\text{call}} = X_{\text{put}} = Se^{(b+\sigma^2/2)T}$$

A related concept is the straddle-symmetric-asset price. Given identical strikes for a put and a call, what asset price will yield the same absolute delta value? This asset price is given by

$$S = Xe^{(-b-\sigma^2/2)T}$$

At this strike and delta-symmetric-asset price, the delta is given by $\frac{e^{(b-r)T}}{2}$ for the call, and $-\frac{e^{(b-r)T}}{2}$ for the put. Only options on nondividend-paying stocks ($b = r$) can simultaneously have an absolute delta of 0.5 (50%) for a put and a call.[1] Moreover, the delta-symmetric strike is also the strike where gamma and vega attain their maximum, *ceteris paribus*. Notice that the maximal gamma and vega, as well as the delta-neutral strikes, are not at-the-money forward (as assumed by many traders). Moreover, an in-the-money put can naturally have absolute delta lower than 0.5, while an out-of-the-money call can have delta higher than 0.5.

For an option that is at the straddle-symmetric-delta strike, the BSM formula can be simplified to

$$c = \frac{Se^{(b-r)T}}{2} - Xe^{-rT}N(-\sigma\sqrt{T})$$

and

$$p = Xe^{-rT}N(\sigma\sqrt{T}) - \frac{Se^{(b-r)T}}{2}$$

At this point the option value will not change based on changes in cost-of-carry (dividend yield, etc.). This is as expected, as we have to adjust the strike accordingly.

Example

What should the strike be for a delta-neutral straddle with nine months to maturity, the risk-free rate 10%, volatility 60%, for a stock trading at 100? $S = 100$, $T = 0.75$, $r = 0.1$, $b = 0.1$, and $\sigma = 0.6$.

$$X_{\text{call}} = X_{\text{put}} = 100e^{(0.1+0.6^2/2)0.75} = 123.3678$$

2.1.3 Strike from Delta

Options are quoted by delta rather than strike in several OTC (over-the-counter) markets. This is a common quotation method in, for instance, the OTC currency options market, where one typically asks for a delta and expects the salesperson to return a price (in terms of volatility or pips) as well as the strike, given a spot reference. In these cases, one needs to find the strike that corresponds to a given delta. Several option software systems solve this numerically using Newton-Raphson or bisection. This is actually not necessary, however. With an inverted cumulative normal distribution function $N^{-1}(\cdot)$, the strike

[1]This clearly also applies to commodity options when the cost-of-carry is r.

can be derived from the delta analytically, as described by Wystrup (1999). For a call option

$$X_{\text{call}} = S\exp[-N^{-1}(\Delta_{\text{call}}e^{(r-b)T})\sigma\sqrt{T} + (b + \sigma^2/2)T] \tag{2.4}$$

and for a put we have

$$X_{\text{put}} = S\exp[N^{-1}(-\Delta_{\text{put}}e^{(r-b)T})\sigma\sqrt{T} + (b + \sigma^2/2)T] \tag{2.5}$$

To get a robust and accurate implementation of this formula, it is necessary to use an accurate approximation of the inverse cumulative normal distribution. The algorithm of Moro (1995) is one possible implementation; this is given in Chapter 13.

Example
What should the strike be for a three-month call stock index option to get a delta of 0.25, the risk-free rate 7%, dividend yield 3%, and volatility 50%, and with the stock index trading at 1800? $S = 1800$, $T = 0.25$, $r = 0.07$, $b = 0.07 - 0.03 = 0.04$, $\sigma = 0.5$, and thus

$$N^{-1}(\Delta_{\text{call}}e^{(r-b)T}) = N^{-1}(0.25e^{(0.07-0.04)0.25}) = -0.6686$$

$$X_{\text{call}} = 1800 \times \exp[0.6686 \times 0.5\sqrt{0.25} + (0.04 + 0.5^2/2)0.25] = 2217.0587$$

That is, to get a delta of 0.25, we need to set the strike to 2217.0587.

2.1.4 Futures Delta from Spot Delta

The delta we have looked at above is known as spot delta—that is, the delta of the option in terms of the underlying asset, we are inputting into the model. Sometimes when we, for example, value an option on a stock inputting the stock price, we can in some markets choose if we want to hedge with the stock itself or alternatively hedge with the stock futures. In that case it is useful to go from spot delta to futures delta

$$\Delta_F = \Delta e^{-bT}$$

where Δ is the delta given earlier. Alternatively, you could naturally have inputted the futures price directly into the BSM formula, and the spot delta would in that case be equal to the futures delta. In the case where you hedge with a forward contract with same expiration as the option, the formula above also holds. This is particularly useful in the FX market, where you typically can choose between hedging with the currency spot or alternatively a forward with expiration matching the option expiration.

2.1.5 DdeltaDvol and DvegaDspot

DdeltaDvol, defined as $\frac{\partial \Delta}{\partial \sigma}$, is mathematically the same as Dvega-Dspot, defined as $\frac{\partial \text{vega}}{\partial S}$ (aka vanna). They both measure approximately how much delta will change due to a small change in the volatility, and how much vega will change due to a small change in the asset price:

$$\frac{\partial^2 c}{\partial S \partial \sigma} = \frac{\partial^2 p}{\partial S \partial \sigma} = \frac{-e^{(b-r)T} d_2}{\sigma} n(d_1), \tag{2.6}$$

where $n(x)$ is the standard normal density

$$n(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

Figure 4 illustrates the DdeltaDvol. The DdeltaDvol can evidently assume positive and negative values. It attains its maximal value at

$$S_L = X e^{-bT - \sigma\sqrt{T}\sqrt{4+T\sigma^2}/2}$$

and attains its minimal value at

$$S_U = X e^{-bT + \sigma\sqrt{T}\sqrt{4+T\sigma^2}/2}$$

Similarly, given the asset price, options with strike X_L attain minimal DdeltaDvol at

$$X_L = S e^{bT - \sigma\sqrt{T}\sqrt{4+T\sigma^2}/2}$$

and options with strike X_U attain maximum DdeltaDvol at

$$X_U = S e^{bT + \sigma\sqrt{T}\sqrt{4+T\sigma^2}/2}$$

A natural question is whether these measures have any real meaning? Recall that Black and Scholes assumed constant volatility when deriving their formula. Despite being theoretically inconsistent with the Black-Scholes analysis, the measures may well represent good approximations. Webb (1999) has more practical information on DvegaDspot and vanna.

Example

Consider a put option on a stock trading at 90, with three months to maturity, strike 80, three month risk-free interest rate of 5%, and

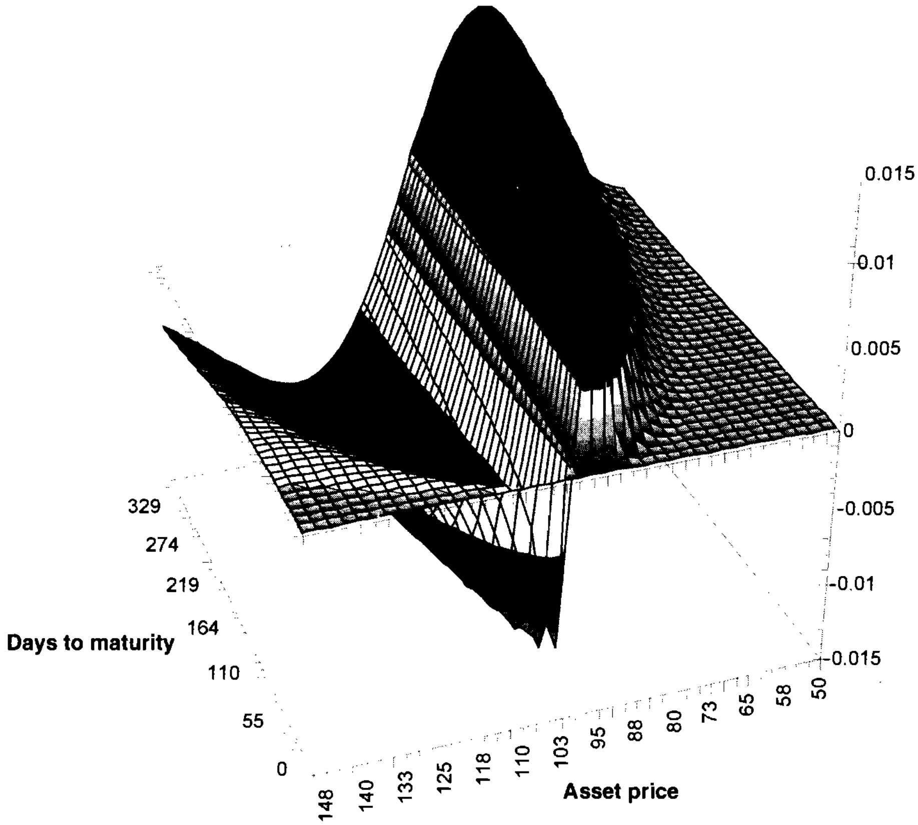


FIGURE 4 DdeltaDvol: $X = 100$, $r = 5\%$, $b = 0\%$, $\sigma = 20\%$.

volatility of 20%. What is the DdeltaDvol? $S = 90$, $X = 80$, $T = 0.25$, $r = 0.05$, $b = 0.05$, $\sigma = 0.2$, thus

$$d_1 = \frac{\ln(90/80) + (0.05 + 0.2^2/2)0.25}{0.2\sqrt{0.25}} = 1.3528$$

$$d_2 = d_1 - \sigma\sqrt{T} = 1.3528 - 0.2\sqrt{0.25} = 1.2528$$

$$n(d_1) = n(1.3528) = 0.1598$$

$$\frac{\partial^2 p}{\partial S \partial \sigma} = \frac{-e^{(0.05-0.05)0.25}1.2528}{0.2}0.1598 = -1.0008$$

If the volatility increases from 20% to 21%, the delta of the put will thus decrease by about one percentage point $\frac{-1.0008}{100}$. Similarly, if the volatility decreases from 20% to 19%, delta will increase by one percent point. Alternatively, this shows that the options vega will decrease by -0.0100 if the stock price increases by one.

2.1.6 DvannaDvol

The second-order partial derivative of delta with respect to volatility, also known as DvannaDvol, is given by

$$\frac{\partial^3 c}{\partial S \partial \sigma^2} = \frac{\partial^3 p}{\partial S \partial \sigma^2} = \frac{-e^{(b-r)T} d_2}{\sigma} n(d_1) \frac{1}{\sigma} \left(d_1 d_2 - \frac{d_1}{d_2} - 1 \right)$$
$$= \text{vanna} \left(\frac{1}{\sigma} \right) \left(d_1 d_2 - \frac{d_1}{d_2} - 1 \right) \tag{2.7}$$

It is necessary to divide by 10,000 to get this Greek on the metric of a one point change in volatility—for example, from 20% to 21%.

Figure 5 illustrates DvannaDvol for varying asset price and time to maturity.

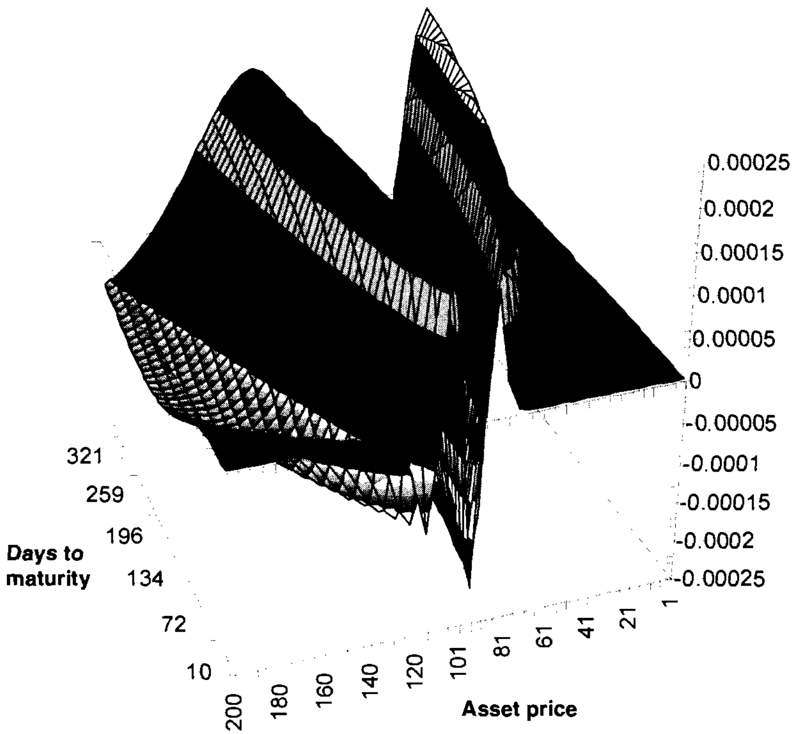


FIGURE 5 DvannaDvol: $X = 100$, $r = 7\%$, $b = 4\%$, $\sigma = 40\%$.

2.1.7 DdeltaDtime, Charm

DdeltatDtime, also known as charm (Garman, 1992) or Delta Bleed, a term used in the excellent book by Taleb (1997), measures the sensitivity of delta to changes in time,

$$-\frac{\partial \Delta_{\text{call}}}{\partial T} = -e^{(b-r)T}\left[n(d_1)\left(\frac{b}{\sigma\sqrt{T}} - \frac{d_2}{2T}\right) + (b-r)N(d_1)\right] \leq\geq 0 \quad (2.8)$$

and

$$-\frac{\partial \Delta_{\text{put}}}{\partial T} = -e^{(b-r)T}\left[n(d_1)\left(\frac{b}{\sigma\sqrt{T}} - \frac{d_2}{2T}\right) - (b-r)N(-d_1)\right] \leq\geq 0 \quad (2.9)$$

This Greek indicates what happens with delta when we move closer to maturity. Figure 6 illustrates the charm value for different values of the underlying asset and different times to maturity.

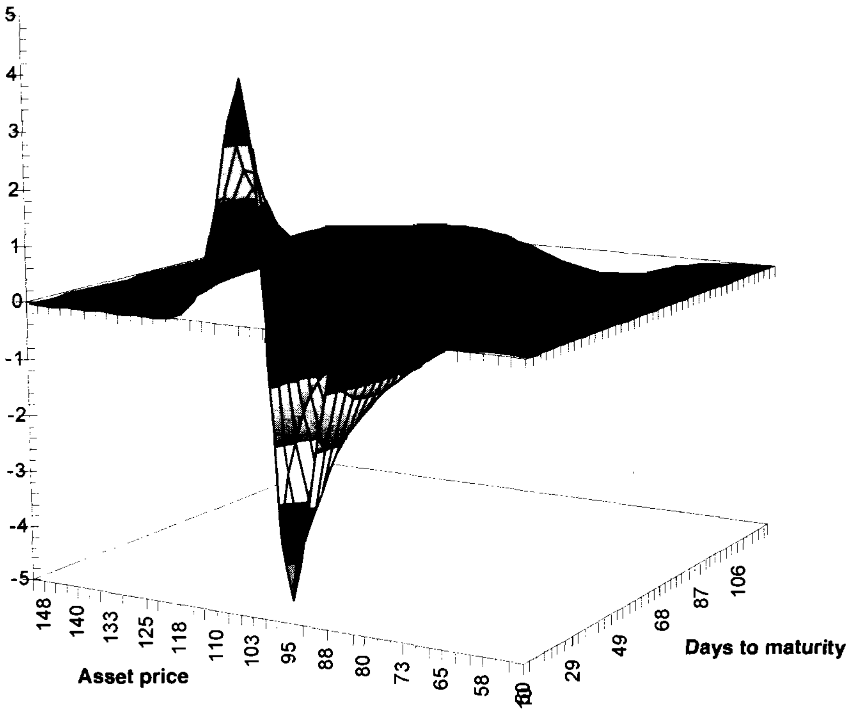


FIGURE 6 Charm: $X = 100$, $r = 5\%$, $b = 0\%$, $\sigma = 30\%$.

One can have both forward and backward bleed; see Taleb (1997). Taleb also points out the importance of taking into account how expected changes in volatility over the given time period affect delta.

All partial derivatives with respect to time have the advantage over other Greeks that we know which direction time moves. We moreover know that time moves at a constant rate. This is in contrast to, for instance, the spot price, volatility, or interest rate.

Example

Consider a European put option on a futures currently priced at 105. The strike price is 90, the time to expiration is three months, the risk-free interest is 14% per year, and the volatility is 24% per year. What is the charm of the option? $S = 105$, $X = 90$, $T = 0.25$, $r = 0.14$, $b = 0$, $\sigma = 0.24$, and thus

$$d_1 = \frac{\ln(105/90) + 0.25 \times 0.24^2/2}{0.24\sqrt{0.25}} = 1.3446$$

$$d_2 = d_1 - \sigma\sqrt{T} = 1.3446 - 0.24\sqrt{0.25} = 1.2246$$

$$n(d_1) = n(1.3446) = 0.1616$$

$$N(-d_1) = N(-1.3446) = 0.0894$$

$$-\frac{\partial \Delta_{\text{put}}}{\partial T} = -e^{(0-0.14)0.25}\left[0.1616\left(\frac{0}{0.24\sqrt{0.25}} - \frac{1.2246}{2 \times 0.25}\right) - (0 - 0.14)0.0894\right] = 0.3700$$

The DdeltaDtime for one day is thereby $\frac{0.3700}{365} = 0.0010$.

2.1.8 Elasticity

The elasticity of an option is its sensitivity in percent to a percent change in the underlying asset price.

Call

$$\Lambda_{\text{call}} = \Delta_{\text{call}}\frac{S}{\text{call}} = e^{(b-r)T}N(d_1)\frac{S}{\text{call}} > 1 \tag{2.10}$$

Put

$$\Lambda_{\text{put}} = \Delta_{\text{put}}\frac{S}{\text{put}} = e^{(b-r)T}[N(d_1) - 1]\frac{S}{\text{put}} < 0 \tag{2.11}$$

The option's elasticity is a useful measure on its own, as well as a good method to estimate the volatility, beta, and expected return from an option.

Example

What is the elasticity of a put option with the same parameters as in the delta example? $S = 105$, $X = 100$, $T = 0.5$, $r = 0.1$, $b = 0$, and $\sigma = 0.36$.

$$\Lambda_{\text{put}} = e^{(0-0.1)0.5}[N(d_1) - 1]\frac{105}{7.6767} = -4.8775$$

Option Volatility

The volatility of an option over a short period of time is approximately equal to the elasticity of the option multiplied by the volatility of the underlying asset σ:[2]

$$\sigma_O \approx \sigma|\Lambda| \tag{2.12}$$

As the lambda (elasticity) of the option will change with both the underlying asset price and time, we can easily see that this at best is an approximation. It still offers some intuition on leverage and risk effects associated with options.

Option Beta

The elasticity also allows us to easily compute an option's beta. If asset prices follow geometric Brownian motions, the continuous-time capital asset pricing model of Merton (1971) holds. Expected return on a generic asset, μ_S, then satisfy the intertemporal CAPM equation

$$\mu_S = r + (r_m - r)\beta_S$$

where r is the risk-free rate, r_m is the expected return on the market portfolio, and β_S is the beta of the asset. To determine the expected return of an option, we need its beta. The beta of a call is given by Black and Scholes (1973):

$$\beta_{\text{call}} = \frac{S}{\text{call}}\Delta_{\text{call}}\beta_S, \tag{2.13}$$

while the beta of a put

$$\beta_{\text{put}} = \frac{S}{\text{put}}\Delta_{\text{put}}\beta_S \tag{2.14}$$

The expected return on a beta neutral option strategy should be equal to the risk-free rate.

---

[2]Bensoussan, Crouhy, and Galai (1995) apply this approximation to find an approximate price for a compound option; see Chapter 4.

Option Sharpe Ratios

The Sharpe (1966) ratio is independent of leverage. The Sharpe ratio of an option is therefore identical to that of the underlying asset:

$$\frac{\mu_O - r}{\sigma_O} = \frac{\mu_S - r}{\sigma}$$

where μ_O is the expected return of the option. This relationship indicates the limited usefulness of the Sharpe ratio as a risk-return measure for options. For instance, shorting a large amount of deep out-of-the-money options may result in a “nice” Sharpe ratio, but, it represents a high-risk strategy. A relevant question is whether you should use the same volatility for all strikes. For instance, deep out-of-the-money stock options typically trade for much higher implied volatility than at-the-money options. Using the volatility smile when computing Sharpe ratios for deep out-of-the-money options can make the Sharpe ratio work, slightly better for options. McDonald (2002) offers a more detailed discussion of option Sharpe ratios.

2.2 GAMMA GREEKS

2.2.1 Gamma

Gamma is the delta’s sensitivity to small changes in the underlying asset price. Gamma is identical for put and call options:

$$\Gamma_{call,put} = \frac{\partial^2 c}{\partial S^2} = \frac{\partial^2 p}{\partial S^2} = \frac{n(d_1)e^{(b-r)T}}{S\sigma\sqrt{T}} > 0 \qquad (2.15)$$

This is the standard gamma measure given in most textbooks like Hull (2005) and Wilmott (2000). It measures the change in delta for a one-unit change in the price of the underlying asset price.

Figure 7 illustrates the gamma of a call for different values of the underlying asset and different times to maturity.

Example

Consider a stock option with nine months to expiration. The stock price is 55, the strike price is 60, the risk-free interest rate is 10% per year, and the volatility is 30% per year. $S = 55$, $X = 60$, $T = 0.75$,

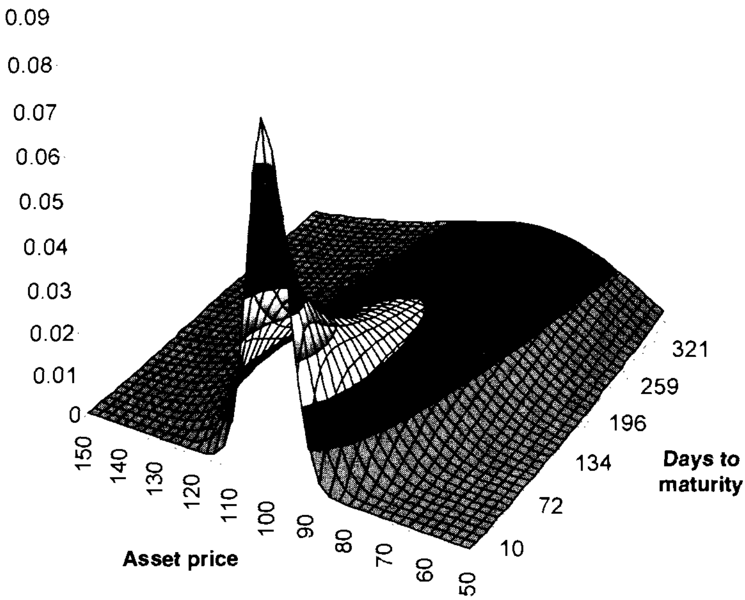


FIGURE 7 Gamma: $X = 100$, $r = 5\%$, $b = 5\%$, $\sigma = 30\%$.

$r = 0.1, b = 0.1, \sigma = 0.3$, which yields

$$d_1 = \frac{\ln(55/60) + (0.1 + 0.3^2/2)0.75}{0.3\sqrt{0.75}} = 0.0837$$

$$n(d_1) = n(0.0837) = \frac{1}{\sqrt{2\pi}} e^{-0.0837^2/2} = 0.3975$$

$$\Gamma_{\text{call,put}} = \frac{0.3975 e^{(0.1-0.1)0.75}}{55 \times 0.3\sqrt{0.75}} = 0.0278$$

2.2.2 Maximal Gamma and the Illusions of Risk

A popular rule of thumb is that gamma is largest for at-the-money or at-the-money-forward options. How good is this rule of thumb? Given a strike price and time to maturity, gamma attains its maximum when the asset price is[3]

$$S_{\bar{\Gamma}} = X e^{(-b-3\sigma^2/2)T}$$

[3]Rubinstein (1990) indicates that this maximum may explain why the greatest demand for calls tend to be just slightly out-of-the money.

Given the asset price and time to maturity, gamma attains its maximum when the strike is

$$X_{\bar{\Gamma}} = Se^{(b+\sigma^2/2)T}$$

Several large investment firms impose risk limits on how much gamma their trades' portfolios can have. In the equity market it is common to use the standard textbook approach to compute gamma, as shown above.

Shorting a long-term call (put) option that later is deep-out-of-the-money (in-the-money) can blow up the gamma risk limits, even if you actually have close to zero gamma risk. The high gamma risk for long-dated deep-out-of-the-money options typically is only an illusion. This illusion of risk can be avoided by looking at percentage changes in the underlying asset (gammaP), as is typically done for FX options.

Gamma Saddle Alexander Adamchuk was the first to make me aware of the fact that gamma has a saddle point.[4] The saddle point is attained for the time to expiry[5]

$$T_S = \frac{1}{2(\sigma^2 + 2b - r)}$$

and at asset price

$$S_{\bar{\Gamma}} = Xe^{(-b-3\sigma^2/2)T_S}$$

The gamma at this point is given by

$$\Gamma_S = \Gamma(S_{\bar{\Gamma}}, T_S) = \frac{\sqrt{\frac{e}{\pi}}\sqrt{\frac{2b-r}{\sigma^2} + 1}}{X} \tag{2.16}$$

Figure 8 illustrates the gamma's saddle point. The saddle point is between the two gamma "mountaintops." This graph also illustrates one of the big limitations in the textbook gamma definition, which is in active use by many option systems and traders. The gamma increases dramatically when we have a long time to maturity and the asset price is close to zero. How can the gamma be larger than for an option closer to at-the-money? Is the real gamma risk that big? No, this is in most

[4]Described by Adamchuk at the Wilmott forum located at www.wilmott.com, February 6, 2002, and even earlier on his Web page.

[5]It is worth mentioning that T_S must be larger than zero for the gamma to have a saddle point. That means b must be larger than $\frac{r-\sigma^2}{2}$, or r must be smaller than $\sigma^2 + 2b$.

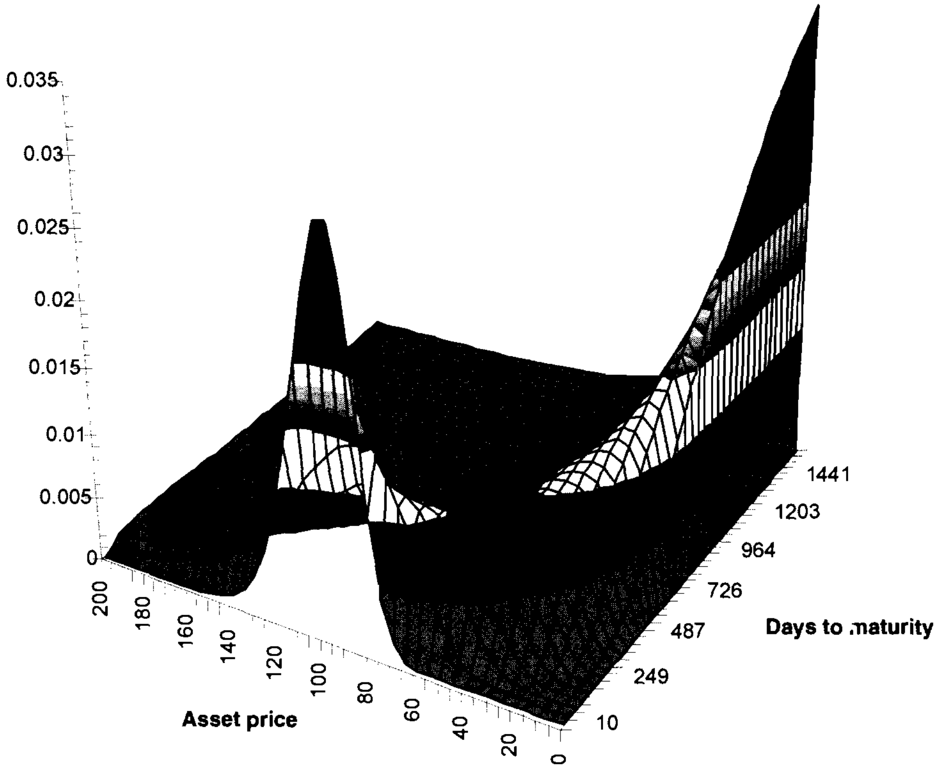


FIGURE 8 Saddle Gamma: $X = 100$, $r = 5\%$, $b = 5\%$, $\sigma = 80\%$.

cases simply an illusion, due to the above unmotivated definition of gamma. Gamma is typically defined as the change in delta for a one-unit change in the asset price. When the asset price is close to zero, a one-unit change is naturally enormous in percent of the asset price. In this case it is also highly unlikely that the asset price will change by one dollar in an instant. In other words, the gamma measurement should be reformulated, as many option systems already have done. It makes far more sense to look at percentage moves in the underlying assets than unit moves. To compare gamma risk from different underlying assets, one should also adjust for the volatility in the underlying assets.

Example

Consider a stock option with strike 500, risk-free rate 8%, and volatility 40%. For what time and stock price does gamma have a saddle point, and what is the gamma at this point? With $X = 500$, $r = 0.08$,

$b = 0.08$, and $\sigma = 0.4$, the saddle time must be (in number of years)

$$T_S = \frac{1}{2(0.4^2 + 2 \times 0.08 - 0.08)} = 2.0833$$

the saddle point stock price must be

$$S_{\bar{\Gamma}} = 500e^{(-0.08-3\times0.4^2/2)2.0833} = 256.7086$$

and the gamma at this point is

$$\Gamma_S = \Gamma(256.7086, 2.0833) = \frac{\sqrt{\frac{e}{\pi}}\sqrt{\frac{2\times0.08-0.08}{0.4^2}+1}}{500} = 0.0023$$

That is, for the saddle point $(S = 256.7086, T = 2.0833)$, we have a gamma of 0.0023.

2.2.3 GammaP

As mentioned, there are several problems with the traditional definition of gamma. A better measure is to look at percentage changes in delta for percentage changes in the underlying asset (gamma percent).[6] This definition yields

$$\Gamma_P = \frac{S\Gamma}{100} = \frac{n(d_1)e^{(b-r)T}}{100\sigma\sqrt{T}} > 0 \tag{2.17}$$

GammaP attains a maximum at an asset price of

$$S_{\bar{\Gamma}_P} = Xe^{(-b-\sigma^2/2)T}$$

Alternatively, given the asset price, the maximal Γ_P occurs at strike

$$X_{\bar{\Gamma}_P} = Se^{(b+\sigma^2/2)T}$$

This is also where we have a straddle-symmetric asset price as well as maximal gamma. This implies that a delta-neutral straddle has maximal Γ_P. In most circumstances, measuring the gamma risk as Γ_P instead of as gamma avoids the illusion of a high gamma risk when the option is far out-of-the-money and the asset price is low. Figure 9 is an illustration of this, using the same parameters as in Figure 8.

[6]Wystrup (1999) also describes how this redefinition of gamma removes the dependence on the spot level S. He calls it "traders' gamma." This measure of gamma has for a long time been popular, particularly in the FX market, but it is still absent in options textbooks, until now!

If the cost-of-carry is very high, it is still possible to experience high Γ_P for deep-out-of-the-money call options with a low asset price and a long time to maturity. This occurs because a high cost-of-carry can make the ratio of a deep-out-of-the money call to the spot close to the at-the-money forward. At this point the spot delta will be close to 50%, and so the Γ_P will be large. This is not an illusion of gamma risk, but a reality. Figure 10 shows Γ_P with the same parameters as in Figure 9, with a cost-of-carry of 60%.

To makes things even more complicated, the high Γ_P for deep-out-of-the-money calls (in-the-money puts) applies only in the case when we are dealing with spot gammaP (changes in spot delta). We can avoid this by looking at future/forward gammaP. However, if you hedge with spot, then spot gammaP is the relevant metric. Only if you hedge with the future/forward, the forward gammaP is the relevant metric.

The forward gammaP we have when the underlying asset is a future/forward and the cost-of-carry is set to zero.

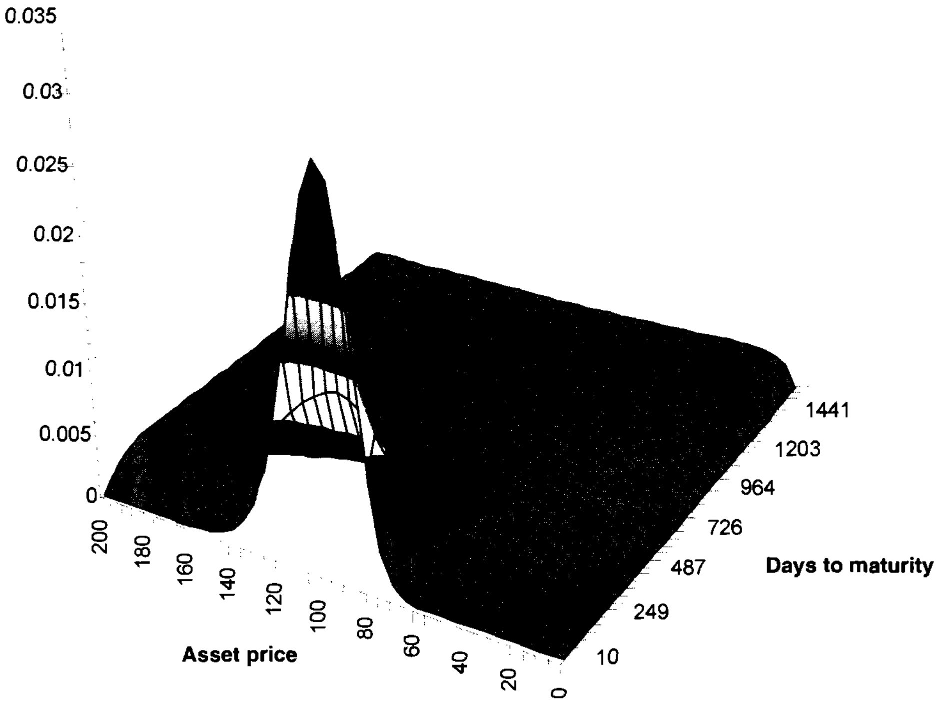


FIGURE 9 GammaP: $X = 100$, $r = 5\%$, $b = 5\%$, $\sigma = 80\%$.

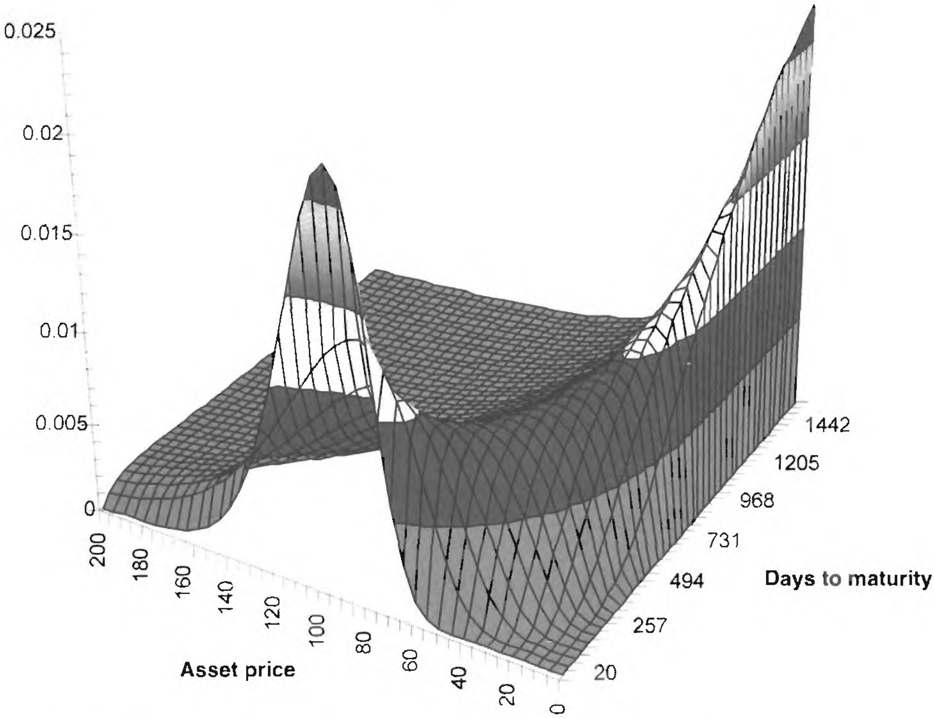


FIGURE 10 Saddle GammaP: $X = 100$, $r = 5\%$, $b = 60\%$, $\sigma = 80\%$.

Example

What is the gammaP for an option on a stock with price 50, strike 50, time to maturity 8 days, risk-free rate 12%, and volatility 15%? $S = 50$, $X = 50$, $T = 8/365$, $r = 0.12$, $b = 0.12$, and $\sigma = 0.15$ yields

$$d_1 = \frac{\ln(50/50) + (0.12 + 0.15^2/2)8/365}{0.15\sqrt{8/365}} = 0.1295$$

$$n(d_1) = n(0.1295) = \frac{1}{\sqrt{2\pi}} e^{-0.1295^2/2} = 0.3956$$

$$\Gamma_P = \frac{0.3956 \times e^{(0.12-0.12)8/365}}{100 \times 0.15\sqrt{8/365}} = 0.1781$$

That is, for a 1% move in the underlying asset, in this case a $50 \times 0.01 = 0.5$ move, the delta will change by about 0.1781.

2.2.4 Gamma Symmetry

Given the same strike, the gamma is identical for both put and call options. Although this equality breaks down when the strikes differ, there is a useful put and call gamma symmetry. The put-call symmetry of Bates (1991) and Carr and Bowie (1994) is given by

$$c(S, X, T, r, b, \sigma) = \frac{X}{Se^{bT}} p\left(S, \frac{(Se^{bT})^2}{X}, T, r, b, \sigma\right)$$

This put-call value symmetry yields the gamma symmetry. The gamma symmetry is more general, however, since it is independent of whether the option is a put or a call. It could, for example, be two calls, two puts, or a put and a call.

$$\Gamma(S, X, T, r, b, \sigma) = \frac{X}{Se^{bT}} \Gamma\left(S, \frac{(Se^{bT})^2}{X}, T, r, b, \sigma\right) \tag{2.18}$$

The put-call symmetry also gives us vega and cost-of-carry symmetries, and in the case of zero cost-of-carry, also theta and rho symmetry. Delta symmetry, however, does not obtain.

2.2.5 DgammaDvol, Zomma

DgammaDvol (aka zomma) is the sensitivity of gamma with respect to changes in implied volatility. DgammaDvol is in my view one of the more important Greeks for options trading. It is given by

$$\text{DgammaDvol}_{\text{call, put}} = \frac{\partial \Gamma}{\partial \sigma} = \Gamma\left(\frac{d_1 d_2 - 1}{\sigma}\right) \lessgtr 0 \tag{2.19}$$

where Γ is the textbook gamma of the option. For the gammaP, we have DgammaPDvol

$$\text{DgammaPDvol}_{\text{call, put}} = \Gamma_P\left(\frac{d_1 d_2 - 1}{\sigma}\right) \lessgtr 0 \tag{2.20}$$

For practical purposes, where one typically wants to look at DgammaDvol for a one-unit volatility change—for example, from 30% to 31%—one should divide the DgammaDvol by 100. Moreover, DgammaDvol and DgammaPDvol are negative for asset prices between S_L and S_U and positive outside this interval, where

$$S_L = Xe^{-bT - \sigma\sqrt{T}\sqrt{4+T\sigma^2}/2}$$

$$S_U = Xe^{-bT + \sigma\sqrt{T}\sqrt{4+T\sigma^2}/2}$$

For a given asset price, the DgammaDvol and DgammaPDvol are negative for strikes between

$$X_L = Se^{bT-\sigma\sqrt{T}\sqrt{4+T\sigma^2}/2}$$

$$X_U = Se^{bT+\sigma\sqrt{T}\sqrt{4+T\sigma^2}/2}$$

and positive for strikes above X_U or below X_L, *ceteris paribus.* In practice, these points will change with other variables and parameters. These levels should therefore be considered good approximations at best.

In general, you want positive DgammaDvol—especially if you don't need to pay for it (flat-volatility smile). In this respect, DgammaDvol actually offers a lot of intuition for how stochastic volatility should affect the BSM values (?). Figure 11 illustrates this point. The DgammaDvol is positive for deep-out-of-the-money options, outside the S_L and S_U interval. For at-the money options and slightly in- or out-of-the-money options, the DgammaDvol is negative.

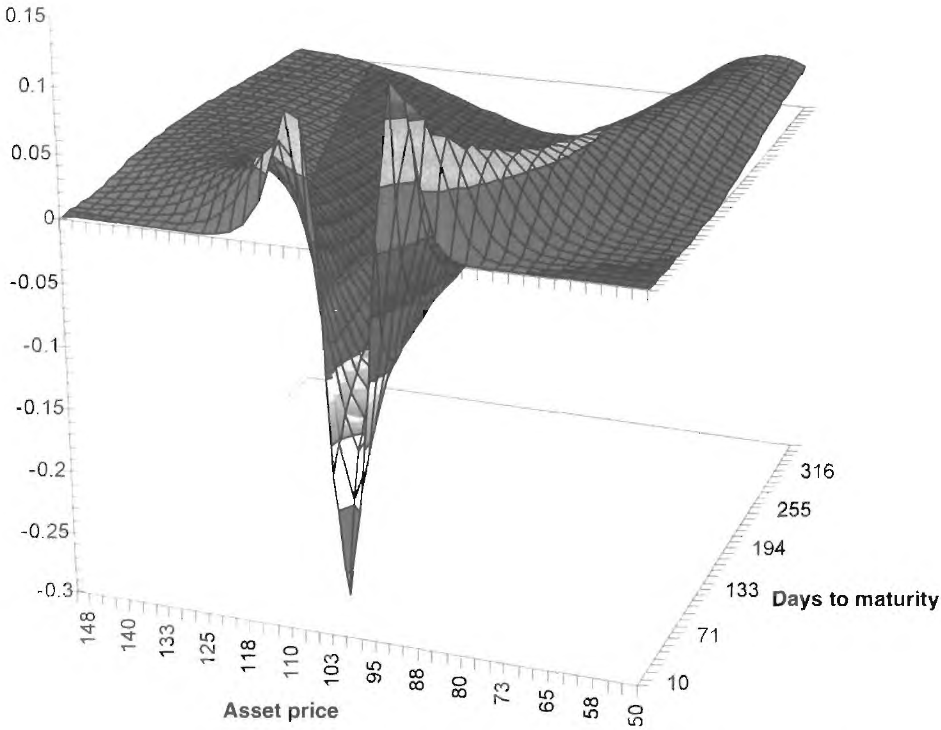


FIGURE 11 DgammaDvol: $X = 100$, $r = 5\%$, $b = 0\%$, $\sigma = 30\%$.

If the volatility is stochastic and uncorrelated with the asset price, then this offers a good indication of which strikes you should use higher/lower implied volatility for when valuing your options. This naturally becomes more complicated in a situation where volatility is correlated with the asset price.

Example
Consider a put option on a futures contract trading at 100, with three months to expiration, strike 80, three-month risk-free interest rate of 5%, and volatility of 26%. What is the DgammaDvol/zomma and also the zommaP? Letting $S = 100$, $X = 80$, $T = 0.25$, $r = 0.05$, $b = 0$, $\sigma = 0.26$, we have

$$d_1 = \frac{\ln(100/80) + 0.26^2/2 \times 0.25}{0.26\sqrt{0.25}} = 1.7815$$

$$d_2 = d_1 - \sigma\sqrt{T} = 1.7815 - 0.26\sqrt{0.25} = 1.6515$$

$$n(d_1) = n(1.7815) = 0.0816$$

$$\Gamma_{\text{call,put}} = \frac{n(d_1)e^{(0-0.05)0.25}}{100 \times 0.26\sqrt{0.25}} = 0.0062$$

$$\text{DgammaDvol}_{\text{call, put}} = \frac{\partial \Gamma}{\partial \sigma} = 0.0062\left(\frac{1.7815 \times 1.6515 - 1}{0.26}\right) = 0.0463$$

In practice, one would typically like to look at the change in gamma for a 1 percent point change in volatility. To do this, we need to to divide by 100. So for a 1 percent point increase (decrease) in volatility, the gamma will increase (decrease) with about 4.63%. To find the DgammaPDvol, we can multiply DgammaDvol with $\frac{S}{100}$. In this particular example, the asset price is 100, so DgammaPDvol will be equal to DgammaDvol.

2.2.6 DgammaDspot, Speed

The third derivative of the option price with respect to spot is known as speed. Speed was probably first mentioned by Garman (1992).

For the BSM formula we get

$$\frac{\partial^3 c}{\partial S^3} = -\frac{\Gamma\left(1 + \frac{d_1}{\sigma\sqrt{T}}\right)}{S} \tag{2.21}$$

A high-speed value indicates that the gamma is very sensitive to changes in the underlying asset. Academics typically claim that third

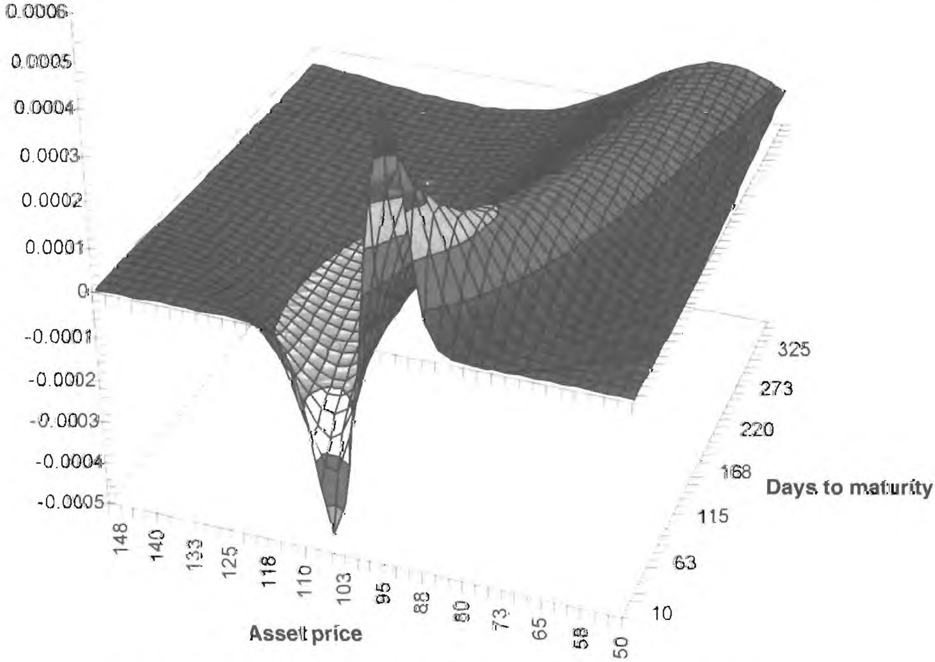


FIGURE 12 Speed: $X = 100$, $r = 5\%$, $b = 0\%$, $\sigma = 30\%$.

or higher order "Greeks" are of no use. For an option trader, on the other hand, it can definitely make sense to have a sense of an option's speed. Interestingly, speed is used by Fouque, Papanicolaou, and Sircar (2000) as a part of a stochastic volatility model adjustment. More to the point, speed is useful when gamma is at its maximum with respect to the asset price. Figure 12 shows the graph of speed with respect to the asset price and time to maturity.

For Γ_P we have an even simpler expression for speed—that is, speedP (speed for percentage gamma)

$$\text{SpeedP} = -\Gamma \frac{d_1}{100\sigma\sqrt{T}} \tag{2.22}$$

Example

Consider an option (call or put) with one month to expiration and a strike of 48 on a stock index contract trading at 50. Assume moreover the one-month risk-free interest rate of 6%, a dividend yield of 5%, and a volatility of 20%. What is the speed and speedP of the option? $S = 50$, $X = 48$, $T = \frac{1}{12} = 0.0833$, $r = 0.06$, $b = 0.06 - 0.05 = 0.01$,

$\sigma = 0.2$, and thus

$$d_1 = \frac{\ln(50/48) + (0.01 + 0.2^2/2) \times 0.0833}{0.2\sqrt{0.0833}} = 0.7504$$

$$n(d_1) = n(0.7504) = 0.3011$$

$$\Gamma_{\text{call,put}} = \frac{n(d_1)e^{(0.01-0.05)\times 0.0833}}{50 \times 0.2\sqrt{0.0833}} = 0.1039$$

$$\frac{\partial^3 c}{\partial S^3} = -\frac{0.1039\left(1 + \frac{0.7504}{0.2\sqrt{0.0833}}\right)}{50} = -0.0291$$

The gamma will thus decrease (increase) with approximately 2.9 percentage points for a unit increase (decrease) in the stock index. The speedP can be found by multiplying the speed by $\frac{S}{100}$, which in this case yields as SpeedP of -0.0135.

2.2.7 DgammaDtime, Color

The change in gamma with respect to small changes in time to maturity, DGammaDtime—also called GammaTheta or color (Garman, 1992)—is given by (assuming we get closer to maturity):

$$-\frac{\partial \Gamma}{\partial T} = \frac{e^{(b-r)T} n(d_1)}{S\sigma\sqrt{T}} \left(r - b + \frac{bd_1}{\sigma\sqrt{T}} + \frac{1 - d_1 d_2}{2T}\right)$$
$$= \Gamma\left(r - b + \frac{bd_1}{\sigma\sqrt{T}} + \frac{1 - d_1 d_2}{2T}\right) \lessgtr 0 \quad (2.23)$$

Divide by 365 to get the sensitivity for a one-day move. In practice, you typically also take into account the expected change in volatility with respect to time. If you, for example, on Friday wonder how your gamma will be on Monday, you typically will assume a higher implied volatility on Monday morning. For Γ_P we have DgammaPDtime

$$-\frac{\partial \Gamma_P}{\partial T} = \Gamma_P\left(r - b + \frac{bd_1}{\sigma\sqrt{T}} + \frac{1 - d_1 d_2}{2T}\right) \lessgtr 0 \quad (2.24)$$

Figure 13 illustrates the DgammaDtime of an option with respect to varying asset price and time to maturity.

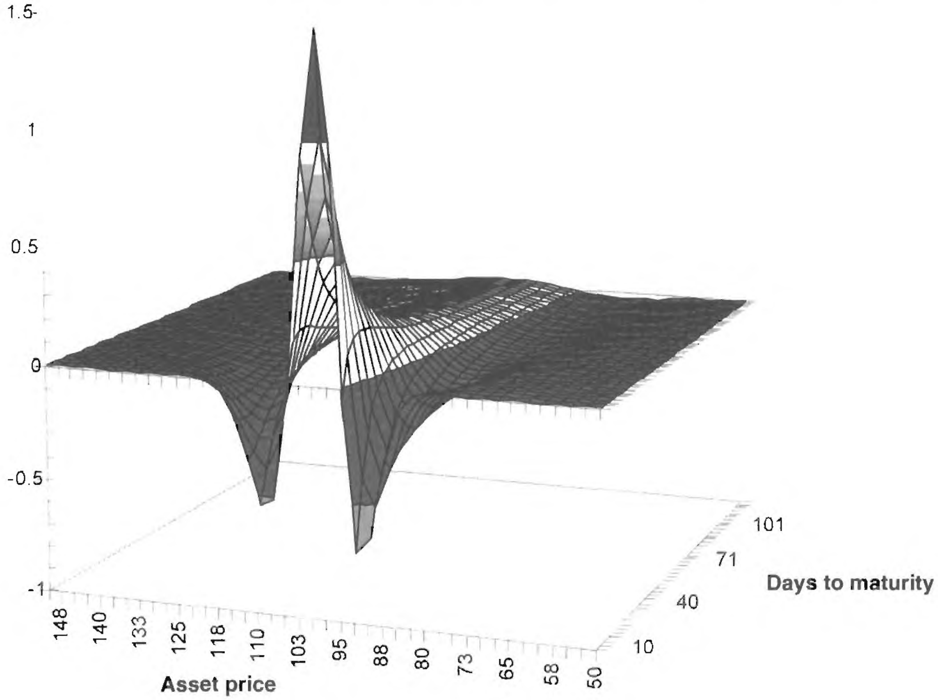


FIGURE 13 DgammaDtime: $X = 100$, $r = 5\%$, $b = 0\%$, $\sigma = 30\%$.

2.3 VEGA GREEKS

2.3.1 Vega

Vega[7] is the option's sensitivity to a small change in the volatility of the underlying asset. Vega is identical for put and call options.

$$\text{Vega}_{\text{call, put}} = \frac{\partial^2 c}{\partial \sigma^2} = \frac{\partial^2 p}{\partial \sigma^2} = Se^{(b-r)T} n(d_1)\sqrt{T} > 0 \tag{2.25}$$

Figure 14 graphs the vega of an option with respect to varying asset price and time to maturity.

Example

Consider a stock index option with nine months to expiration. The stock index price is 55, the strike price is 60, the risk-free interest

[7]While the names of many other options sensitivities have corresponding Greek letters, vega is the name of a star also known as Alpha Lyrae.

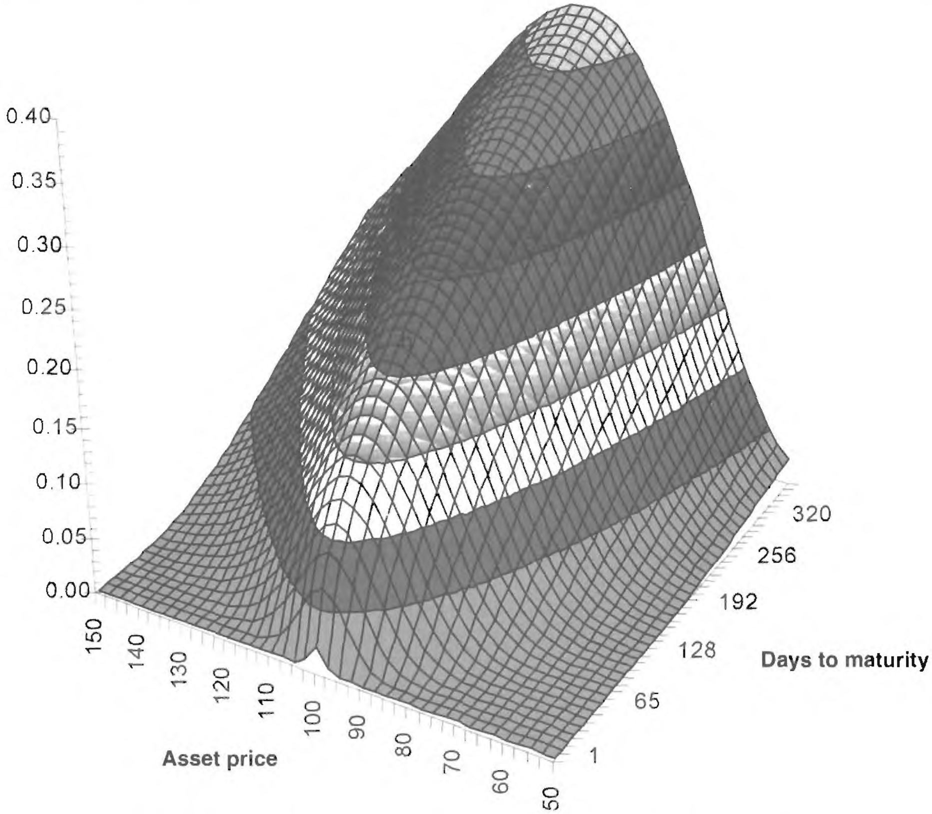


FIGURE 14 Vega: $X = 100$, $r = 5\%$, $b = 5\%$, $\sigma = 30\%$.

rate is 10.50% per year, the dividend yield 3.55%, and the volatility is 30% per year. What is the vega? $S = 55$, $X = 60$, $T = 0.75$, $r = 0.105$, $b = 0.105 - 0.0355 = 0.0695$, $\sigma = 0.3$, which yields

$$d_1 = \frac{\ln(55/60) + (0.0695 + 0.3^2/2)0.75}{0.3\sqrt{0.75}} = -0.0044$$

$$n(d_1) = n(-0.0044) = 0.3989$$

$$\text{Vega}_{\text{call,put}} = 55e^{(0.0695-0.105)0.75} \times 0.3989\sqrt{0.75} = 18.5027$$

To convert this into a vega number for a one-percent point volatility move, we need to divide the vega number by 100. So if the volatility goes from 30% to 31%, the option value will increase by approximately 0.1850.

Vega Local Maximum

When you are trying to profit from moves in implied volatility, it is useful to know where the option attains its maximum vega. For a given strike price, vega attains its maximum when the asset price is

$$S = Xe^{(-b+\sigma^2/2)T}$$

At this asset price we also have in-the-money risk-neutral probability symmetry (treated later in this chapter). Moreover, at this asset price the Black-Scholes-Merton (BSM) formula simplifies to

$$c = Se^{(b-r)T}N(\sigma\sqrt{T}) - \frac{Xe^{-rT}}{2}$$

$$p = \frac{Xe^{-rT}}{2} - Se^{(b-r)T}N(-\sigma\sqrt{T})$$

Similarly, the strike that maximizes vega given the asset price is

$$X = Se^{(b+\sigma^2/2)T}$$

Vega Maximum Time

In the Black-76 model ($b = 0$), vega attains its maximum when the time to maturity is equal to

$$T = \frac{2\left[1 + \sqrt{1 + (8r\frac{1}{\sigma^2} + 1)\ln(S/X)^2}\right]}{8r + \sigma^2}$$

Example

Assume a option on a futures contract trading at 80, with a strike price of 65. The risk-free rate is 5% and the volatility is 30%. For what time does this option have its maximum vega, *ceteris paribus*? $S = 80$, $X = 65$, $r = 0.05$, $b = 0$, $\sigma = 0.3$.

$$T = \frac{2\left[1 + \sqrt{1 + (8 \times 0.05 \times \frac{1}{0.3^2} + 1)\ln(80/65)^2}\right]}{8 \times 0.05 + 0.3^2} = 8.6171$$

Vega Global Maximum

For options with a long time to maturity, the maximum vega is not necessarily increasing with the time to maturity, as many traders believe. Indeed, vega has a global maximum at time

$$T_{\bar{V}} = \frac{1}{2r}$$

and asset price

$$S_{\check{V}} = Xe^{(-b+\sigma^2/2)T_{\check{V}}} = Xe^{\frac{-b+\sigma^2/2}{2r}}.$$

At this global maximum, vega itself, described by Alexander Adamchuk,[8] is equal to the following simple expression:

$$\text{Vega}(S_{\check{V}}, T_{\check{V}}) = \frac{X}{2\sqrt{re\pi}} \tag{2.26}$$

Figure 15 shows the graph of vega with respect to the asset price and time. The intuition behind the vega-top/mountain is that the effect of discounting at some point in time dominates volatility (vega): The lower the interest rate, the lower the effect of discounting, and the higher the relative effect of volatility on the option price. As the

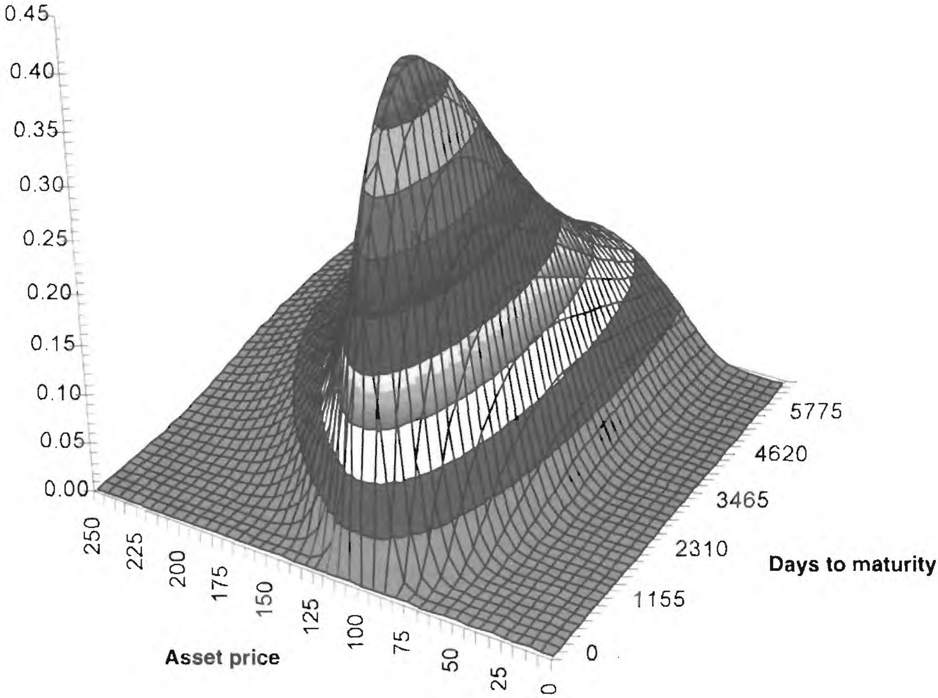


FIGURE 15 Vega: $X = 100$, $r = 15\%$, $b = 0\%$, $\sigma = 12\%$.

[8]Described by Adamchuk at the Wilmott forum located at www.wilmott.com on February 6, 2002.

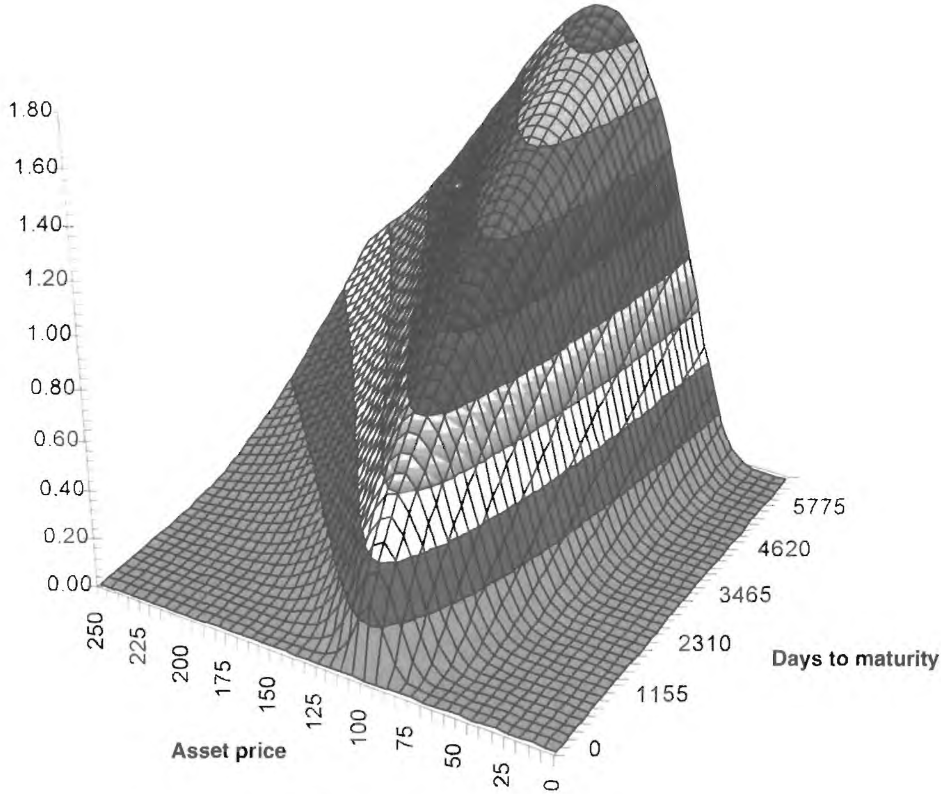


FIGURE 16 Vega: $X = 100$, $r = 0\%$, $b = 0\%$, $\sigma = 12\%$.

risk-free rate goes to zero, the time for the global maximum goes to infinity—that is, we will have no global maximum when the risk-free rate is zero. Figure 16 is the same as Figure 15 but with zero interest rate.

The effect of vega being a decreasing function of time to maturity typically kicks in only for options with very long times to maturity—unless the interest rate is very high. It is not, however, uncommon for caps and floors traders to use the Black-76 formula to compute vegas for options with 10 to 15 years to expiration (caplets).

Example

Consider a stock option with strike 500, risk-free rate 8%, and volatility 40%. For what time and stock price do we have the global maximum vega point, and what is the vega at this point? Thus, $X = 500$, $r = 0.08$,

$b = 0.08$, $\sigma = 0.4$. The time must be (in number of years)

$$T_{\hat{V}} = \frac{1}{2 \times 0.08} = 6.2500$$

and the stock price must be

$$S_{\hat{V}} = 500e^{(-0.08+0.4^2/2)6.2500} = 500$$

and at this stock price and time to maturity, the global maximum vega is

$$\text{Vega}(500, 6.2500) = \frac{500}{2\sqrt{0.08e\pi}} = 302.4634$$

To get the global max vega on the metric of one-percent-point volatility, we need to divide it by 100.

2.3.2 Vega Symmetry

For options with different strikes, we have the following vega symmetry:

$$\text{Vega}(S, X, T, r, b, \sigma) = \frac{X}{Se^{bT}} \text{Vega}\left(S, \frac{(Se^{bT})^2}{X}, T, r, b, \sigma\right) \tag{2.27}$$

As for the gamma symmetry, see Haug (2003). This symmetry is in theory independent of the options being calls or puts.

2.3.3 Vega-Gamma Relationship

Following is a simple and useful relationship between vega and gamma, described by Taleb (1997), among others:

$$\text{Vega} = \Gamma \sigma S^2 T$$

Example

Consider a stock option with nine months to expiration. The stock price is 55, the strike price is 60, the risk-free interest rate is 10% per year, and the volatility is 30% per year. Moreover, the gamma of the option is 0.0278 (as we calculated in the gamma example above). What is the vega? $S = 55$, $X = 60$, $T = 0.75$, $r = 0.1$, $b = 0.1$, $\sigma = 0.3$, $\Gamma = 0.0278$, and

$$\text{Vega} = 0.0278 \times 0.3 \times 55^2 \times 0.75 = 18.9358$$

To look at vega for a one-percent-point move in implied volatility, we need to divide it by 100, so the vega of the option is 0.1894.

2.3.4 Vega from Delta

Given that we know the delta, what is the vega? Vega and delta are related by a simple formula described by Wystrup (2002):

$$\text{Vega} = Se^{(b-r)T}\sqrt{T}n[N^{-1}(e^{(r-b)T}|\Delta|)], \tag{2.28}$$

where $N^{-1}(\cdot)$ is the inverted cumulative normal distribution, $n(\cdot)$ is the normal density function, and Δ is the delta of a call or put option. Using the vega-gamma relationship we can rewrite this relationship to express gamma as a function of the delta

$$\Gamma = \frac{e^{(b-r)T}n[N^{-1}(e^{(r-b)T}|\Delta|)]}{S\sigma\sqrt{T}} \tag{2.29}$$

Relationships, such as those above, between delta and other option sensitivities are particularly useful in the FX options markets, where one often considers a particular delta rather than a strike.

2.3.5 VegaP

The traditional textbook vega measures the dollar change in the option price for a percentage *point* change in volatility. When you are comparing the vega risk of options on different assets, it makes more sense to look at percentage changes in volatility. This metric can be constructed simply by multiplying the standard vega with $\frac{\sigma}{10}$, which gives what is known as vegaP (percentage change in option price for a 10-percent change in volatility):

$$\text{VegaP} = \frac{\sigma}{10}Se^{(b-r)T}n(d_1)\sqrt{T} \geq 0 \tag{2.30}$$

VegaP attains its global maximum at the same asset price and time as for vega. Some options systems use traditional textbook vega, while others use vegaP.

When you are comparing vegas for options with different maturities (calendar spreads), it makes more sense to look at some kind of weighted vega, or alternatively, vega bucketing,[9] because short-term implied volatilities are typically more volatile than long-term implied volatilities. Several options systems implement some type of vega weighting described later in this chapter or vega bucketing. See Haug (1993) and Taleb (1997) for more details.

---

[9]Vega bucketing simply refers to dividing the vega risk into time buckets.

2.3.6 Vega Leverage, Vega Elasticity

The percentage change in option value with respect to percentage point change in volatility is given by

$$\text{VegaLeverage}_{\text{call}} = \text{Vega}\frac{\sigma}{\text{call}} \geq 0 \qquad (2.31)$$

$$\text{VegaLeverage}_{\text{put}} = \text{Vega}\frac{\sigma}{\text{put}} \geq 0 \qquad (2.32)$$

The vega elasticity is highest for out-of-the-money options.

Figure 17 illustrates the vega leverage of a put option.

2.3.7 DvegaDvol, Vomma

DvegaDvol (aka vega convexity, vomma; see Webb, 1999, or Volga) is the sensitivity of vega to changes in implied volatility. Together with DgammaDvol, vomma is in my view one of the most important Greeks.

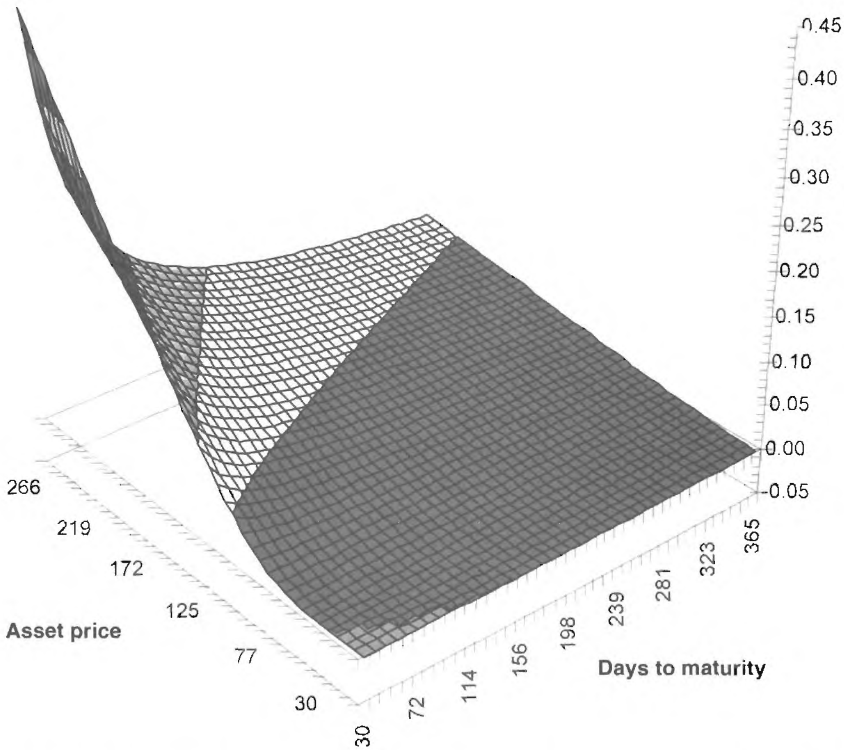


FIGURE 17 Vega leverage put option: $X = 100$, $r = 5\%$, $b = 0\%$, $\sigma = 60\%$.

DvegaDvol is given by

$$\text{DvegaDvol} = \frac{\partial^2 c}{\partial \sigma^2} = \frac{\partial^2 p}{\partial \sigma^2} = \text{Vega}\left(\frac{d_1 d_2}{\sigma}\right) \leq \geq 0 \tag{2.33}$$

For practical purposes, where one "typically" wants to look at vomma for the change of one percentage *point* in the volatility, one should divide vomma by 10,000.

In case of DvegaPDvol, we have

$$\text{DvegaPDvol} = \text{VegaP}\left(\frac{d_1 d_2}{\sigma}\right) \leq \geq 0 \tag{2.34}$$

Options far out-of-the-money have the highest vomma. More precisely, given the strike price, vomma is positive outside the interval

$$(S_L = Xe^{(-b-\sigma^2/2)T}, S_U = Xe^{(-b+\sigma^2/2)T})$$

Given the asset price, the vomma is positive outside the interval

$$(X_L = Se^{(b-\sigma^2/2)T}, X_U = Se^{(b+\sigma^2/2)T})$$

Notice that this is relevant only before conducting the trade.

If you are long options, you typically want to have as high a positive DvegaDvol as possible. If short options, you typically want negative DvegaDvol. Positive DvegaDvol tells you that you will earn more for every percentage point increase in volatility, and if implied volatility is falling, you will lose less and less—that is, you have positive vega convexity.

While DgammaDvol is most relevant for the volatility of the actual volatility of the underlying asset, DvegaDvol is more relevant for the volatility of the implied volatility. Although the volatility of implied volatility and the volatility of actual volatility will typically have high correlation, this is not always the case. DgammaDvol is relevant for traditional dynamic delta hedging under stochastic volatility. DvegaDvol trading has little to do with traditional dynamic delta hedging. DvegaDvol trading is a bet on changes on the price (i.e., changes in implied volatility) for uncertainty in supply and demand, stochastic actual volatility (remember, this is correlated to implied volatility), jumps and any other model risk, and factors that affect the option price but that are not taken into account in the BSM formula. A DvegaDvol trader does not necessarily need to identify the exact reason for the implied volatility to change. If you think the implied volatility will be volatile in the short term, you should typically try to find options with high DvegaDvol. Figure 18 shows the graph of DvegaDvol for changes in asset price and time to maturity.

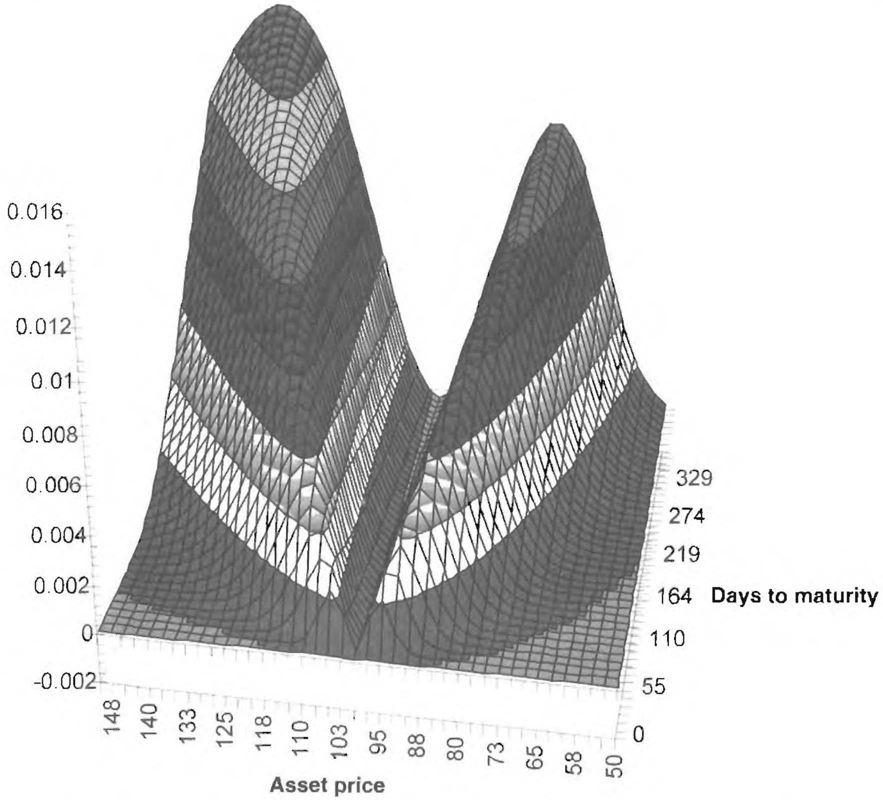


FIGURE 18 DvegaDvol: $X = 100$, $r = 5\%$, $b = 0\%$, $\sigma = 20\%$.

Example

Consider a put option with nine months to expiration and strike 130 on a futures contract trading at 90, and assume moreover the nine-month risk-free interest rate is 5% and volatility is 28%. What is the DvegaDvol/vomma and also the vommaP? With $S = 90$, $X = 130$, $T = 0.75$, $r = 0.05$, $b = 0$, and $\sigma = 0.28$, we have

$$d_1 = \frac{\ln(90/130) + 0.28^2/2 \times 0.75}{0.28\sqrt{0.75}} = -1.3952$$

$$d_2 = d_1 - \sigma\sqrt{T} = -1.3952 - 0.28\sqrt{0.75} = -1.6377$$

$$n(d_1) = n(-1.3952) = 0.1507$$

$$\text{Vega}_{call,put} = 90e^{(0-0.05)0.75} \times 0.1507\sqrt{0.75} = 11.3158$$

$$\text{DvegaDvol} = \frac{\partial^2 c}{\partial \sigma^2} = \frac{\partial^2 p}{\partial \sigma^2} = 11.3158\left(\frac{-1.3952 \times -1.6377}{0.28}\right) = 92.3444$$

For every percentage point increase (decrease) in implied volatility, the vega of the option will increase (decrease) by $\frac{92.3444}{10000} = 0.0092$—that is, we have a positive DvegaDvol/vomma. The DvegaPDvol is given simply by

$$\frac{\sigma}{10}\text{DvegaDvol} = \frac{0.28}{10} \times 92.3444 = 2.5856$$

2.3.8 DvommaDvol, Ultima

The vomma's sensitivity to a change in volatility (ultima) is given by

$$\begin{aligned}\frac{\partial^3 c}{\partial \sigma^3} = \frac{\partial^3 p}{\partial \sigma^3} &= \frac{\partial^2 c}{\sigma^2}\frac{1}{\sigma}\left(d_1 d_2 - \frac{d_1}{d_2} - \frac{d_2}{d_1} - 1\right) \\ &= \text{Vomma}\left(\frac{1}{\sigma}\right)\left(d_1 d_2 - \frac{d_1}{d_2} - \frac{d_2}{d_1} - 1\right) \qquad (2.35)\end{aligned}$$

To get this sensitivity in the metric of a one volatility point move, we have to divide it by 1,000,000.

Figure 19 illustrates DvommaDvol.

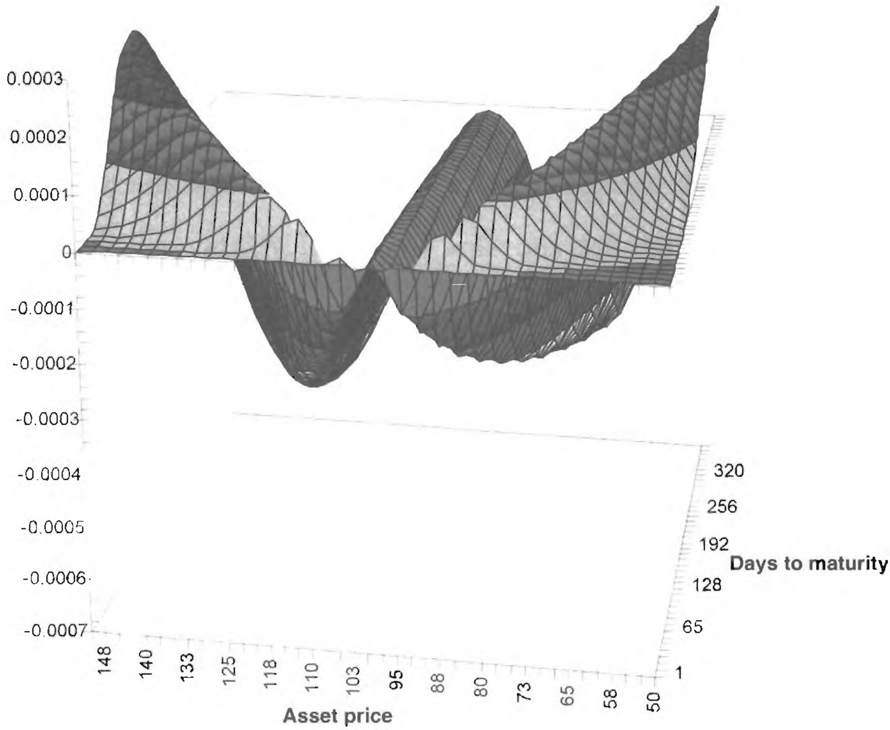


FIGURE 19 DvommaDvol: $X = 100$, $r = 5\%$, $b = 5\%$, $\sigma = 30\%$.

2.3.9 DvegaDtime

DvegaDtime is the change in vega with respect to changes in time. Since we typically are looking at decreasing time to maturity, we express this as minus the partial derivative

$$\text{DvegaDtime} = -\frac{\partial \text{Vega}}{\partial T} = \text{Vega}\left(r - b + \frac{bd_1}{\sigma\sqrt{T}} - \frac{1 + d_1 d_2}{2T}\right) \lessgtr 0 \qquad (2.36)$$

For practical purposes, where one typically wants to express the sensitivity for a one percentage point change in volatility to a one-day change in time, one should divide the DVegaDtime by 36500, or 25200 if you look at trading days only. Figure 20 illustrates DVegaDtime. Figure 21 shows DvegaDtime for a wider range of parameters

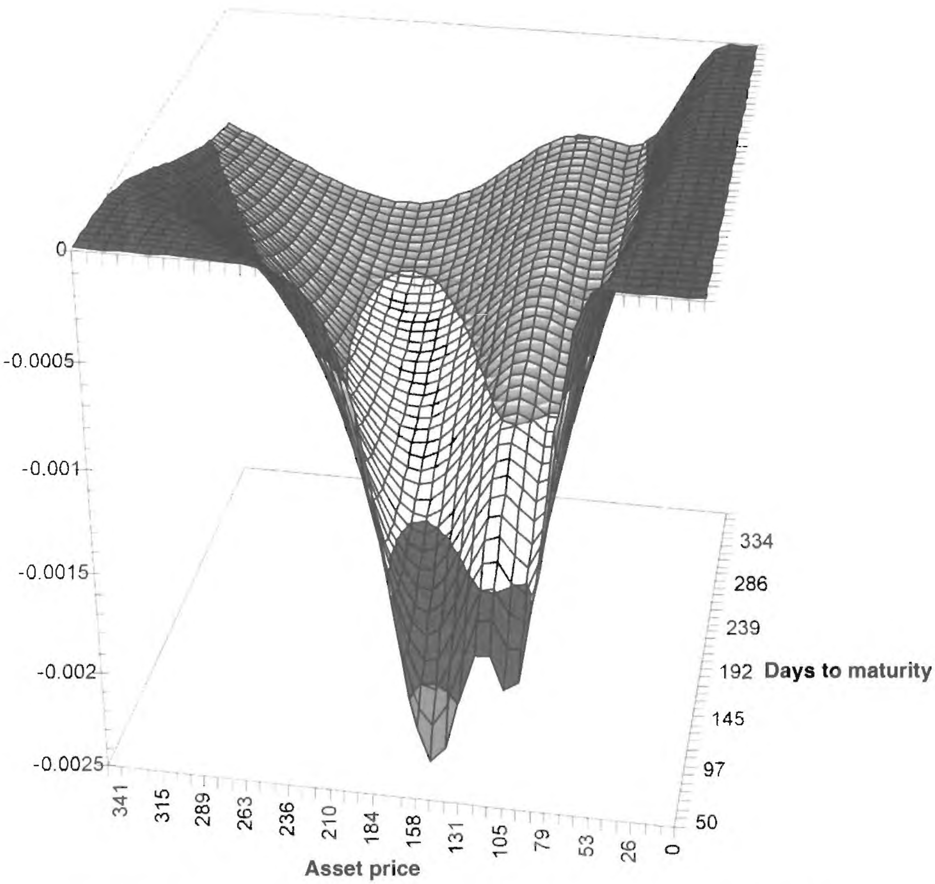


FIGURE 20 DvegaDtime: $X = 100$, $r = 5\%$, $b = 0\%$, $\sigma = 50\%$.

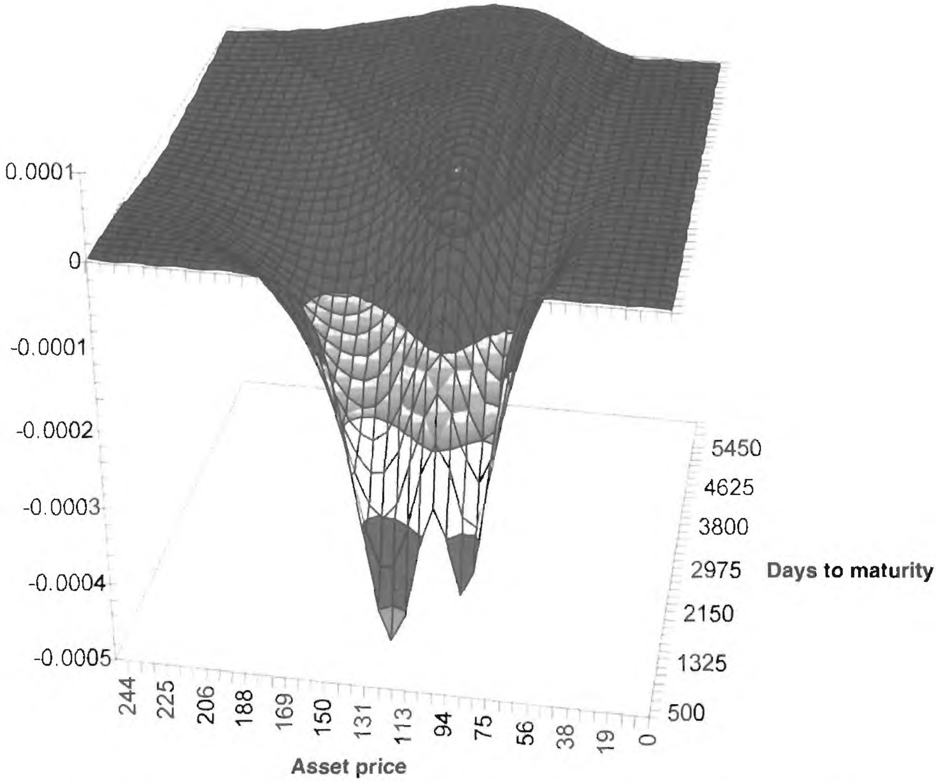


FIGURE 21 DvegaDtime: $X = 100$, $r = 12\%$, $b = 0\%$, $\sigma = 12\%$.

and a lower implied volatility. As expected from Figure 15, we can here see that DvegaDtime can be positive.

2.4 VARIANCE GREEKS

Instead of looking at partial derivatives with respect to the volatility σ, we can look at the sensitivity with respect to the variance $V = \sigma^2$. These sensitivities play a part in several stochastic volatility models—for instance, Hull and White (1987) and Hull and White (1988), covered in Chapter 6: "Black-Scholes-Merton Alternatives."

2.4.1 Variance Vega

Variance vega is the BSM formula's sensitivity to a small change in the variance of the underlying assets' instantaneous rate of return.

Assume the variance is given by V ($V = \sigma^2$)

$$\frac{\partial c}{\partial V} = \frac{\partial p}{\partial V} = Se^{(b-r)T}n(d_1)\frac{\sqrt{T}}{2\sqrt{V}} = Se^{(b-r)T}n(d_1)\frac{\sqrt{T}}{2\sigma} > 0, \tag{2.37}$$

where

$$d_1 = \frac{\ln(S/X) + (b + V/2)T}{\sqrt{VT}} = \frac{\ln(S/X) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$$

The variance vega is simply equal to the standard vega divided by 2σ.

2.4.2 DdeltaDvar

DdeltaDvar is the change in delta for a change in the variance (variance vanna)

$$\frac{\partial^2 c}{\partial S \partial V} = \frac{\partial^2 p}{\partial S \partial V} = -Se^{(b-r)T}n(d_1)\frac{d_2}{2V}$$
$$= -Se^{(b-r)T}n(d_1)\frac{d_2}{2\sigma^2} \tag{2.38}$$

2.4.3 Variance Vomma

Variance vomma is the variance vega's sensitivity to a small change in the variance.

$$\frac{\partial^2 c}{\partial V^2} = \frac{Se^{(b-r)T}\sqrt{T}}{4V^{3/2}}n(d_1)(d_1 d_2 - 1)$$
$$= \frac{Se^{(b-r)T}\sqrt{T}}{4\sigma^3}n(d_1)(d_1 d_2 - 1) \tag{2.39}$$

2.4.4 Variance Ultima

Variance ultima is the BSM formulas' third derivative with respect to variance.

$$\frac{\partial^3 c}{\partial V^3} = \frac{Se^{(b-r)T}\sqrt{T}}{8V^{5/2}}n(d_1)[(d_1 d_2 - 1)(d_1 d_2 - 3) - (d_1^2 + d_2^2)]$$
$$= \frac{Se^{(b-r)T}\sqrt{T}}{8\sigma^5}n(d_1)[(d_1 d_2 - 1)(d_1 d_2 - 3) - (d_1^2 + d_2^2)] \tag{2.40}$$

2.5 VOLATILITY - TIME GREEKS

What if we want to look at the sensitivity of the Black-Scholes-Merton formula with respect to a change in volatility-time, Ω, defined as $\Omega = \sigma\sqrt{T}$—the standard deviation of the rate of return of the underlying asset over a time period T? Assuming zero cost-of-carry and zero interest rate yields

$$\text{Vega}_{VT} = \frac{\partial c}{\partial \Omega} = Sn(d_1),$$

where

$$d_1 = \frac{\ln(S/X) + \Omega^2/2}{\Omega} = \frac{\ln(S/X) + T\sigma^2/2}{\sigma\sqrt{T}}$$

2.6 THETA GREEKS

2.6.1 Theta

Theta is the option's sensitivity to a small change in time to maturity. As time to maturity decreases,[10] it is common to express theta as minus the partial derivative with respect to time.

Call

$$\Theta_{\text{call}} = -\frac{\partial c}{\partial T} = -\frac{Se^{(b-r)T}n(d_1)\sigma}{2\sqrt{T}} - (b-r)Se^{(b-r)T}N(d_1) - rXe^{-rT}N(d_2) \lessgtr 0 \tag{2.41}$$

Put

$$\Theta_{\text{put}} = -\frac{\partial p}{\partial T} = -\frac{Se^{(b-r)T}n(d_1)\sigma}{2\sqrt{T}} + (b-r)Se^{(b-r)T}N(-d_1) + rXe^{-rT}N(-d_2) \lessgtr 0 \tag{2.42}$$

Figures 22 and 23 graph the theta of, respectively, a call and put option.

[10] This is in contrast to the other option sensitivities where the underlying variable can move in either direction, see also Draper (1721).

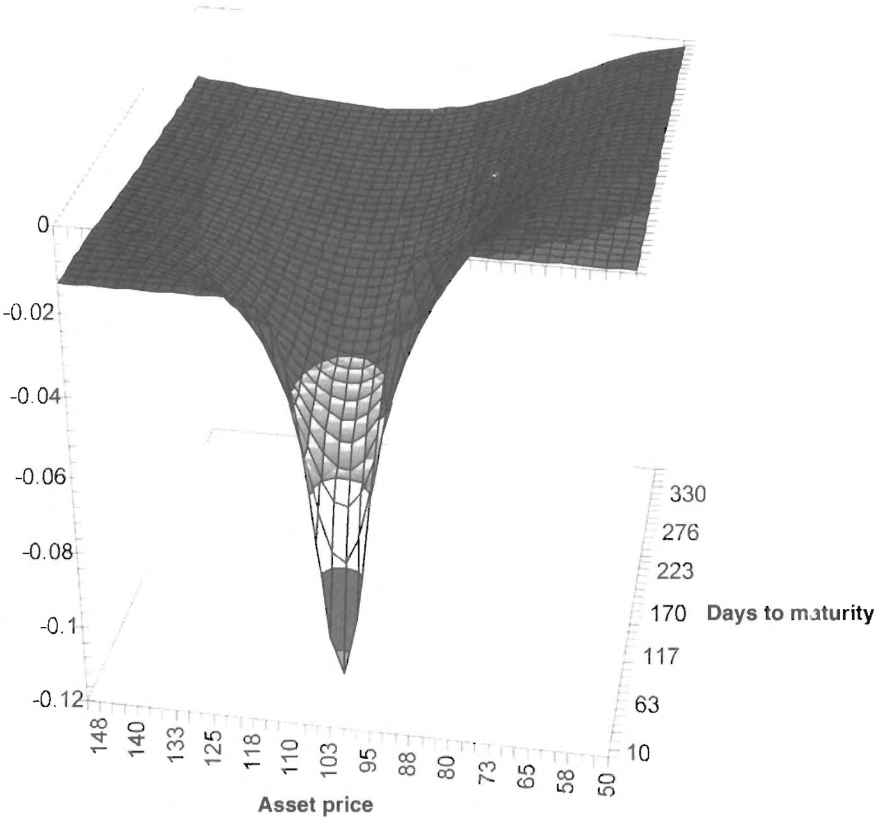


FIGURE 22 Theta call option: $X = 100$, $r = 5\%$, $b = 5\%$, $\sigma = 30\%$.

Example

Consider a European put option on a stock index currently priced at 430. The strike price is 405, the time to expiration is one month, the risk-free interest rate is 7% per year, the dividend yield is 5% per year, and the volatility is 20% per year. $S = 430$, $X = 405$, $T = 0.0833$, $r = 0.07$, $b = 0.07 - 0.05 = 0.02$, and $\sigma = 0.2$ yields

$$d_1 = \frac{\ln(430/405) + (0.02 + 0.2^2/2)0.0833}{0.2\sqrt{0.0833}} = 1.0952$$

$$d_2 = 1.0952 - 0.2\sqrt{0.0833} = 1.0375$$

$$n(d_1) = n(1.0952) = \frac{1}{\sqrt{2\pi}} e^{-1.0952^2/2} = 0.2190$$

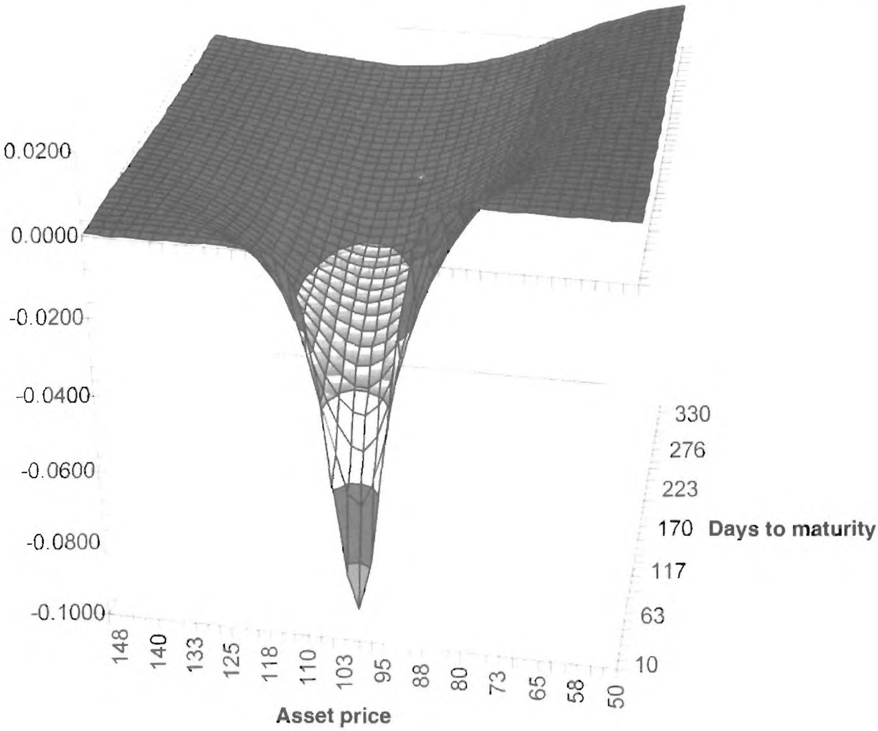


FIGURE 23 Theta put option: $X = 100$, $r = 5\%$, $b = 5\%$, $\sigma = 30\%$.

$$N(-d_1) = N(-1.0952) = 0.1367 \quad N(-d_2) = N(-1.0375) = 0.1498$$

$$\Theta_{\text{put}} = \frac{-430e^{(0.02-0.07)0.0833}n(d_1)0.2}{2\sqrt{0.0833}}$$
$$+ (0.02 - 0.07)430e^{(0.02-0.07)0.0833}N(-d_1)$$
$$+ 0.07 \times 405e^{-0.07\times 0.0833}N(-d_2) = -31.1924$$

Theta for a one-day time decay is thus $-31.1924/365 = -0.0855$

Driftless Theta

In practice, it is often also of interest to know the driftless theta, θ, which measures time decay without taking into account the drift of the underlying or discounting. The driftless theta thereby isolates the effect time decay has on uncertainty, assuming constant volatility.

Uncertainty affects the option through both time and volatility, since the latter is a measure of uncertainty during an infinitesimal time period. We have

$$\theta_{\text{call}} = \theta_{\text{put}} = \theta = -\frac{Sn(d_1)\sigma}{2\sqrt{T}} \leq 0 \tag{2.43}$$

Figure 24 graphs the driftless theta of an option.

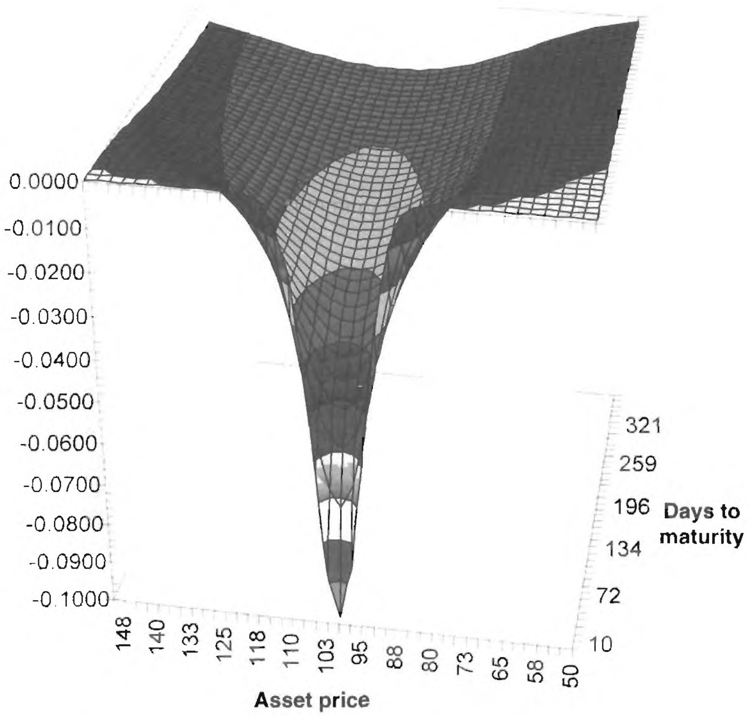


FIGURE 24 Driftless theta: $X = 100$, $r = 0\%$, $b = 0\%$, $\sigma = 30\%$.

Example

What is the driftless theta with the same input parameters as in the theta example above? With $S = 430$, $X = 405$, $T = 0.0833$, $r = 0.07$, $b = 0.07 - 0.05 = 0.02$, and $\sigma = 0.2$, we get

$$d_1 = \frac{\ln(430/405) + (0.02 + 0.2^2/2)0.0833}{0.2\sqrt{0.0833}} = 1.0952$$

$$n(d_1) = n(1.0952) = \frac{1}{\sqrt{2\pi}} e^{-1.0952^2/2} = 0.2190$$

$$\theta_{\text{call}} = \theta_{\text{put}} = \theta = -\frac{430 \times 0.2190 \times 0.2}{2\sqrt{0.0833}} = -32.4862$$

For a one-day move, the driftless theta is $\frac{-32.4862}{365} = -0.0890$.

2.6.2 Theta Symmetry

In the case of driftless theta for options with different strikes, we have the following symmetry that holds for puts and calls:

$$\theta(S, X, T, 0, 0, \sigma) = \frac{X}{S}\theta\left(S, \frac{S^2}{X}, T, 0, 0, \sigma\right) \tag{2.44}$$

Theta-Vega Relationship
Vega and driftless theta have the simple relationship:

$$\theta = -\frac{\text{Vega} \times \sigma}{2T}$$

Bleed-Offset Volatility
A more practical relationship between theta and vega is what is known as bleed-offset volatility. It measures how much the volatility must increase to offset the theta-bleed/time decay. Bleed-offset volatility can be found simply by dividing the one-day theta by vega, $\frac{\Theta}{\text{Vega}}$. In the case of positive theta, you can actually have negative offset volatility. Deep-in-the-money European options can have positive theta, and in this case the offset volatility will be negative.

Theta-Gamma Relationship
There is a simple relationship between driftless gamma and driftless theta:

$$\Gamma = \frac{-2\theta}{S^2\sigma^2}$$

2.7 RHO GREEKS

2.7.1 Rho

Rho is the option's sensitivity to small changes in the risk-free interest rate.

Call

$$\rho_{\text{call}} = \frac{\partial c}{\partial r} = TXe^{-rT}N(d_2) > 0 \tag{2.45}$$

Figure 25 graphs the rho of a call option. In the case of an option on a futures, we have

$$\rho_{\text{call}} = \frac{\partial c}{\partial r} = -Tc < 0 \tag{2.46}$$

Figure 26 graphs the rho of a call option on a futures contract.

Put

$$\rho_{\text{put}} = \frac{\partial p}{\partial r} = -TXe^{-rT}N(-d_2) < 0 \tag{2.47}$$

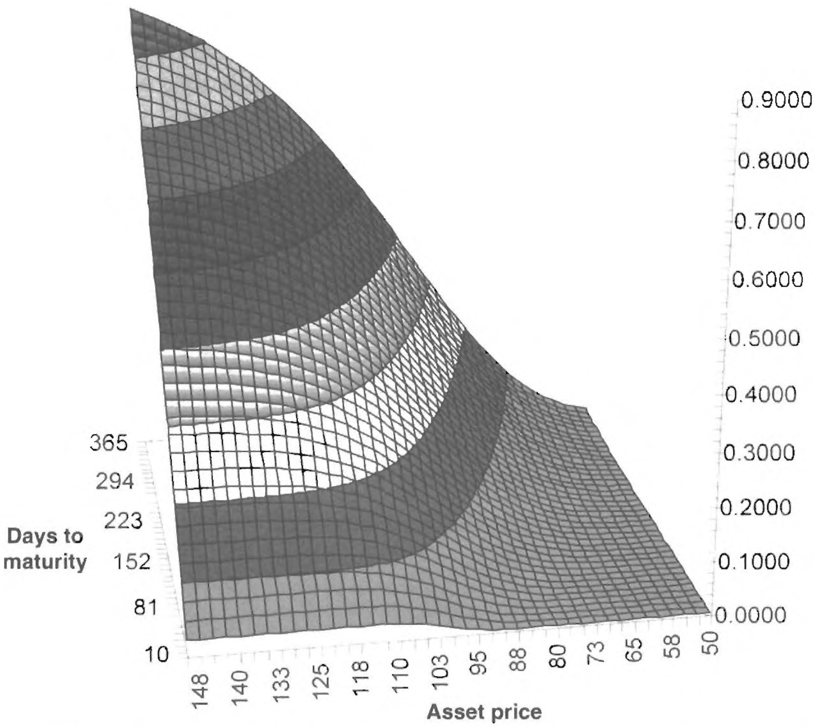


FIGURE 25 Rho call option: $X = 100$, $r = 5\%$, $b = 5\%$, $\sigma = 30\%$.

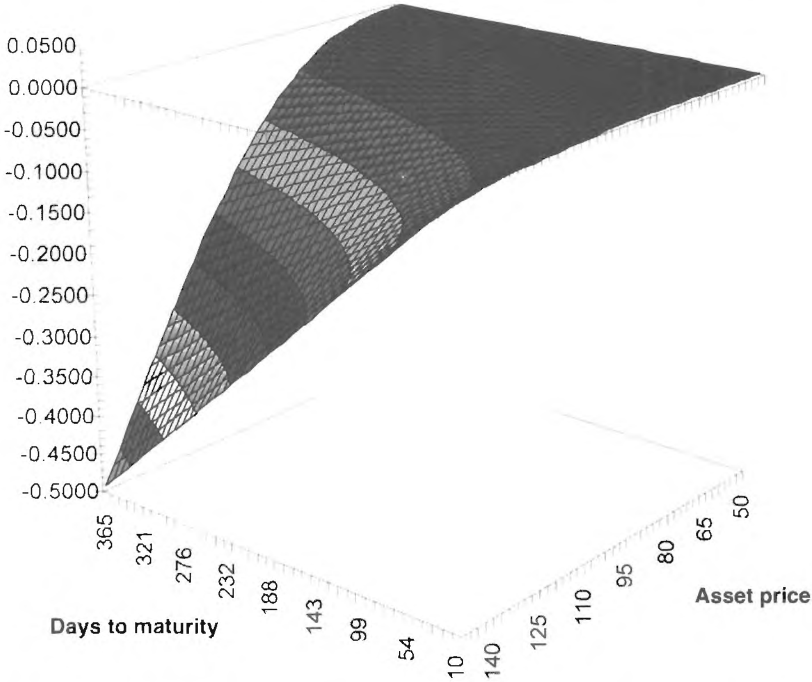


FIGURE 26 Rho call option on futures: $X = 100$, $r = 5\%$, $b = 0\%$, $\sigma = 30\%$.

Figure 27 graphs the rho of a put option. In the case of an option on a futures, we have

$$\rho_{\text{put}} = \frac{\partial c}{\partial r} = -Tp < 0 \tag{2.48}$$

Figure 28 graphs the rho of a put option on a futures contract.

Example

Consider a European call option on a stock currently priced at 72. The strike price is 75, the time to expiration is one year, the risk-free interest rate is 9% per year and the volatility is 19% per year. Thus, $S = 72$, $X = 75$, $T = 1$, $r = 0.09$, $b = 0.09$, $\sigma = 0.19$, and

$$d_2 = \frac{\ln(72/75) + (0.09 - 0.19^2/2)1}{0.19\sqrt{1}} = 0.1638$$

$$N(d_2) = N(0.1638) = 0.5651$$

$$\rho_{\text{call}} = 1 \times 75e^{-0.09\times 1}N(d_2) = 38.7325$$

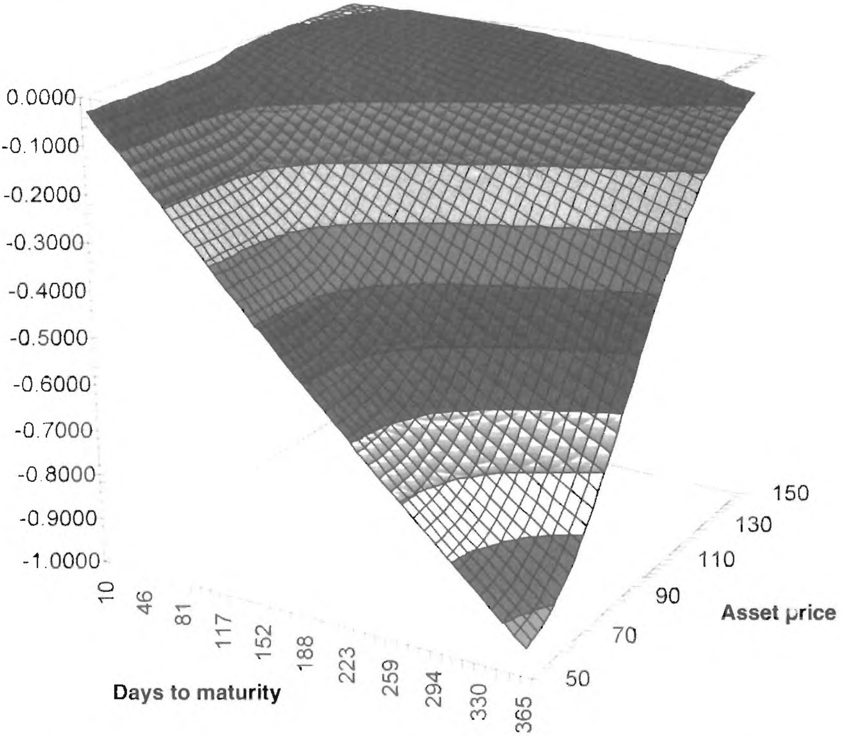


FIGURE 27 Rho put option: $X = 100$, $r = 5\%$, $b = 5\%$, $\sigma = 30\%$.

If the risk-free interest rate goes from 9% to 10%, the call price will thus increase by approximately 0.3873.

2.7.2 Phi/Rho-2

Phi (also known as rho-2) is the option's sensitivity to a change in the dividend yield, or the foreign interest rate in the case of a currency option. For a call it is given by

$$\Phi_{\text{call}} = \frac{\partial c}{\partial q} = -TSe^{(b-r)T}N(d_1) < 0, \tag{2.49}$$

and for a put option we have

$$\Phi_{\text{put}} = \frac{\partial p}{\partial q} = TSe^{(b-r)T}N(-d_1) > 0 \tag{2.50}$$

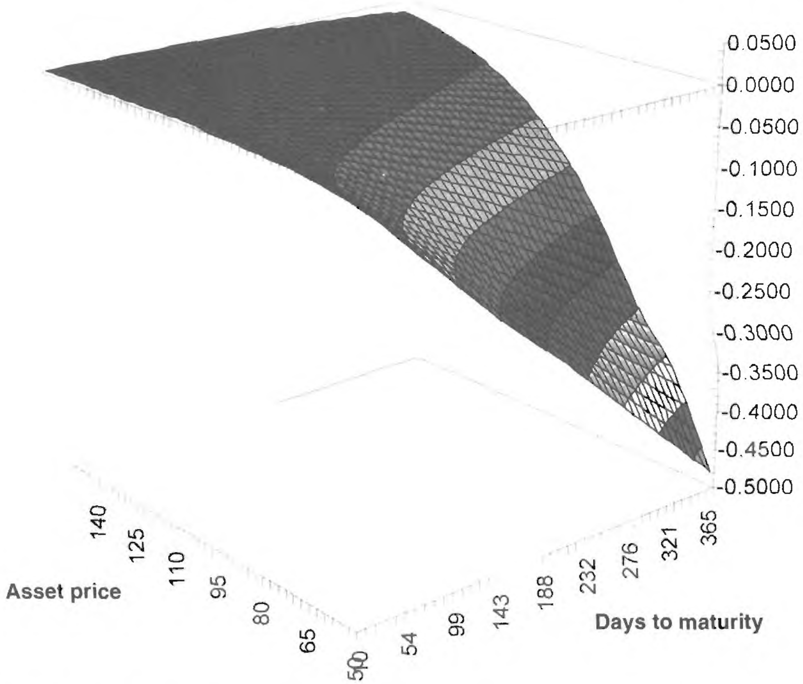


FIGURE 28 Rho put option on futures: $X = 100$, $r = 5\%$, $b = 0\%$, $\sigma = 30\%$.

As discussed in the presentation of the BSM formula (Chapter 1), the dividend yield or foreign interest rate enters the equation indirectly through b.

Figures 29 and 30 graph the phi/rho-2 for, respectively, a call and a put option.

Example

Consider a put option on a stock index currently trading at 733, with six months to expiration, a strike price of 453, a dividend yield of 7.68%, a volatility of the index of 28%, and a risk-free rate of 10.68%. What is the sensitivity of the option to a one-percentage-point change in the dividend yield? With $S = 733$, $X = 453$, $T = 0.5$, $r = 0.1068$, $b = 0.1068 - 0.0768 = 0.03$, and $\sigma = 0.28$, we get

$$d_1 = \frac{\ln(733/453) + (0.03 + 0.28^2/2)0.5}{0.28\sqrt{0.5}} = 2.6055$$

$$N(-d_1) = N(-2.6055) = 0.0046$$

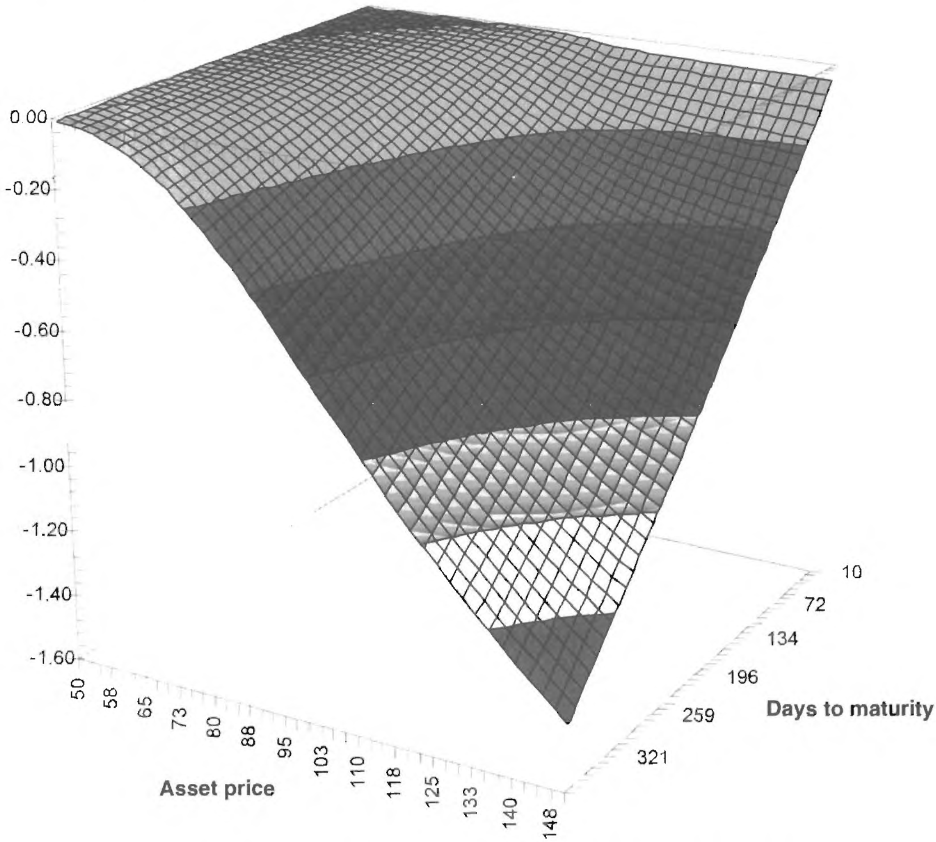


FIGURE 29 Phi call option: $X = 100$, $r = 5\%$, $b = 4\%$, $\sigma = 30\%$.

$$\Phi_{\text{put}} = 0.5 \times 733e^{(0.03-0.1068)0.5}N(-d_1) = 1.6180$$

If the continuous dividend yield increases from 7.68% to 8.68%, the put value thus increases by 0.1618 (1.6180/100). If the dividend yield decreases by the same amount, the option value decreases by 0.1618.

2.7.3 Carry Rho

This is the option's sensitivity to a small change in the cost-of-carry rate.

Cost-of-Carry Call

$$\frac{\partial c}{\partial b} = TSe^{(b-r)T}N(d_1) > 0 \tag{2.51}$$

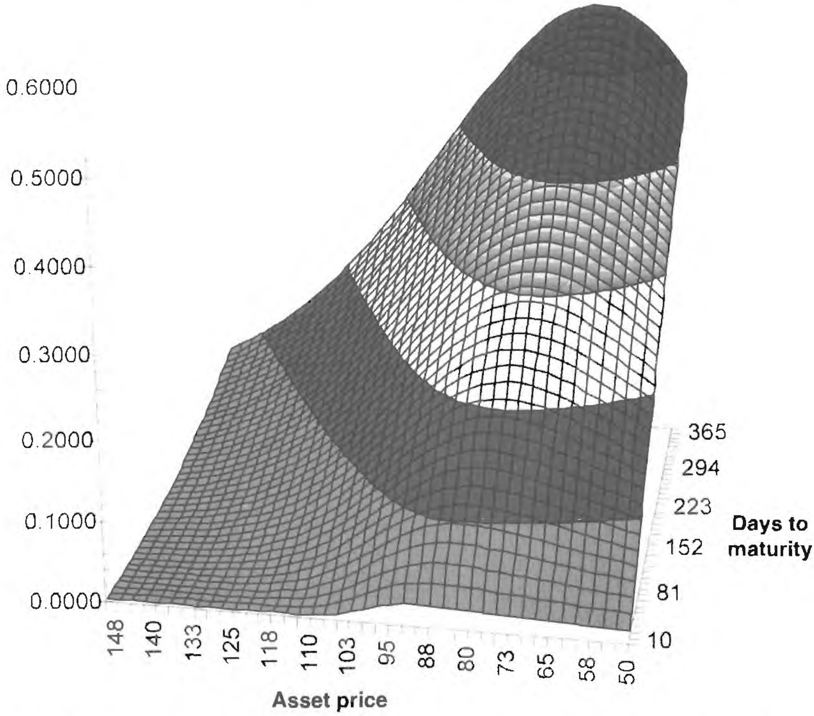


FIGURE 30 Phi put option: $X = 100$, $r = 5\%$, $b = 4\%$, $\sigma = 30\%$.

Cost-of-Carry Put

$$\frac{\partial p}{\partial b} = -TSe^{(b-r)T}N(-d_1) < 0 \tag{2.52}$$

Figures 31 and 32 graph the sensitivity to change in carry for, respectively, a call and a put option.

Example

What is the sensitivity to cost-of-carry for a put option on a stock index with a current price of 500, three months to expiration, a strike price of 490, a risk-free interest rate of 8% per year, a dividend yield of 5% per year, and a volatility of 15% per year? $S = 500$, $X = 490$, $T = 0.25$, $r = 0.08$, $b = 0.08 - 0.05 = 0.03$, $\sigma = 0.15$, and thus

$$d_1 = \frac{\ln(500/490) + (0.03 + 0.15^2/2)0.25}{0.15\sqrt{0.25}} = 0.4069$$

$$N(-d_1) = N(-0.4069) = 0.3421$$

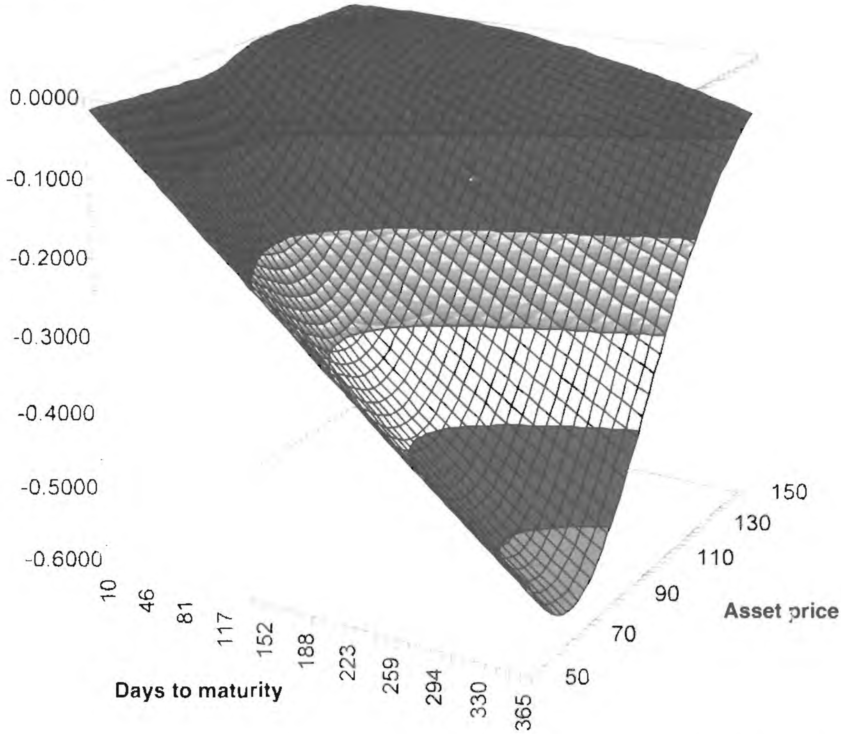


FIGURE 31 Carry call option: $X = 100$, $r = 5\%$, $b = 4\%$, $\sigma = 30\%$.

$$\frac{\partial p}{\partial b} = -0.25 \times 500 e^{(0.03-0.08)0.25} N(-d_1) = -42.2254$$

For a one-percentage-point sensitivity, we have to divide this by 100. If the cost-of-carry should increase from 3% to 4%, then the option value will fall by approximately −0.4223.

2.8 PROBABILITY GREEKS

In this section we look at risk-neutral probabilities in relation to the BSM formula. Keep in mind that such risk-adjusted probabilities could be very different from real-world probabilities.[11]

[11]Risk-neutral probabilities are simply real-world probabilities that have been adjusted for risk. It is therefore not necessary to adjust for risk also in the discount factor for cash flows. This makes it valid to compute market prices as simple expectations of cash flows, with the *risk-adjusted probabilities*, discounted at the *riskless interest rate*—hence the common name "risk-neutral" probabilities, which is somewhat of a misnomer.

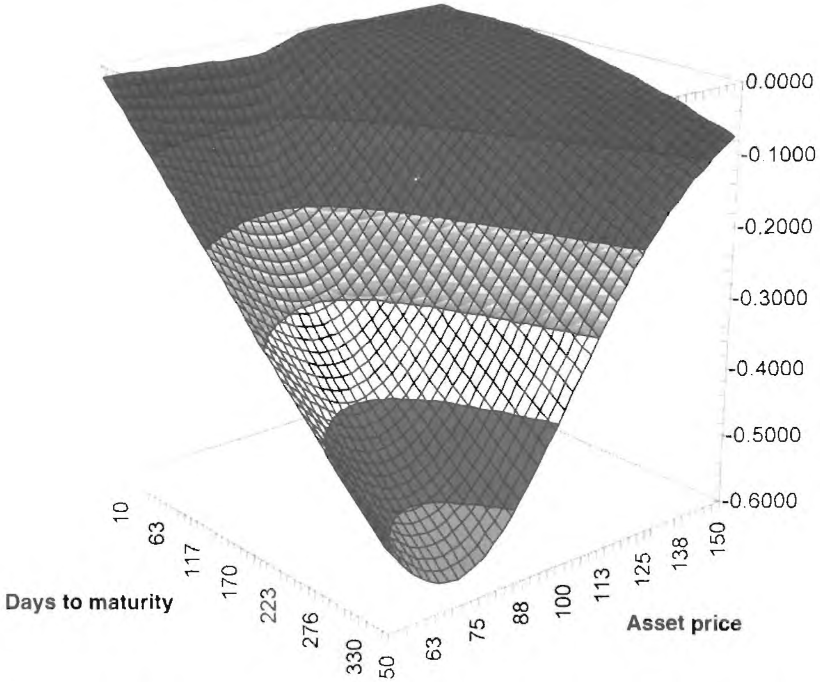


FIGURE 32 Carry put option: $X = 100$, $r = 5\%$, $b = 4\%$, $\sigma = 30\%$.

2.8.1 In-the-Money Probability

In the Black and Scholes (1973)/Merton (1973) model, the risk-neutral probability for a call option finishing in-the-money is

$$\zeta_{\text{call}} = N(d_2) > 0 \tag{2.53}$$

and for a put option

$$\zeta_{\text{put}} = N(-d_2) > 0 \tag{2.54}$$

This is the risk-neutral probability of ending up in-the-money at maturity. It is not identical to the real-world probability of ending up in-the-money. The real probability we simply cannot extract from options prices alone. Figures 33 and 34 graph the risk-neutral probability of ending up in-the-money for, respectively, a call and a put option.

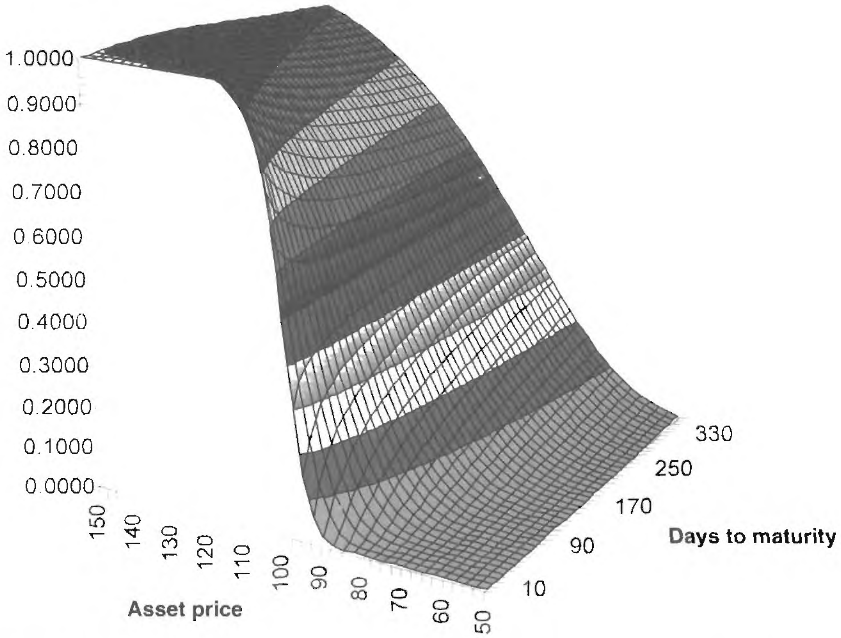


FIGURE 33 In-the-money probability call option: $X = 100$, $r = 5\%$, $b = 4\%$, $\sigma = 30\%$.

A related sensitivity is the strike delta, which is the partial derivatives of the option formula with respect to the strike price

$$\frac{\partial c}{\partial X} = -e^{-rT} N(d_2) > 0 \qquad (2.55)$$

$$\frac{\partial p}{\partial X} = e^{-rT} N(-d_2) > 0 \qquad (2.56)$$

This can be interpreted as the discounted risk-neutral probability of ending up in-the-money (assuming you take the absolute value of the call strike delta).

Example
Consider a put futures option with three months to expiration. The futures price is 100, the strike price is 95, the risk-free interest rate is 8% per year, and the volatility is 12% per year. Thus, $S = 100$, $X = 95$,

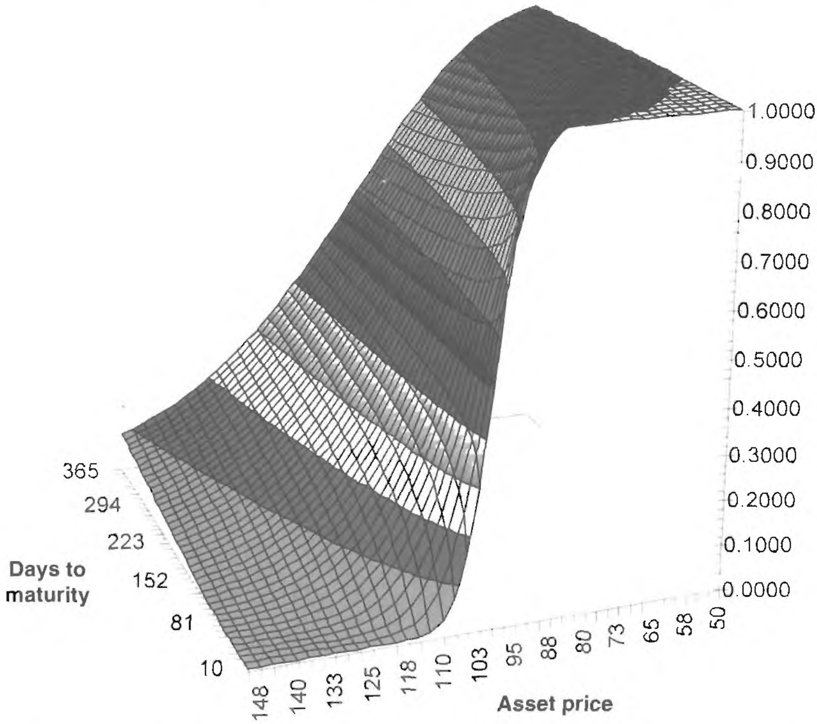


FIGURE 34 In-the-money probability put option: $X = 100$, $r = 5\%$, $b = 4\%$, $\sigma = 30\%$.

$T = 0.25$, $r = 0.08$, $b = 0$, and $\sigma = 0.12$.

$$d_2 = \frac{\ln(100/95) - 0.25 \times 0.12^2/2}{0.12\sqrt{0.25}} = 0.8249$$

$$N(-d_2) = N(-0.8249) = 0.2047$$

That is, the risk-neutral probability for the put to end up in-the-money is about 20.47%. For a call option with the same input parameters, the risk-neutral probability for ending up in the money at maturity is

$$N(d_2) = N(0.8249) = 0.7953$$

Probability Mirror Strikes

For a put and a call to have the same risk-neutral probability of finishing in-the-money, we can find the probability symmetric strikes

$$X_{\text{put}} = \frac{S^2}{X_{\text{call}}} e^{(2b-\sigma^2)T} \qquad X_{\text{call}} = \frac{S^2}{X_{\text{put}}} e^{(2b-\sigma^2)T} \tag{2.57}$$

where X_{put} is the put strike and X_{call} is the call strike. This naturally reduces to $N[d_2(X_{\text{call}})] = N[d_2(X_{\text{put}})]$. A special case is $X_{\text{call}} = X_{\text{put}}$, a probability mirror straddle (probability-neutral straddle). We have this at

$$X_{\text{call}} = X_{\text{put}} = Se^{(b-\sigma^2/2)T}$$

At this point the risk-neutral probability of ending up in-the-money is 0.5 for both the put and the call. Standard puts and calls will not have the same value at this point. The same value for a put and a call occurs when the options are at-the-money forward, $X = S^{bT}$. However, for a cash-or-nothing option (see Chapter 4), we will also have value-symmetry for puts and calls at the risk-neutral probability strike. Moreover, at the probability-neutral straddle, we will also have vega symmetry as well as zero vomma.

Strikes from Probability

Another interesting formula returns the strike of an option, given the risk-neutral probability p_i of ending up in-the-money. The strike of a call is given by

$$X_c = S\exp[-N^{-1}(p_i)\sigma\sqrt{T} + (b - \sigma^2/2)T], \tag{2.58}$$

where $N^{-1}(\cdot)$ is the inverse cumulative normal distribution described in Chapter 13. The strike for a put is given by

$$X_p = S\exp[N^{-1}(p_i)\sigma\sqrt{T} + (b - \sigma^2/2)T] \tag{2.59}$$

2.8.2 DzetaDvol

Zeta's sensitivity to a small change in the implied volatility is given by

$$\frac{\partial \zeta_{\text{call}}}{\partial \sigma} = -n(d_2)\left(\frac{d_1}{\sigma}\right) \lessgtr 0 \tag{2.60}$$

and for a put

$$\frac{\partial \zeta_{\text{put}}}{\partial \sigma} = n(d_2)\left(\frac{d_1}{\sigma}\right) \lessgtr 0 \tag{2.61}$$

Divide by 100 to get the associated measure for percentage point volatility changes.

2.8.3 DzetaDtime

The in-the-money risk-neutral probability's sensitivity to moving closer to maturity is given by

$$-\frac{\partial \zeta_{\text{call}}}{\partial T} = n(d_2)\left(\frac{b}{\sigma\sqrt{T}} - \frac{d_1}{2T}\right) \leq\geq 0 \tag{2.62}$$

and for a put

$$-\frac{\partial \zeta_{\text{put}}}{\partial T} = -n(d_2)\left(\frac{b}{\sigma\sqrt{T}} - \frac{d_1}{2T}\right) \leq\geq 0 \tag{2.63}$$

Divide by 365 to get the sensitivity for a one-day move.

2.8.4 Risk-Neutral Probability Density

The second-order partial derivative of the BSM formula with respect to the strike price yields the risk-neutral probability density of the underlying asset; see Breeden and Litzenberger (1978) (aka the strike gamma).

$$\text{RND} = \frac{\partial^2 c}{\partial X^2} = \frac{\partial^2 p}{\partial X^2} = \frac{n(d_2)e^{-rT}}{X\sigma\sqrt{T}} \geq 0 \tag{2.64}$$

Figure 35 illustrates the risk-neutral probability density with respect to time and asset price. With the same volatility for any asset price, this is naturally the lognormal distribution of the asset price.

2.8.5 From in-the-Money Probability to Density

Given the in-the-money risk-neutral probability, p_i, the risk-neutral probability density is given by

$$\text{RND} = \frac{e^{-rT}n[N^{-1}(p_i)]}{X\sigma\sqrt{T}}, \tag{2.65}$$

where $n(\cdot)$ is the normal density given in Chapter 13.

2.8.6 Probability of Ever Getting in-the-Money

For in-the-money options the probability of ever getting in-the-money (hitting the strike) before maturity naturally equals unity. The risk-neutral probability for an out-of-the-money call ever getting in-the-money is

$$p_c = (X/S)^{\mu+\lambda}N(-z) + (X/S)^{\mu-\lambda}N(-z + 2\lambda\sigma\sqrt{T}) \tag{2.66}$$

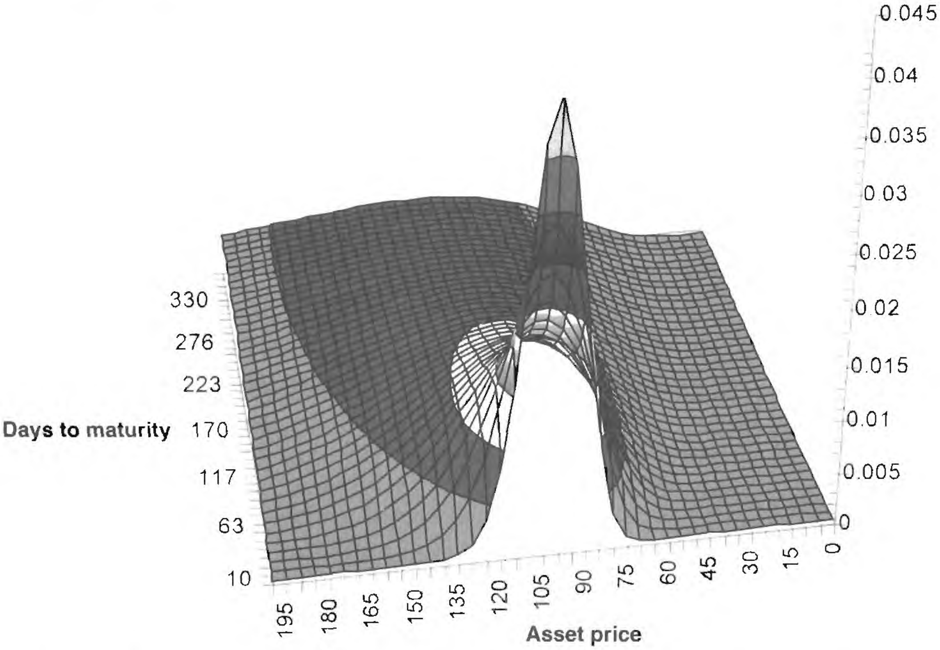


FIGURE 35 Risk-neutral-density: $X = 100$, $r = 5\%$, $b = 0\%$, $\sigma = 20\%$.

Similarly, the risk-neutral probability for an out-of-the-money put ever getting in-the-money (hitting the strike) before maturity is

$$p_p = (X/S)^{\mu+\lambda} N(z) + (X/S)^{\mu-\lambda} N(z - 2\lambda\sigma\sqrt{T}), \qquad (2.67)$$

where

$$z = \frac{\ln(X/S)}{\sigma\sqrt{T}} + \lambda\sigma\sqrt{T} \qquad \mu = \frac{b - \sigma^2/2}{\sigma^2} \qquad \lambda = \sqrt{\mu^2 + \frac{2r}{\sigma^2}}$$

This is equal to the barrier hit probability used for computing the value of a rebate, developed by Reiner and Rubinstein (1991a), described in Chapter 4. Alternatively, the probability of ever getting in-the-money before maturity can be calculated in a very simple way in a binomial tree, using Brownian bridge probabilities.

2.9 GREEKS AGGREGATIONS

When adding up option sensitivities from different options, we must be careful. In the case of options on the same underlying asset, we can typically add together the deltas and gammas and thetas. When it comes to any sensitivity related to volatility, we must also be careful, because implied volatilities from options with different maturities do not move parallel; in general, short-term volatilities tend to move much more around than long-term volatilities. Most experienced traders for this reason use some type of weighting of the vega. The same is true with anything related to sensitivities to interest-rate moves; for this reason, one also needs to do similar weighting for rho and phi.

2.9.1 Net Weighted Vega Exposure

In the formula for net weighted vega, Haug (1993) takes into account the fact that the term structure of volatility does not shift in a parallel fashion. This can be useful when adding the vega risk of a portfolio of options on the same underlying security where the time to maturity of the options differ.

$$NWV = \sum_{T=1}^{m} \sum_{i=1}^{n} Q_{i,T} \text{vega}_{i,T} \frac{\Psi_T}{\Psi_R} \rho_{\sigma(T),\sigma(R)}, \tag{2.68}$$

where

m = Number of different maturities in the option portfolio.
n = Number of different strikes with time to maturity T.
$\text{vega}_{i,T}$ = Vega value of an option with strike i and time to maturity T.
$Q_{i,T}$ = Number or quantity of options in the portfolio with vega $_{i,T}$.
Ψ_T = Volatility of volatility with time to maturity T.
Ψ_R = Volatility of reference volatility.
$\rho_{\sigma(T),\sigma(R)}$ = Correlation between the volatility with time to maturity T and the reference volatility.

Example

Let us assume that we own the portfolio shown in Table 2-2. How will the portfolio react to shifts in the term structure of volatility? To calculate *NWV*, we need estimates of future volatility of volatilities with different maturities, and correlation coefficients between different volatilities. We chose a reference volatility equal to the volatility of the option with the longest time to maturity, that is, 120-day volatility today, 119-day volatility tomorrow, and so on. Assume we have calculated the following historical volatilities of volatilities: 6.5 percentage points 30-day, 5.5 percentage points 60-day, 4.0 percentage points

TABLE 2-2

Option Portfolio

($S = 100, \sigma = 0.25, r = 0.1, b = 0.1$)

Days to maturity	120	60	60	30
Strike	105.00	85.00	100.00	100.00
Call price	4.99	16.53	4.88	3.27
Vega	22.86	3.11	15.81	11.31
Number of contracts	450	100	−400	−300
Volatility of volatility	4.00%	5.50%	5.50%	6.50%
Correlation coefficients	1.00	0.85	0.85	0.65

Espen Gaarder Haug, "Opportunities and Perils of Using Option Sensitivities," *Journal of Financial Engineering,* vol. 2, no. 3, September 1993. Used by permission.

120-day, and correlation coefficients of 0.65 between 30-day volatility and the reference volatility (120-day), 0.85 for the 60-day volatility and, naturally, 1.0 between 120-day volatility and the reference volatility. It follows that $\Psi_1 = 6.5$, $\Psi_2 = 5.5$, $\Psi_3 = 4.0$, $\Psi_R = 4.0$, $\rho_{1,R} = 0.65$, $\rho_{2,R} = 0.85$, $\rho_{3,R} = 1.0$.

$$\begin{aligned} NWV &= \sum_{T=1}^{m}\sum_{i=1}^{n} Q_{i,T}\,\text{vega}_{i,T}\frac{\Psi_T}{\Psi_R}\rho_{\sigma(T),\sigma(R)} \\ &= -300 \times 11.31 \times \frac{6.5}{4.0} \times 0.65 - 400 \times 15.81 \times \frac{5.5}{4.0} \times 0.85 \\ &\quad + 100 \times 3.11 \times \frac{5.5}{4.0} \times 0.85 + 450 \times 22.86 \times \frac{4.0}{4.0} \times 1.0 \\ &= -324.55 \end{aligned}$$

Hence, for each percentage point rise in the reference volatility (120-day), we will lose approximately \$325.

Application

Several investment banks and commercial derivatives software systems use some form of weighted vega measure. There are several variants of how to compute the weighted measure, some preferring to use weights based on historical data, others preferring to specify their own subjective weights based on their expectations. Taleb (1997) also describes several ways to calculate weighted vega.

2.10 AT-THE-MONEY FORWARD APPROXIMATIONS

The at-the-money forward approximations published by Brenner and Subrahmanyam (1994) can be used for options that are at-the-money forward. To be at-the-money forward is defined as $S = Xe^{-bT}$ or $X = Se^{bT} = F$ (strike price equals the forward price of the underlying asset).

2.10.1 Approximation of the Black-Scholes-Merton Formula

$$c \approx 0.4Se^{(b-r)T}\sigma\sqrt{T}, \quad p \approx 0.4Se^{(b-r)T}\sigma\sqrt{T} \tag{2.69}$$

Example

Consider a put option with three months to expiration. The futures price is 70, the strike price is 70, the risk-free interest rate is 5% per year, the cost-of-carry is 0%, and the volatility is 28% per year. $S = 70$, $X = 70$, $T = 0.25$, $r = 0.05$, $b = 0$, $\sigma = 0.28$, and

$$p \approx 0.4 \times 70e^{(0-0.05)T}0.28\sqrt{0.25} \approx 3.8713$$

For comparison, the exact Black-76 price is 3.8579.

2.10.2 Delta

$$\Delta_{\text{call}} \approx e^{(b-r)T}(1/2 + 0.2\sigma\sqrt{T}), \quad \Delta_{\text{put}} \approx e^{(b-r)T}(0.2\sigma\sqrt{T} - 1/2) \tag{2.70}$$

2.10.3 Gamma

$$\Gamma_{\text{call,put}} \approx \frac{e^{(b-r)T}0.4}{S\sigma\sqrt{T}} \tag{2.71}$$

2.10.4 Vega

$$\text{Vega}_{\text{call,put}} \approx Se^{(b-r)T}0.4\sqrt{T} \tag{2.72}$$

2.10.5 Theta

$$\Theta_{\text{call}} \approx -\frac{Se^{(b-r)T}0.4\sigma}{2\sqrt{T}} - Se^{(b-r)T}[b(1/2 + 0.2\sigma\sqrt{T}) - 0.4r\sigma\sqrt{T}] \tag{2.73}$$

$$\Theta_{\text{put}} \approx -\frac{Se^{(b-r)T}0.4\sigma}{2\sqrt{T}} + Se^{(b-r)T}[b(1/2 - 0.2\sigma\sqrt{T}) + 0.4r\sigma\sqrt{T}] \tag{2.74}$$

2.10.6 Rho

$$\rho_{\text{call}} \approx TXe^{-rT}(1/2 - 0.2\sigma\sqrt{T}) \tag{2.75}$$

$$\rho_{\text{put}} \approx -TXe^{-rT}(1/2 + 0.2\sigma\sqrt{T}) \tag{2.76}$$

For options on futures:

$$\rho_{\text{call}} = -Tc \tag{2.77}$$

$$\rho_{\text{put}} = -Tp \tag{2.78}$$

2.10.7 Cost-of-Carry

$$\frac{\partial c}{\partial b} \approx TSe^{(b-r)T}(1/2 + 0.2\sigma\sqrt{T}) \tag{2.79}$$

$$\frac{\partial p}{\partial b} \approx -TSe^{(b-r)T}(1/2 - 0.2\sigma\sqrt{T}) \tag{2.80}$$

2.11 NUMERICAL GREEKS

So far we have looked only at analytical Greeks. A frequently used alternative is to use numerical Greeks, also known as finite difference approximations. One of the main strength of numerical Greeks is that their computation is independent of the model under consideration. The finite difference approximation will give us the Greeks we need as long as we have an accurate model to compute the value of the derivative.

2.11.1 First-Order Greeks

First-order partial derivatives, $\frac{\partial f(x)}{\partial x}$, can be approximated by the two-sided finite difference method:

$$\frac{c(S + \Delta S, X, T, r, b, \sigma) - c(S - \Delta S, X, T, r, b, \sigma)}{2\Delta S} \tag{2.81}$$

In the case of derivatives with respect to time, we know what direction time will move and it is more accurate (for what is happening in the "real" world) to use a backward derivative (a one-sided finite difference):

$$\Theta \approx \frac{c(S, X, T, r, b, \sigma) - c(S, X, T - \Delta T, r, b, \sigma)}{\Delta T} \tag{2.82}$$

Numerical Greeks have several advantages over analytical ones. If, for instance, we have a sticky delta volatility smile, then we can change the volatilities accordingly when calculating the numerical delta. (We have a sticky delta volatility smile when the shape of the volatility smile sticks to the deltas but not to the strike; in other words, the volatility for a given strike will move as the underlying moves.)

$$\Delta_{\text{call}} \approx \frac{c(S+\Delta S, X, T, r, b, \sigma_1) - c(S-\Delta S, X, T, r, b, \sigma_2)}{2\Delta S} \tag{2.83}$$

Numerical Greeks are moreover model-independent, while the analytical Greeks presented above are specific to the BSM model.

2.11.2 Second-Order Greeks

For gamma and other second-order derivatives, $\frac{\partial^2 f(x)}{\partial x^2}$, (for example, gamma or DvegaDvol), we can use the central finite difference method:

$$\Gamma \approx \frac{c(S+\Delta S, \ldots) - 2c(S, \ldots) + c(S-\Delta S, \ldots)}{\Delta S^2} \tag{2.84}$$

If the option is very close to maturity (a few hours) and approximately at-the-money, the analytical gamma can approach ∞, which is naturally an illusion of your real risk. The reason is simply that analytical partial derivatives are accurate only for infinite small changes, while in practice one sees only discrete changes. The numerical gamma solves this problem and offers a more accurate gamma in these cases. This is particularly true with regard to barrier options; see Taleb (1997).

2.11.3 Third-Order Greeks

For speed and other third-order derivatives, $\frac{\partial^3 f(x)}{\partial x^3}$, we can, for example, use the following approximation:

$$\begin{aligned}\text{Speed} \approx \frac{1}{\Delta S^3}[&c(S+2\Delta S, \ldots) - 3c(S+\Delta S, \ldots) \\ &+ 3c(S, \ldots) - c(S-\Delta S, \ldots)]\end{aligned} \tag{2.85}$$

2.11.4 Mixed Greeks

What about mixed derivatives, $\frac{\partial f(x,y)}{\partial x \partial y}$, for example, DdeltaDvol and charm? These can be calculated numerically by

$$\text{DdeltaDvol} \approx \frac{1}{4\Delta S \Delta\sigma} \times [c(S+\Delta S,\ldots,\sigma+\Delta\sigma) - c(S+\Delta S,\ldots,\sigma-\Delta\sigma) - c(S-\Delta S,\ldots,\sigma+\Delta\sigma) + c(S-\Delta S,\ldots,\sigma-\Delta\sigma)] \quad (2.86)$$

In the case of DdeltaDvol, one would typically divide it by 100 to get the "right" notation—that is, for a one-point change in volatility.

2.11.5 Third-Order Mixed Greeks

In the case of Greeks like DgammaDvol, we need the third-order mixed Greek, $\frac{\partial^3 f(x,y)}{\partial x^2 \partial y}$:

$$\text{DgammaDvol} \approx \frac{1}{2\Delta\sigma \Delta S^2} \times [c(S+\Delta S,\ldots,\sigma+\Delta\sigma) - 2c(S,\ldots,\sigma+\Delta\sigma) + c(S-\Delta S,\ldots,\sigma+\Delta\sigma) - c(S+\Delta S,\ldots,\sigma-\Delta\sigma) + 2c(S,\ldots,\sigma-\Delta\sigma) - c(S-\Delta S,\ldots,\sigma-\Delta\sigma)] \quad (2.87)$$

In the case of DgammaDvol, one would again typically divide it by 100 to get the "right" notation.

Computer algorithm

The code below illustrates how to use numerical Greeks to calculate the sensitivities of the Black-Scholes-Merton formula. The dS variable is equivalent to ΔS in the formulas above. The user can supply its value or let it default to 0.01. The function calls the *GBlackScholes*(·) algorithm described in Chapter 1.

```
Function GBlackScholesNGreeks(OutPutFlag As String, CallPutFlag As String, _
S As Double, X As Double, T As Double,  r As Double, b As Double, _
v As Double, Optional dS)

    If IsMissing(dS) Then
        dS = 0.01
    End If


    If OutPutFlag = "p" Then ' Value
        GBlackScholesNGreeks = GBlackScholes(CallPutFlag, S, X, T, r, b, v)
    ElseIf OutPutFlag = "d" Then 'Delta
```

```
    GBlackScholesNGreeks = (GBlackScholes(CallPutFlag, S + dS, X, T, r, b, v) _
- GBlackScholes(CallPutFlag, S - dS, X, T, r, b, v)) / (2 * dS)
ElseIf OutPutFlag = "e" Then 'Elasticity
    GBlackScholesNGreeks = (GBlackScholes(CallPutFlag, S + dS, X, T, r, b, v) _
     - GBlackScholes(CallPutFlag, S - dS, X, T, r, b, v)) / (2 * dS) _
      * S / GBlackScholes(CallPutFlag, S, X, T, r, b, v)
ElseIf OutPutFlag = "g" Then 'Gamma
    GBlackScholesNGreeks = (GBlackScholes(CallPutFlag, S + dS, X, T, r, b, v) _
     - 2 * GBlackScholes(CallPutFlag, S, X, T, r, b, v) _
      + GBlackScholes(CallPutFlag, S - dS, X, T, r, b, v)) / dS^2
ElseIf OutPutFlag = "gv" Then 'DGammaDVol
    GBlackScholesNGreeks = (GBlackScholes(CallPutFlag, S + dS, X, T, r, b, v + 0.01) _
- 2 * GBlackScholes(CallPutFlag, S, X, T, r, b, v + 0.01) _
+ GBlackScholes(CallPutFlag, S - dS, X, T, r, b, v + 0.01) _
  - GBlackScholes(CallPutFlag, S + dS, X, T, r, b, v - 0.01) _
+ 2 * GBlackScholes(CallPutFlag, S, X, T, r, b, v - 0.01) _
- GBlackScholes(CallPutFlag, S - dS, X, T, r, b, v - 0.01)) _
 / (2 * 0.01 * dS^2) / 100
ElseIf OutPutFlag = "gp" Then 'GammaP
    GBlackScholesNGreeks = S / 100 * (GBlackScholes(CallPutFlag, _
    S + dS, X, T, r, b, v) - 2 * GBlackScholes(CallPutFlag, S, X, T, r, b, v)
_
+ GBlackScholes(CallPutFlag, S - dS, X, T, r, b, v)) / dS^2
ElseIf OutPutFlag = "dddv" Then 'DDeltaDvol
    GBlackScholesNGreeks = 1 / (4 * dS * 0.01) _
* (GBlackScholes(CallPutFlag, S + dS, X, T, r, b, v + 0.01) _
- GBlackScholes(CallPutFlag, S + dS, X, T, r, b, v - 0.01) _
     - GBlackScholes(CallPutFlag, S - dS, X, T, r, b, v + 0.01) _
 + GBlackScholes(CallPutFlag, S - dS, X, T, r, b, v - 0.01)) / 100
ElseIf OutPutFlag = "v" Then 'Vega
     GBlackScholesNGreeks = (GBlackScholes(CallPutFlag, S, X, T, r, b, v + 0.01) _
 - GBlackScholes(CallPutFlag, S, X, T, r, b, v - 0.01)) / 2
ElseIf OutPutFlag = "vv" Then 'DvegaDvol/vomma
    GBlackScholesNGreeks = (GBlackScholes(CallPutFlag, S, X, T, r, b, v + 0.01) _
 - 2 * GBlackScholes(CallPutFlag, S, X, T, r, b, v) _
 + GBlackScholes(CallPutFlag, S, X, T, r, b, v - 0.01)) / 0.01^2 / 10000
ElseIf OutPutFlag = "vp" Then 'VegaP
     GBlackScholesNGreeks = v / 0.1 _
 * (GBlackScholes(CallPutFlag, S, X, T, r, b, v + 0.01) _
 - GBlackScholes(CallPutFlag, S, X, T, r, b, v - 0.01)) / 2
 ElseIf OutPutFlag = "dvdv" Then 'DvegaDvol
    GBlackScholesNGreeks = (GBlackScholes(CallPutFlag, S, X, T, r, b, v + 0.01) _
 - 2 * GBlackScholes(CallPutFlag, S, X, T, r, b, v) _
 + GBlackScholes(CallPutFlag, S, X, T, r, b, v - 0.01))
ElseIf OutPutFlag = "t" Then 'Theta
     If T <= 1 / 365 Then
      GBlackScholesNGreeks = GBlackScholes(CallPutFlag, S, X, 1e-05, r, b, v) _
 - GBlackScholes(CallPutFlag, S, X, T, r, b, v)
    Else
       GBlackScholesNGreeks = GBlackScholes(CallPutFlag, S, X, T - 1 / 365, r, b, v) _
 - GBlackScholes(CallPutFlag, S, X, T, r, b, v)
    End If
  ElseIf OutPutFlag = "r" Then 'Rho
     GBlackScholesNGreeks = (GBlackScholes(CallPutFlag, S, X, T, r + 0.01, b + 0.01, v) _
 - GBlackScholes(CallPutFlag, S, X, T, r - 0.01, b - 0.01, v)) / (2)
 ElseIf OutPutFlag = "fr" Then 'Futures options rho
     GBlackScholesNGreeks = (GBlackScholes(CallPutFlag, S, X, T, r + 0.01, b, v) _
 - GBlackScholes(CallPutFlag, S, X, T, r - 0.01, b, v)) / (2)
 ElseIf OutPutFlag = "f" Then 'Rho2
     GBlackScholesNGreeks = (GBlackScholes(CallPutFlag, S, X, T, r, b - 0.01, v) _
 - GBlackScholes(CallPutFlag, S, X, T, r, b + 0.01, v)) / (2)
 ElseIf OutPutFlag = "b" Then 'Carry
    GBlackScholesNGreeks = (GBlackScholes(CallPutFlag, S, X, T, r, b + 0.01, v) _
 - GBlackScholes(CallPutFlag, S, X, T, r, b - 0.01, v)) / (2)
ElseIf OutPutFlag = "s" Then 'Speed
    GBlackScholesNGreeks = 1 / dS^3 _
 * (GBlackScholes(CallPutFlag, S + 2 * dS, X, T, r, b, v) _
```

TABLE 2-3

($S = 98, X = 100, T = 0.25, r = 0.1, b = 0.05, \sigma = 0.3$)

	Analytical	Numerical
Delta Δ	0.503105	0.503105
Elasticity	9.059951	9.059951
Gamma Γ	0.026794	0.026794
DGammaDvol	−0.000896	−0.000897
GammaP Γ_P	0.026258	0.026258
Vega	0.192999	0.192999
DvegaDvol	−0.000019	−0.000019
VegaP	0.578998	0.578997
Theta Θ (1 day)	−0.036989	−0.037080
Rho ρ	0.109656	0.109655
Rho futures option	−0.012124	−0.012124
Phi/Rho2	−0.123261	−0.123262
Carry Rho	0.123261	0.123262
DDeltaDvol	0.001659	0.001660
Strike delta	−0.438623	−0.438623
Speed	−0.000317	−0.000317
Risk Neutral Density	0.025733	0.025733

```
    - 3 * GBlackScholes(CallPutFlag, S + dS, X, T, r, b, v) _
        + 3 * GBlackScholes(CallPutFlag, S, X, T, r, b, v) _
    - GBlackScholes(CallPutFlag, S - dS, X, T, r, b, v))
    ElseIf OutPutFlag = "dx" Then 'Strike Delta
      GBlackScholesNGreeks = (GBlackScholes(CallPutFlag, S, X + dS, T, r, b, v) _
    - GBlackScholes(CallPutFlag, S, X - dS, T, r, b, v)) / (2 * dS)
    ElseIf OutPutFlag = "dxdx" Then 'Gamma
      GBlackScholesNGreeks = (GBlackScholes(CallPutFlag, S, X + dS, T, r, b, v) _
    - 2 * GBlackScholes(CallPutFlag, S, X, T, r, b, v) _
    + GBlackScholes(CallPutFlag, S, X - dS, T, r, b, v)) / dS^2
  End If

End Function
```

Table 2-3 shows numerical and analytical Greeks from the BSM formula.

2.12 GREEKS FROM CLOSED-FORM APPROXIMATIONS

Even if a closed-form approximation may be sufficiently accurate to calculate an option price, the accuracy will generally decrease as we move on to the Greeks. The closed-form approximation can thus be good enough to calculate the price, but the gamma can be highly inaccurate, and the third derivative like speed even more so.

2.13 APPENDIX B: TAKING PARTIAL DERIVATIVES

This appendix shows in detail how to find some of the most common partial derivatives of the BSM formula. In today's age of computers, one will typically solve such partial derivatives using mathematical software, like Mathematica or Maple. We first repeat the BSM formula for easy reference:

$$c = Se^{(b-r)T}N(d_1) - Xe^{-rT}N(d_2)$$

$$p = Xe^{-rT}N(-d_2) - Se^{(b-r)T}N(-d_1)$$

$$d_1 = \frac{\ln(S/X) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = \frac{\ln(S/X) + (b - \sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T}$$

$b = r$	Gives the Black and Scholes (1973) stock option model.
$b = r - q$	Gives the Merton (1973) stock option model with continuous dividend yield q.
$b = 0$	Gives the Black (1976) futures option model.
$b = 0$ and $r = 0$	Gives the Asay (1982) margined futures option model.
$b = r - r_f$	Gives the Garman and Kohlhagen (1983) currency option model.

The following relationships are also useful:

$$d_2 = d_1 - \sigma\sqrt{T}$$

$$d_2^2 = d_1^2 - 2d_1\sigma\sqrt{T} + \sigma^2 T$$

$$= d_1^2 - 2[\ln(S/X) + (b + \sigma^2/2)T] + \sigma^2 T$$

$$= d_1^2 - 2\ln(Se^{bT}/X)$$

$$n(d_2) = \frac{1}{\sqrt{2\pi}}e^{-d_2^2/2}$$

$$n(d_2) = \frac{1}{\sqrt{2\pi}}e^{-d_1^2/2 + \ln(Se^{bT}/X)}$$

$$= \frac{1}{\sqrt{2\pi}}e^{-d_1^2/2}e^{\ln(Se^{bT}/X)}$$

$$= n(d_1)Se^{bT}/X$$

$$n(d_1) = \frac{1}{\sqrt{2\pi}} e^{-d_1^2/2}$$

$$n(d_1) = n(d_2)X/Se^{bT}$$

Partial Derivatives

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp(-z^2/2)dz$$

$$\frac{\partial N(x)}{\partial S} = n(x)\frac{\partial x}{\partial S}$$

Delta

$$\begin{aligned}
\Delta_{\text{call}} = \frac{\partial c}{\partial S} &= e^{(b-r)T} N(d_1) + Se^{(b-r)T} \frac{\partial N(d_1)}{\partial S} - Xe^{-rT} \frac{\partial N(d_2)}{\partial S} \\
&= e^{(b-r)T} N(d_1) + Se^{(b-r)T} \frac{\partial N(d_1)}{\partial d_1}\frac{\partial d_1}{\partial S} - Xe^{-rT} \frac{\partial N(d_2)}{\partial d_2}\frac{\partial d_2}{\partial S} \\
&= e^{(b-r)T} N(d_1) + Se^{(b-r)T} n(d_1)\frac{\partial d_1}{\partial S} - Xe^{-rT} n(d_2)\frac{\partial d_2}{\partial S} \\
&= e^{(b-r)T} N(d_1) + Se^{(b-r)T} n(d_1)\frac{\partial d_1}{\partial S} - Xe^{-rT} n(d_1)Se^{bT}/X\frac{\partial d_1}{\partial S} \\
&= e^{(b-r)T} N(d_1) > 0
\end{aligned}$$

$$\begin{aligned}
\Delta_{\text{put}} = \frac{\partial p}{\partial S} &= Xe^{-rT} \frac{\partial N(-d_2)}{\partial S} - e^{(b-r)T} N(-d_1) - Se^{(b-r)T} \frac{\partial N(-d_1)}{\partial S} \\
&= Xe^{-rT} n(-d_2)\frac{-\partial d_2}{\partial S} - e^{(b-r)T} N(-d_1) - Se^{(b-r)T} n(-d_1)\frac{-\partial d_1}{\partial S} \\
&= Xe^{-rT} n(-d_1)Se^{bT}/X\frac{\partial d_1}{\partial S} - e^{(b-r)T} N(-d_1) + Se^{(b-r)T} n(-d_1)\frac{\partial d_1}{\partial S} \\
&= -e^{(b-r)T} N(-d_1) < 0
\end{aligned}$$

Gamma

$$\begin{aligned}
\Gamma_{\text{call}} = \frac{\partial^2 c}{\partial S^2} = \frac{\partial \Delta_{\text{call}}}{\partial S} &= \frac{\partial e^{(b-r)T} N(d_1)}{\partial S} \\
&= \frac{n(d_1)e^{(b-r)T}}{S\sigma\sqrt{T}} > 0
\end{aligned}$$

$$\Gamma_{\text{put}} = \frac{\partial^2 p}{\partial S^2} = \frac{\partial \Delta_{\text{put}}}{\partial S} = \frac{-\partial e^{(b-r)T} N(-d_1)}{\partial S}$$

$$= \frac{n(d_1) e^{(b-r)T}}{S\sigma\sqrt{T}} > 0$$

Strike

$$\frac{\partial c}{\partial X} = Se^{(b-r)T} \frac{\partial N(d_1)}{\partial X} - e^{-rT} N(d_2) - Xe^{-rT} \frac{\partial N(d_2)}{\partial X}$$

$$= Se^{(b-r)T} \frac{\partial N(d_1)}{\partial d_1} \frac{\partial d_1}{\partial X} - e^{-rT} N(d_2) - Xe^{-rT} \frac{\partial N(d_2)}{\partial d_2} \frac{\partial d_2}{\partial X}$$

$$= Se^{(b-r)T} n(d_1) \frac{\partial d_1}{\partial X} - e^{-rT} N(d_2) - Xe^{-rT} n(d_2) \frac{\partial d_2}{\partial X}$$

$$= Se^{(b-r)T} n(d_1) \frac{\partial d_1}{\partial X} - e^{-rT} N(d_2) - Xe^{-rT} n(d_1) Se^{bT}/X \frac{\partial d_1}{\partial X}$$

$$= -e^{-rT} N(d_2) < 0$$

$$\frac{\partial p}{\partial X} = e^{-rT} N(-d_2) + Xe^{-rT} \frac{\partial N(-d_2)}{\partial X} - Se^{(b-r)T} \frac{\partial N(-d_1)}{\partial X}$$

$$= e^{-rT} N(-d_2) + Xe^{-rT} n(-d_2) \frac{\partial d_2}{\partial X} - Se^{(b-r)T} n(-d_1) \frac{\partial d_1}{\partial X}$$

$$= e^{-rT} N(-d_2) + Xe^{-rT} n(-d_1) Se^{bT}/X \frac{\partial d_1}{\partial X} - Se^{(b-r)T} n(-d_1) \frac{\partial d_1}{\partial X}$$

$$= e^{-rT} N(-d_2) > 0$$

$$\frac{\partial^2 c}{\partial X^2} = \frac{-\partial e^{-rT} N(d_2)}{\partial X}$$

$$= \frac{n(d_2) e^{-rT}}{X\sigma\sqrt{T}} > 0$$

$$\frac{\partial^2 p}{\partial X^2} = \frac{\partial e^{-rT} N(-d_2)}{\partial X}$$

$$= \frac{n(d_2) e^{-rT}}{X\sigma\sqrt{T}} > 0$$

Rho

$$\rho_{\text{call}} = \frac{\partial c}{\partial r} = TSe^{(b-r)T}N(d_1) - TSe^{(b-r)T}N(d_1) + Se^{(b-r)T}\frac{\partial N(d_1)}{\partial r}$$
$$+ TXe^{-rT}N(d_2) - Xe^{-rT}\frac{\partial N(d_2)}{\partial r}$$
$$= Se^{(b-r)T}n(d_1)\frac{\partial d_1}{\partial r} + TXe^{-rT}N(d_2)$$
$$- Xe^{-rT}n(d_1)Se^{bT}/X\frac{\partial d_1}{\partial r}$$
$$= TXe^{-rT}N(d_2) > 0$$

$$\rho_{\text{put}} = \frac{\partial p}{\partial r} = -TXe^{-rT}N(-d_2) + Xe^{-rT}\frac{\partial N(-d_2)}{\partial r}$$
$$- TSe^{(b-r)T}N(-d_1) + TSe^{(b-r)T}N(-d_1) - Se^{(b-r)T}\frac{\partial N(-d_1)}{\partial r}$$
$$= -TXe^{-rT}N(-d_2) < 0$$

For options on futures, where the cost-of-carry is zero, $b = 0$, we have

$$\rho_{\text{call}} = \frac{\partial c}{\partial r} = -TSe^{(b-r)T}N(d_1) + Se^{(b-r)T}\frac{\partial N(d_1)}{\partial r}$$
$$+ TXe^{-rT}N(d_2) - Xe^{-rT}\frac{\partial N(d_2)}{\partial r}$$
$$= -TSe^{(b-r)T}N(d_1) + Se^{(b-r)T}n(d_1)\frac{\partial d_1}{\partial r}$$
$$+ TXe^{-rT}N(d_2) - Xe^{-rT}n(d_1)Se^{bT}/X\frac{\partial d_1}{\partial r}$$
$$= -TSe^{(b-r)T}N(d_1) + TXe^{-rT}N(d_2)$$
$$= -Tc < 0$$

$$\rho_{\text{put}} = \frac{\partial p}{\partial r} = -TXe^{-rT}N(-d_2) + Xe^{-rT}\frac{\partial N(-d_2)}{\partial r}$$
$$+ TSe^{(b-r)T}N(-d_1) - Se^{(b-r)T}\frac{\partial N(-d_1)}{\partial r}$$
$$= -Tp < 0$$

Cost-of-Carry

$$\frac{\partial c}{\partial b} = TSe^{(b-r)T}N(d_1) + Se^{(b-r)T}\frac{\partial N(d_1)}{\partial b} - Xe^{-rT}\frac{\partial N(d_2)}{\partial b}$$

$$= TSe^{(b-r)T}N(d_1) + Se^{(b-r)T}n(d_1)\frac{\partial d_1}{\partial b} - Xe^{-rT}n(d_1)Se^{bT}/X\frac{\partial d_1}{\partial b}$$

$$= TSe^{(b-r)T}N(d_1) > 0$$

$$\frac{\partial p}{\partial b} = Xe^{-rT}\frac{\partial N(-d_2)}{\partial b} - TSe^{(b-r)T}N(-d_1) - Se^{(b-r)T}\frac{\partial N(-d_1)}{\partial b}$$

$$= -TSe^{(b-r)T}N(-d_1) > 0$$

Vega

$$\text{Vega}_{\text{call}} = \frac{\partial c}{\partial \sigma} = Se^{(b-r)T}\frac{\partial N(d_1)}{\partial \sigma} - Xe^{-rT}\frac{\partial N(d_2)}{\partial \sigma}$$

$$= Se^{(b-r)T}n(d_1)\frac{\partial d_1}{\partial \sigma} - Xe^{-rT}n(d_2)\frac{\partial d_2}{\partial \sigma}$$

$$= Se^{(b-r)T}n(d_1)\frac{\partial d_1}{\partial \sigma} - Xe^{-rT}n(d_1)Se^{bT}/X\frac{\partial d_2}{\partial \sigma}$$

$$= Se^{(b-r)T}n(d_1)\left[\frac{\partial d_1}{\partial \sigma} - \frac{\partial d_2}{\partial \sigma}\right]$$

$$= Se^{(b-r)T}n(d_1)\sqrt{T} > 0$$

$$\text{Vega}_{\text{put}} = \frac{\partial p}{\partial \sigma} = Xe^{-rT}\frac{\partial N(-d_2)}{\partial \sigma} - Se^{(b-r)T}\frac{\partial N(-d_1)}{\partial \sigma}$$

$$= Xe^{-rT}n(-d_2)\left[-\frac{\partial d_2}{\partial \sigma}\right] - Se^{(b-r)T}n(-d_1)\left[-\frac{\partial d_1}{\partial \sigma}\right]$$

$$= Se^{(b-r)T}n(d_1)\left[\frac{\partial d_1}{\partial \sigma} - \frac{\partial d_2}{\partial \sigma}\right]$$

$$= Se^{(b-r)T}n(d_1)\sqrt{T} > 0$$

$$\frac{\partial d_1}{\partial \sigma} - \frac{\partial d_2}{\partial \sigma} = \left[-\frac{\ln(Se^{bT}/X)}{\sigma^2\sqrt{T}} + \frac{1}{2}\sqrt{T}\right] - \left[-\frac{\ln(Se^{bT}/X)}{\sigma^2\sqrt{T}} - \frac{1}{2}\sqrt{T}\right] = \sqrt{T}$$

Theta

$$
\begin{aligned}
\Theta_{\text{call}} = -\frac{\partial c}{\partial T} &= -(b-r)Se^{(b-r)T}N(d_1) - Se^{(b-r)T}\frac{\partial N(d_1)}{\partial T} \\
&\quad - rXe^{-rT}N(d_2) + Xe^{-rT}\frac{\partial N(d_2)}{\partial T} \\
&= -(b-r)Se^{(b-r)T}N(d_1) - Se^{(b-r)T}n(d_1)\frac{\partial d_1}{\partial T} \\
&\quad - rXe^{-rT}N(d_2) + Xe^{-rT}n(d_1)Se^{bT}/X\frac{\partial d_2}{\partial T} \\
&= Se^{(b-r)T}n(d_1)\left[\frac{\partial d_2}{\partial T} - \frac{\partial d_1}{\partial T}\right] \\
&\quad - (b-r)Se^{(b-r)T}N(d_1) - rXe^{-rT}N(d_2) \\
&= -\frac{Se^{(b-r)T}n(d_1)\sigma}{2\sqrt{T}} - (b-r)Se^{(b-r)T}N(d_1) \\
&\quad - rXe^{-rT}N(d_2) \lessgtr 0
\end{aligned}
$$

$$
\begin{aligned}
\Theta_{\text{put}} = -\frac{\partial p}{\partial T} &= rXe^{-rT}N(-d_2) - Xe^{-rT}\frac{\partial N(-d_2)}{\partial T} \\
&\quad + (b-r)Se^{(b-r)T}N(-d_1) + Se^{(b-r)T}\frac{\partial N(-d_1)}{\partial T} \\
&= rXe^{-rT}N(-d_2) + (b-r)Se^{(b-r)T}N(-d_1) \\
&\quad \times Se^{(b-r)T}n(d_1)\left[\frac{\partial d_2}{\partial T} - \frac{\partial d_1}{\partial T}\right] \\
&= -\frac{Se^{(b-r)T}n(d_1)\sigma}{2\sqrt{T}} + (b-r)Se^{(b-r)T}N(-d_1) \\
&\quad + rXe^{-rT}N(-d_2) \lessgtr 0
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial d_2}{\partial T} - \frac{\partial d_1}{\partial T} &= \left[-\frac{\ln(S/X)}{2\sigma T^{3/2}} + \frac{b}{2\sigma\sqrt{T}} - \frac{\sigma}{4\sqrt{T}}\right] \\
&\quad - \left[-\frac{\ln(S/X)}{2\sigma T^{3/2}} + \frac{b}{2\sigma\sqrt{T}} + \frac{\sigma}{4\sqrt{T}}\right] = -\frac{\sigma}{2\sqrt{T}}
\end{aligned}
$$

CHAPTER 3

ANALYTICAL FORMULAS FOR AMERICAN OPTIONS

Nowadays people know the price of everything, and the value of nothing.

Oscar Wilde

An American option can be exercised at any time up to its expiration date. This added freedom complicates the valuation of American options relative to their European counterparts. With a few exceptions, it is not possible to find an exact formula for the value of American options. Several researchers have, however, come up with excellent closed-form approximations. These approximations have become especially popular because they execute quickly on computers compared to numerical techniques. At the end of the chapter, we look at closed-form solutions for perpetual American options.

3.1 THE BARONE-ADESI AND WHALEY APPROXIMATION

The quadratic approximation method by Barone-Adesi and Whaley (1987) can be used to price American call and put options on an underlying asset with cost-of-carry rate b. When $b \geq r$, the American call value is equal to the European call value and can then be found by using the generalized Black-Scholes-Merton (BSM) formula. The model is fast and accurate for most practical input values.

American Call

$$C(S, X, T) = \begin{cases} c_{BSM}(S, X, T) + A_2(S/S^*)^{q_2} & \text{when } S < S^* \\ S - X & \text{when } S \geq S^* \end{cases},$$

where $c_{BSM}(S, X, T)$ is the general Black-Scholes-Merton call formula, and

$$A_2 = \frac{S^*}{q_2}\{1 - e^{(b-r)T} N[d_1(S^*)]\}$$

$$d_1(S) = \frac{\ln(S/X) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$q_2 = \frac{-(N-1) + \sqrt{(N-1)^2 + 4M/K}}{2}$$

$$M = 2r/\sigma^2, \qquad N = 2b/\sigma^2, \qquad K = 1 - e^{-rT}$$

American Put

$$P(S, X, T) = \begin{cases} p_{BSM}(S, X, T) + A_1(S/S^{**})^{q_1} & \text{when } S > S^{**} \\ X - S & \text{when } S \le S^{**} \end{cases},$$

where $p_{BSM}(S, X, T)$ is the generalized BSM put option formula, and

$$A_1 = -\frac{S^{**}}{q_1}\{1 - e^{(b-r)T} N[-d_1(S^{**})]\}$$

$$q_1 = \frac{-(N-1) - \sqrt{(N-1)^2 + 4M/K}}{2},$$

where S^* is the critical commodity price for the call option that satisfies

$$S^* - X = c(S^*, X, T) + \{1 - e^{(b-r)T} N[d_1(S^*)]\} S^* \frac{1}{q_2}$$

$$LHS(S_i) = S_i - X$$

$$RHS(S_i) = c(S_i, X, T) + \{1 - e^{(b-r)T} N[d_1(S_i)]\} S_i \frac{1}{q_2}$$

This equation can be solved by using a Newton–Raphson algorithm. The slope of RHS at S_i is

$$\frac{\partial RHS}{\partial S_i} = b_i = e^{(b-r)T} N[d_1(S_i)](1 - 1/q_2) + \left\{1 - \frac{e^{(b-r)T} n[d_1(S_i)]}{\sigma\sqrt{T}}\right\} \frac{1}{q_2}$$

Given an initial value S_i, it follows directly from the Newton–Raphson method that the next and better estimate, S_{i+1}, is

$$S_{i+1} = \frac{[X - RHS(S_i) - b_i S_i]}{(1 - b_i)}$$

The iterative procedure should continue until the relative absolute error falls within an acceptable tolerance level. For instance:

$$|LHS(S_i) - RHS(S_i)|/X < 0.00001,$$

and S^{**} is the critical commodity price for the put option that satisfies

$$X - S^{**} = p(S^{**}, X, T) - \{1 - e^{(b-r)T} N[-d_1(S^{**})]\} S^{**} \frac{1}{q_1}$$

$$VS(S_j) = X - S_j$$

$$HS(S_j) = p(S_j, X, T) - \{1 - e^{(b-r)T} N[-d_1(S_j)]\} S_j \frac{1}{q_1}$$

$$\frac{\partial HS}{\partial S_j} = b_j = -e^{(b-r)T} N[-d_1(S_j)](1 - 1/q_1) - \left\{1 + \frac{e^{(b-r)T} n[-d_1(S_j)]}{\sigma\sqrt{T}}\right\} \frac{1}{q_1}$$

$$S_{j+1} = \frac{[X - HS(S_j) + b_j S_j]}{(1 + b_j)}$$

As always with the use of the Newton–Raphson method, we need a seed value. Barone-Adesi and Whaley suggest using

$$S_1^* = X + [S^*(\infty) - X][1 - e^{h_2}] \qquad h_2 = -(bT + 2\sigma\sqrt{T}) \left[\frac{X}{S^*(\infty) - X}\right]$$

$$S_1^{**} = S^{**}(\infty) + [X - S^{**}(\infty)]e^{h_1} \qquad h_1 = (bT - 2\sigma\sqrt{T}) \left[\frac{X}{X - S^{**}(\infty)}\right],$$

where $S(\infty)$ is the critical price when time to expiration is infinite:

$$S^*(\infty) = \frac{X}{1 - 2[-(N-1) + \sqrt{(N-1)^2 + 4M}]^{-1}}$$

$$S^{**}(\infty) = \frac{X}{1 - 2[-(N-1) - \sqrt{(N-1)^2 + 4M}]^{-1}}$$

Table 3-1 compares option values given by the Barone-Adesi and Whaley Approximation (American style) and by the Black-76 formula (European style).

Computer algorithm

The *BAWAmericanCallApprox*$(\cdot)$ function can be used to calculate the value of an American call option using the Barone-Adesi and Whaley formula.

```
Function BAWAmericanCallApprox(S As Double, X As Double, T As Double, _
r As Double, b As Double, v As Double) As Double

    Dim Sk As Double, N As Double, K As Double
    Dim d1 As Double, Q2 As Double, a2 As Double

    If b >= r Then
        BAWAmericanCallApprox = GBlackScholes("c", S, X, T, r, b, v)
    Else
```

TABLE 3-1

Comparison of the Barone-Adesi and Whaley American Approximation and the Black–Scholes/Black-76 European Model

($X = 100, r = 0.1, b = 0$)

Futures price:		90	100	110	90	100	110
Call Options		BAW			Black-76		
$T = 0.1$	$\sigma = 0.15$	0.0206	1.8771	10.0089	0.0205	1.8734	9.9413
	$\sigma = 0.25$	0.3159	3.1280	10.3919	0.3150	3.1217	10.3556
	$\sigma = 0.35$	0.9495	4.3777	11.1679	0.9474	4.3693	11.1381
$T = 0.5$	$\sigma = 0.15$	0.8208	4.0842	10.8087	0.8069	4.0232	10.5769
	$\sigma = 0.25$	2.7437	6.8015	13.0170	2.7026	6.6997	12.7857
	$\sigma = 0.35$	5.0063	9.5106	15.5689	4.9329	9.3679	15.3080
Put Options		BAW			Black-76		
$T = 0.1$	$\sigma = 0.15$	10.0000	1.8770	0.0410	9.9210	1.8734	0.0408
	$\sigma = 0.25$	10.2533	3.1277	0.4562	10.2155	3.1217	0.4551
	$\sigma = 0.35$	10.8787	4.3777	1.2402	10.8479	4.3693	1.2376
$T = 0.5$	$\sigma = 0.15$	10.5595	4.0842	1.0822	10.3192	4.0232	1.0646
	$\sigma = 0.25$	12.4419	6.8014	3.3226	12.2149	6.6997	3.2734
	$\sigma = 0.35$	14.6945	9.5104	5.8823	14.4452	9.3679	5.7963

```
        Sk = Kc(X, T, r, b, v)
        N = 2 * b / v^2
        K = 2 * r / (v^2 * (1 - Exp(-r * T)))
        d1 = (Log(Sk / X) + (b + v^2 / 2) * T) / (v * Sqr(T))
        Q2 = (-(N - 1) + Sqr((N - 1)^2 + 4 * K)) / 2
        a2 = (Sk / Q2) * (1 - Exp((b - r) * T) * CND(d1))
        If S < Sk Then
            BAWAmericanCallApprox = GBlackScholes("c", S, X, T, r, b, v) _
            + a2 * (S / Sk)^Q2
        Else
            BAWAmericanCallApprox = S - X
        End If
    End If

End Function
```

The $Kc(\cdot)$ function below uses a Newton-Raphson algorithm to solve for the critical commodity price for a call option.

```
Function Kc(X As Double, T As Double, _
        r As Double, b As Double, v As Double) As Double

    Dim N As Double, m As Double
    Dim su As Double, Si As Double
    Dim h2 As Double, K As Double
    Dim d1 As Double, Q2 As Double, q2u As Double
```

```
    Dim LHS As Double, RHS As Double
    Dim bi As Double, E As Double

    '// Calculation of seed value, Si
    N = 2 * b / v^2
    m = 2 * r / v^2
    q2u = (-(N - 1) + Sqr((N - 1)^2 + 4 * m)) / 2
    su = X / (1 - 1 / q2u)
    h2 = -(b * T + 2 * v * Sqr(T)) * X / (su - X)
    Si = X + (su - X) * (1 - Exp(h2))

    K = 2 * r / (v^2 * (1 - Exp(-r * T)))
    d1 = (Log(Si / X) + (b + v^2 / 2) * T) / (v * Sqr(T))
    Q2 = (-(N - 1) + Sqr((N - 1)^2 + 4 * K)) / 2
    LHS = Si - X
    RHS = GBlackScholes("c", Si, X, T, r, b, v) _
          + (1 - Exp((b - r) * T) * CND(d1)) * Si / Q2
    bi = Exp((b - r) * T) * CND(d1) * (1 - 1 / Q2) _
          + (1 - Exp((b - r) * T) * CND(d1) / (v * Sqr(T))) / Q2
    E = 1e-06
    '// Newton Raphson algorithm for finding critical price Si
    While Abs(LHS - RHS) / X > E
        Si = (X + RHS - bi * Si) / (1 - bi)
        d1 = (Log(Si / X) + (b + v^2 / 2) * T) / (v * Sqr(T))
        LHS = Si - X
        RHS = GBlackScholes("c", Si, X, T, r, b, v) _
        + (1 - Exp((b - r) * T) * CND(d1)) * Si / Q2
        bi = Exp((b - r) * T) * CND(d1) * (1 - 1 / Q2) _
        + (1 - Exp((b - r) * T) * ND(d1) / (v * Sqr(T))) / Q2
    Wend
        Kc = Si
End Function
```

3.2 THE BJERKSUND AND STENSLAND (1993) APPROXIMATION

The Bjerksund and Stensland (1993b) approximation can be used to price American options on stocks, futures, and currencies. The method is analytical and extremely computer-efficient. Bjerksund and Stensland's approximation is based on an exercise strategy corresponding to a flat boundary I (trigger price). Numerical investigation indicates that the Bjerksund and Stensland model is somewhat more accurate for long-term options than the Barone-Adesi and Whaley model presented earlier; however, an even more precise approximation is the Bjerksund and Stendsland (2002) approximation soon to be presented.

$$
\begin{aligned}
C = {} & \alpha S^{\beta} - \alpha\phi(S, T, \beta, I, I) \\
& + \phi(S, T, 1, I, I) - \phi(S, T, 1, X, I) \\
& - X\phi(S, T, 0, I, I) + X\phi(S, T, 0, X, I),
\end{aligned}
\tag{3.1}
$$

where

$$\alpha = (I - X)I^{-\beta},$$

$$\beta = \left(\frac{1}{2} - \frac{b}{\sigma^2}\right) + \sqrt{\left(\frac{b}{\sigma^2} - \frac{1}{2}\right)^2 + 2\frac{r}{\sigma^2}}$$

The function $\phi(S, T, \gamma, H, I)$ is given by

$$\phi(S, T, \gamma, H, I) = e^{\lambda} S^{\gamma} \left[N(d) - \left(\frac{I}{S}\right)^{\kappa} N\left(d - \frac{2\ln(I/S)}{\sigma\sqrt{T}}\right)\right]$$

$$\lambda = \left[-r + \gamma b + \frac{1}{2}\gamma(\gamma - 1)\sigma^2\right] T$$

$$d = -\frac{\ln(S/H) + [b + (\gamma - \frac{1}{2})\sigma^2]T}{\sigma\sqrt{T}}$$

$$\kappa = \frac{2b}{\sigma^2} + (2\gamma - 1),$$

and the trigger price I is defined as

$$I = B_0 + (B_\infty - B_0)(1 - e^{h(T)})$$

$$h(T) = -(bT + 2\sigma\sqrt{T})\left(\frac{B_0}{B_\infty - B_0}\right)$$

$$B_\infty = \frac{\beta}{\beta - 1} X$$

$$B_0 = \max\left[X, \left(\frac{r}{r - b}\right) X\right]$$

If $S \geq I$, it is optimal to exercise the option immediately, and the value must be equal to the intrinsic value of $S - X$. On the other hand, if $b \geq r$, it will never be optimal to exercise the American call option before expiration, and the value can be found using the generalized BSM formula. The value of the American put is given by the Bjerksund and Stensland put-call transformation

$$P(S, X, T, r, b, \sigma) = C(X, S, T, r - b, -b, \sigma),$$

where $C(\cdot)$ is the value of the American call with risk-free rate $r - b$ and drift $-b$. With the use of this transformation, it is not necessary to develop a separate formula for an American put option.

Example

Consider an American-style call option with nine months to expiry. The stock price is 42, the strike price is 40, the risk-free rate is 4% per year, the dividend yield is 8% per year and the volatility is 35% per year. $S = 42$, $X = 40$, $T = 0.75$, $r = 0.04$, $b = 0.04 - 0.08 = -0.04$, and $\sigma = 0.35$.

$$\beta = \left(\frac{1}{2} - \frac{-0.04}{0.35^2}\right) + \sqrt{\left(\frac{-0.04}{0.35^2} - \frac{1}{2}\right)^2 + 2\frac{0.04}{0.35^2}} = 1.9825,$$

and the trigger price I is

$$B_\infty = \frac{\beta}{\beta - 1}40 = 80.7134$$

$$B_0 = \max\left[40, \left(\frac{0.04}{0.04 - (-0.04)}\right)40\right] = 40$$

$$h(T) = -(-0.04 \times 0.75 + 2 \times 0.35\sqrt{0.75})\left(\frac{B_0}{B_\infty - B_0}\right) = -0.5661$$

$$I = B_0 + (B_\infty - B_0)(1 - e^{h(T)}) = 57.5994$$

$$\alpha = (I - 40)I^{-\beta} = 0.005695,$$

and finally the American call value is

$$\begin{aligned} C &= \alpha 42^\beta - \alpha\phi(42, 0.75, \beta, I, I) \\ &+ \phi(42, 0.75, 1, I, I) - \phi(42, 0.75, 1, 40, I) \\ &- 40\phi(42, 0.75, 0, I, I) + 40\phi(42, 0.75, 0, 40, I) = 5.2704 \end{aligned}$$

The value of a similar European call is 5.0975.

Computer algorithm

The computer code for the Bjerksund and Stensland American option approximation consists of three functions. The first one checks if the option is a call or put. If the option is a put, the function uses the American put-call transformation. The function then calls the main function *BSAmericanCallApprox*(·), which calculates the option value. The main function uses two other functions: the *phi*(·) function, which in the formula above is described as $\phi(S, T, \gamma, H, I)$, and the *GBlackScholes*(·) function, which is the generalized BSM formula described in Chapter 1.

```
Function BSAmericanCallApprox(S As Double, X As Double, T As Double, _
r As Double, b As Double, v As Double) As Double

    Dim BInfinity As Double, B0 As Double
```

```
    Dim ht As Double, i As Double
    Dim Alpha As Double, Beta As Double

    If b >= r Then '// Never optimal to exersice before maturity
            BSAmericanCallApprox = GBlackScholes("c", S, X, T, r, b, v)
    Else
        Beta = (1 / 2 - b / v^2) + Sqr((b / v^2 - 1 / 2)^2 + 2 * r / v^2)
        BInfinity = Beta / (Beta - 1) * X
        B0 = Max(X, r / (r - b) * X)
        ht = -(b * T + 2 * v * Sqr(T)) * B0 / (BInfinity - B0)
        i = B0 + (BInfinity - B0) * (1 - Exp(ht))
        Alpha = (i - X) * i^(-Beta)
        If S >= i Then
            BSAmericanCallApprox = S - X
        Else
            BSAmericanCallApprox = Alpha * S ^ Beta _
       - Alpha * phi(S, T, Beta, i, i, r, b, v) _
      + phi(S, T, 1, i, i, r, b, v) - phi(S, T, 1, X, i, r, b, v) _
      - X * phi(S, T, 0, i, i, r, b, v) + X * phi(S, T, 0, X, i, r, b, v)
        End If
    End If

End Function


Function phi(S As Double, T As Double, gamma As Double, h As Double, i As Double
        r As Double, b As Double, v As Double) As Double

    Dim lambda As Double, kappa As Double
    Dim d As Double

    lambda = (-r + gamma * b + 0.5 * gamma * (gamma - 1) * v^2) * T
    d = -(Log(S/h) + (b + (gamma - 0.5) * v^2) * T) / (v * Sqr(T))
    kappa = 2 * b / (v^2) + (2 * gamma - 1)
    phi = Exp(lambda) * S^gamma * (CND(d) - (i / S)^kappa _
    * CND(d - 2 * Log(i / S) / (v * Sqr(T))))

End Function
```

where $CND(\cdot)$ is the cumulative normal distribution function described in Appendix A at the end of Chapter 1. Example: *BSAmericanApprox*("*c*", 42, 40, 0.75, 0.04, −0.04, 0.35) returns an American call value of 5.2704 as in the numerical example above.

3.3 THE BJERKSUND AND STENSLAND (2002) APPROXIMATION

The Bjerksund and Stensland (2002) approximation divides the time to maturity into two parts, each with a separate flat exercise boundary. It is thus a straightforward generalization of the Bjerksund-Stensland 1993 algorithm. The method is fast and efficient and should be more accurate than the Barone-Adesi and Whaley (1987) and the Bjerksund and Stensland (1993b) approximations. The algorithm requires an accurate cumulative bivariate normal approximation.

Several approximations that are described in the literature are not sufficiently accurate, but the Genze algorithm presented in Chapter 13 should do.

$$\begin{aligned} C = {} & \alpha_2 S^\beta - \alpha_2 \phi(S, t_1, \beta, I_2, I_2) \\ & + \phi(S, t_1, 1, I_2, I_2) - \phi(S, t_1, 1, I_1, I_2) \\ & - X\phi(S, t_1, 0, I_2, I_2) + X\phi(S, t_1, 0, I_1, I_2) \\ & + \alpha_1 \phi(S, t_1, \beta, I_1, I_2) - \alpha_1 \Psi(S, T, \beta, I_1, I_2, I_1, t_1) \\ & + \Psi(S, T, 1, I_1, I_2, I_1, t_1) - \Psi(S, T, 1, X, I_2, I_1, t_1) \\ & - X\Psi(S, T, 0, I_1, I_2, I_1, t_1) + \Psi(S, T, 0, X, I_2, I_1, t_1), \end{aligned} \tag{3.2}$$

where

$$\alpha_1 = (I_1 - X) I_1^{-\beta}, \quad \alpha_2 = (I_2 - X) I_2^{-\beta}$$

$$\beta = \left(\frac{1}{2} - \frac{b}{\sigma^2}\right) + \sqrt{\left(\frac{b}{\sigma^2} - \frac{1}{2}\right)^2 + 2\frac{r}{\sigma^2}}$$

The function $\phi(S, T, \gamma, H, I)$ is given by

$$\phi(S, T, \gamma, H, I) = e^\lambda S^\gamma \left[N(-d) - \left(\frac{I}{S}\right)^\kappa N(-d_2) \right]$$

$$d = \frac{\ln(S/H) + [b + (\gamma - \frac{1}{2})\sigma^2]T}{\sigma\sqrt{T}}$$

$$d_2 = \frac{\ln(I^2/(SH)) + [b + (\gamma - \frac{1}{2})\sigma^2]T}{\sigma\sqrt{T}}$$

$$\lambda = -r + \gamma b + \frac{1}{2}\gamma(\gamma - 1)\sigma^2$$

$$\kappa = \frac{2b}{\sigma^2} + (2\gamma - 1),$$

and the trigger price I is defined as

$$I_1 = B_0 + (B_\infty - B_0)(1 - e^{h_1})$$

$$I_2 = B_0 + (B_\infty - B_0)(1 - e^{h_2})$$

$$h_1 = -(bt_1 + 2\sigma\sqrt{t_1})\left(\frac{X^2}{(B_\infty - B_0)B_0}\right)$$

$$h_2 = -(bT + 2\sigma\sqrt{T})\left(\frac{X^2}{(B_\infty - B_0)B_0}\right)$$

$$t_1 = \frac{1}{2}(\sqrt{5} - 1)T$$

$$B_\infty = \frac{\beta}{\beta - 1}X$$

$$B_0 = \max\left[X, \left(\frac{r}{r-b}\right)X\right]$$

Moreover, the function $\Psi(S, T, \gamma, H, I_2, I_1, t_1)$ is given by

$$\Psi(S, T, \gamma, H, I_2, I_1, t_1, r, b, \sigma) = e^{\lambda T} S^\gamma \left[M(-e_1, -f_1, \rho) - (I_2/S)^\kappa M(-e_2, -f_2, \rho) \right.$$
$$\left. - (I_1/S)^\kappa M(-e_3, -f_3, -\rho) + (I_1/I_2)^\kappa M(-e_4, -f_4, -\rho)\right],$$

where

$$e_1 = \frac{\ln(S/I_1) + (b + (\gamma - \frac{1}{2})\sigma^2)t_1}{\sigma\sqrt{t_1}} \qquad e_2 = \frac{\ln(I_2^2/(SI_1)) + (b + (\gamma - \frac{1}{2})\sigma^2)t_1}{\sigma\sqrt{t_1}}$$

$$e_3 = \frac{\ln(S/I_1) - (b + (\gamma - \frac{1}{2})\sigma^2)t_1}{\sigma\sqrt{T}} \qquad e_4 = \frac{\ln(I_2^2/(SI_1)) - (b + (\gamma - \frac{1}{2})\sigma^2)t_1}{\sigma\sqrt{t_1}}$$

$$f_1 = \frac{\ln(S/H) + (b + (\gamma - \frac{1}{2})\sigma^2)T}{\sigma\sqrt{T}} \qquad f_2 = \frac{\ln(I_2^2/(SH)) + (b + (\gamma - \frac{1}{2})\sigma^2)T}{\sigma\sqrt{T}}$$

$$f_3 = \frac{\ln(I_2^2/(SH)) + (b + (\gamma - \frac{1}{2})\sigma^2)T}{\sigma\sqrt{T}} \qquad f_4 = \frac{\ln(SI_1^2/(HI_2^2)) + (b + (\gamma - \frac{1}{2})\sigma^2)T}{\sigma\sqrt{T}}$$

Table 3-2 gives numerical values for the Bjerksund-Stensland (2002) approximation and also their 1993 approximation. The 2002 version of their model is more accurate but slightly more computer-intensive.

Computer algorithm

The computer code for the Bjerksund and Stensland 2002 American option approximation consists of four functions. The first one checks if the option is a call or a put. If the option is a put, the function uses the American put-call transformation. The function then calls the main function *BSAmericanCallApprox*2002(·), which calculates the option value. The main function uses two other functions: the *phi*(·) function, which in the formula above is described as $\phi(S, T, \gamma, H, I)$ and the *ksi*(·) function described as $\Psi(S, T, \gamma, H, I_2, I_1, t_1, r, b, \sigma)$, and the *GBlackScholes*(·) function, which is the generalized BSM formula described in Chapter 1.

```
Function BSAmericanApprox2002(CallPutFlag As String, S As Double, X As Double, _
T As Double, r As Double, b As Double, v As Double) As Double

    If CallPutFlag = "c" Then
        BSAmericanApprox2002 = BSAmericanCallApprox2002(S, X, T, r, b, v)
```

TABLE 3-2

Comparison of Values from the Bjerksund and Stensland (2002) and (1993) Models

$(X = 100, r = 0.1, b = 0)$

Futures price:		90	100	110	90	100	110
Call Options		Bjerksund-Stensland-02			Bjerksund-Stensland-93		
$T = 0.1$	$\sigma = 0.15$	0.0205	1.8757	10.0000	0.0206	1.8769	10.0061
	$\sigma = 0.25$	0.3151	3.1256	10.3725	0.3159	3.1277	10.3901
	$\sigma = 0.35$	0.9479	4.3746	11.1578	0.9495	4.3777	11.1678
$T = 0.5$	$\sigma = 0.15$	0.8099	4.0628	10.7898	0.8208	4.0841	10.8085
	$\sigma = 0.25$	2.7180	6.7661	12.9814	2.7436	6.8013	13.0167
	$\sigma = 0.35$	4.9665	9.4608	15.5137	5.0062	9.5103	15.5684
Put Options		Bjerksund-Stensland-02			Bjerksund-Stensland-93		
$T = 0.1$	$\sigma = 0.15$	10.0000	1.8757	0.0408	10.0000	1.8769	0.0410
	$\sigma = 0.25$	10.2280	3.1256	0.4552	10.2530	3.1277	0.4562
	$\sigma = 0.35$	10.8663	4.3746	1.2383	10.8785	4.3777	1.2402
$T = 0.5$	$\sigma = 0.15$	10.5400	4.0628	1.0689	10.5592	4.0841	1.0822
	$\sigma = 0.25$	12.4097	6.7661	3.2932	12.4417	6.8014	3.3226
	$\sigma = 0.35$	14.6445	9.4608	5.8374	14.6943	9.5103	5.8822

```
    ElseIf CallPutFlag = "p" Then   '// Use put-call transformation
        BSAmericanApprox2002 = BSAmericanCallApprox2002(X, S, T, r - b, -b, v)
    End If

End Function

Function BSAmericanCallApprox2002(S As Double, X As Double, T As Double, _
r As Double, b As Double, v As Double) As Double

    Dim BInfinity As Double, B0 As Double
    Dim ht1 As Double, ht2 As Double, I1 As Double, I2 As Double
    Dim alfa1 As Double, alfa2 As Double, Beta As Double, t1 As Double

    t1 = 1 / 2 * (Sqr(5) - 1) * T

    If b >= r Then   '// Never optimal to exersice before maturity
        BSAmericanCallApprox2002 = GBlackScholes("c", S, X, T, r, b, v)
    Else

        Beta = (1/2-b/v^2) + Sqr((b/v^2-1/2)^2+2*r/v^2)
        BInfinity = Beta/(Beta-1)*X
        B0 = Max(X, r/(r-b)*X)

        ht1 = -(b*t1+2*v*Sqr(t1))*X^2/((BInfinity-B0) * B0)
        ht2 = -(b * T + 2 * v * Sqr(T)) * X^2 / ((BInfinity - B0) * B0)
        I1 = B0 + (BInfinity - B0) * (1 - Exp(ht1))
        I2 = B0 + (BInfinity - B0) * (1 - Exp(ht2))
        alfa1 = (I1 - X) * I1^(-Beta)
        alfa2 = (I2 - X) * I2^(-Beta)

        If S >= I2 Then
            BSAmericanCallApprox2002 = S - X
        Else
            BSAmericanCallApprox2002 = alfa2 * S^Beta - alfa2 * phi(S, t1, Beta, I2, I2, r, b, v) _
                + phi(S, t1, 1, I2, I2, r, b, v) - phi(S, t1, 1, I1, I2, r, b, v) _
                - X * phi(S, t1, 0, I2, I2, r, b, v) + X * phi(S, t1, 0, I1, I2, r, b, v) _
                + alfa1 * phi(S, t1, Beta, I1, I2, r, b, v) - alfa1 * ksi(S, T, Beta, I1, I2, I1, t1, r, b,
                + ksi(S, T, 1, I1, I2, I1, t1, r, b, v) - ksi(S, T, 1, X, I2, I1, t1, r, b, v) _
```

```
                - X * ksi(S, T, 0, I1, I2, I1, t1, r, b, v) + X * ksi(S, T, 0, X, I2, I1, t1, r, b, v)

        End If
    End If
End Function


Public Function BSAmericanCallApprox2002(S As Double, X As Double, T As Double, _
r As Double, b As Double, v As Double) As Double

    Dim BInfinity As Double, B0 As Double
    Dim ht1 As Double, ht2 As Double, I1 As Double, I2 As Double
    Dim alfa1 As Double, alfa2 As Double, Beta As Double, t1 As Double

    t1 = 1 / 2 * (Sqr(5) - 1) * T

    If b >= r Then   '// Never optimal to exersice before maturity
            BSAmericanCallApprox2002 = GBlackScholes("c", S, X, T, r, b, v)
    Else

        Beta = (1/2-b/v^2) + Sqr((b/v^2-1/2)^2+2*r/v^2)
        BInfinity = Beta/(Beta-1)*X
        B0 = Max(X, r/(r-b)*X)

        ht1 = -(b*t1+2*v*Sqr(t1))*X^2/((BInfinity-B0) * B0)
        ht2 = -(b * T + 2 * v * Sqr(T)) * X^2 / ((BInfinity - B0) * B0)
        I1 = B0 + (BInfinity - B0) * (1 - Exp(ht1))
        I2 = B0 + (BInfinity - B0) * (1 - Exp(ht2))
        alfa1 = (I1 - X) * I1^(-Beta)
        alfa2 = (I2 - X) * I2^(-Beta)

        If S >= I2 Then
            BSAmericanCallApprox2002 = S - X
        Else
            BSAmericanCallApprox2002 = alfa2 * S^Beta - alfa2 * phi(S, t1, Beta, I2, I2, r, b, v) _
                + phi(S, t1, 1, I2, I2, r, b, v) - phi(S, t1, 1, I1, I2, r, b, v) _
                - X * phi(S, t1, 0, I2, I2, r, b, v) + X * phi(S, t1, 0, I1, I2, r, b, v) _
                + alfa1 * phi(S, t1, Beta, I1, I2, r, b, v) - alfa1 * ksi(S, T, Beta, I1, I2, I1, t1, r, b, v) _
                + ksi(S, T, 1, I1, I2, I1, t1, r, b, v) - ksi(S, T, 1, X, I2, I1, t1, r, b, v) _
                - X * ksi(S, T, 0, I1, I2, I1, t1, r, b, v) + X * ksi(S, T, 0, X, I2, I1, t1, r, b, v)

        End If
    End If
End Function

Function ksi(S As Double, T2 As Double, gamma As Double, h As Double, I2 As Double, _
                I1 As Double, t1 As Double, r As Double, b As Double, v As Double) As Double

    Dim e1 As Double, e2 As Double, e3 As Double, e4 As Double
    Dim f1 As Double, f2 As Double, f3 As Double, f4 As Double
    Dim rho As Double, kappa As Double, lambda As Double

    e1 = (Log(S/I1)+(b+(gamma-0.5)*v^2)*t1)/(v*Sqr(t1))
    e2 = (Log(I2^2/(S*I1))+(b+(gamma-0.5)*v^2)*t1)/(v*Sqr(t1))
    e3 = (Log(S/I1)-(b+(gamma-0.5)*v^2)*t1)/(v*Sqr(t1))
    e4 = (Log(I2^2/(S*I1))-(b+(gamma-0.5)*v^2)*t1)/(v*Sqr(t1))

    f1 = (Log(S/h)+(b+(gamma-0.5)*v^2)*T2)/(v*Sqr(T2))
    f2 = (Log(I2^2/(S*h))+(b+(gamma-0.5)*v^2)*T2)/(v*Sqr(T2))
    f3 = (Log(I1^2/(S*h))+(b+(gamma-0.5)*v^2)*T2)/(v*Sqr(T2))
    f4 = (Log((S*I1^2)/(h*I2^2)) _
            + (b+(gamma-0.5)*v^2)*T2)/(v*Sqr(T2))

    rho = Sqr(t1/T2)
    lambda = -r+gamma*b+0.5*gamma*(gamma-1)*v^2
    kappa = 2s*b/(v^2)+(2*gamma-1)

    ksi = Exp(lambda * T2) * S^gamma * (CBND(-e1, -f1, rho) _
                - (I2 / S)^kappa * CBND(-e2, -f2, rho) _
                - (I1 / S)^kappa * CBND(-e3, -f3, -rho) _
                + (I1 / I2)^kappa * CBND(-e4, -f4, -rho))


End Function
```

```
Function phi(S As Double, T As Double, gamma As Double, h As Double, i As Double, _
        r As Double, b As Double, v As Double) As Double

    Dim lambda As Double, kappa As Double
    Dim d As Double

    lambda = (-r + gamma * b + 0.5 * gamma * (gamma - 1) * v^2) * T
    d = -(Log(S / h) + (b + (gamma - 0.5) * v^2) * T) / (v * Sqr(T))
    kappa = 2 * b / (v^2) + (2 * gamma - 1)
    phi = Exp(lambda) * S^gamma * (CND(d) - (i / S) ^ kappa _
    * CND(d - 2 * Log(i / S) / (v * Sqr(T))))

End Function
```

3.4 PUT-CALL TRANSFORMATION AMERICAN OPTIONS

The Bjerksund and Stensland (1993a) put-call transformation is very useful when calculating American option values. If you have a formula for an American call, the relationship below will give the value for the American put.

$$P(S, X, T, r, b, \sigma) = C(X, S, T, r - b, -b, \sigma)$$

3.5 AMERICAN PERPETUAL OPTIONS

While there in general is no closed-form solution for American options (except for nondividend-paying stock call options) it is possible to find a closed-form solution for options with an infinite time to expiration. The reason is that the time to expiration will always be the same: infinite. The time to maturity, therefore, does not depend on at what point in time we look at the valuation problem, which makes the valuation problem independent of time McKean (1965) and Merton (1973)[1] gives closed-form solutions for American perpetual options. For a call option we have

$$c = \frac{X}{y_1 - 1}\left(\frac{y_1 - 1}{y_1}\frac{S}{X}\right)^{y_1}, \tag{3.3}$$

where

$$y_1 = \frac{1}{2} - \frac{b}{\sigma^2} + \sqrt{\left(\frac{b}{\sigma^2} - \frac{1}{2}\right)^2 + \frac{2r}{\sigma^2}}$$

If $b >= r$, then there is never optimal to exercise a call option. In the case of an American perpetual put, we have

$$p = \frac{X}{1 - y_2}\left(\frac{y_2 - 1}{y_2}\frac{S}{X}\right)^{y_2}, \tag{3.4}$$

[1]See also Aase (2005) and Gerber and Shiu (1994).

where

$$y_2 = \frac{1}{2} - \frac{b}{\sigma^2} - \sqrt{\left(\frac{b}{\sigma^2} - \frac{1}{2}\right)^2 + \frac{2r}{\sigma^2}}$$

In practice, one can naturally discuss if there is such a thing as infinite time to maturity. For instance, credit risk could play an important role: Even when you are buying an option from an AAA bank, there is no guarantee the bank will be around forever.

Table 3-3 shows values of perpetual American call option values.

TABLE 3-3

Examples of American Perpetual Call Options Values

($X = 100, r = 0.1, b = 0.02$)

σ	$S = 70$	$S = 80$	$S = 90$	$S = 100$	$S = 110$	$S = 120$	$S = 130$
10%	4.3229	6.6426	9.7027	13.6174	18.5035	24.4804	31.6700
15%	6.8933	9.7777	13.3089	17.5358	22.5052	28.2622	34.8503
20%	9.6812	13.0497	16.9817	21.4931	26.5984	32.3112	38.6441
25%	12.5392	16.3295	20.6133	25.3893	30.6563	36.4134	42.6597
30%	15.3861	19.5513	24.1519	29.1774	34.6188	40.4679	46.7174

Example

Consider a perpetual call option with strike 100, risk-free rate 10%, on a stock index trading at 90 with continuous dividend yield 8% and volatility 25%. What is the value of the option? $S = 90$, $X = 100$, $r = 0.1$, $b = 0.1 - 0.08 = 0.02$, $\sigma = 0.25$, which yields

$$y_1 = \frac{1}{2} - \frac{0.02}{0.25^2} + \sqrt{\left(\frac{0.02}{0.25^2} - \frac{1}{2}\right)^2 + \frac{2 \times 0.1}{0.25^2}} = 1.9779$$

$$c = \frac{100}{1.9779 - 1}\left(\frac{1.9779 - 1}{1.9779}\frac{90}{100}\right)^{1.9779} = 20.6133$$

CHAPTER

4

EXOTIC OPTIONS—SINGLE ASSET

With derivatives you can have almost any payoff pattern you want. If you can draw it on paper, or describe it in words, someone can design a derivative that gives you that payoff.

Fischer Black

In this chapter I present a large class of analytical formulas for so-called exotic options on one underlying asset. The underlying asset is assumed to follow a geometric Brownian motion $dS = \mu S dt + \sigma S dz$, where μ is the expected instantaneous rate of return on the underlying asset, σ is the instantaneous volatility of the rate of return, and dz is a Wiener process. The volatility and risk-free rate is assumed to be constant throughout the life of the option. Most of the formulas are written on a general form, including a cost-of-carry term, which makes it possible to use the same formula to price options on a large class of underlying assets: stocks, stock indices paying a dividend yield, currencies, and futures. Since the formulas are closed-form solutions, they can, in general, only be used to price European-style options. A few cases admit closed-form solutions or approximations for the American counterpart. See Chapter 7 for more on pricing American-style exotic options.

4.1 VARIABLE PURCHASE OPTIONS

Handley (2001) describes how to value variable purchase options (VPO). A VPO is basically a call option, but where the number of underlying shares is stochastic rather than fixed, or more precisely, a deterministic function of the asset price. The strike price of a VPO is typically a fixed discount to the underlying share price at maturity. The payoff at maturity is equal to $\max[N \times S - X]$, where N is the number of shares. VPOs may be an interesting tool for firms that need to raise capital relatively far into the future at a given time. The number of underlying shares N is decided on at maturity and is equal to

$$N = \frac{X}{S_T(1-D)},$$

where X is the strike price, S_T is the asset price at maturity, and D is the fixed discount expressed as a proportion $0 \geq D \leq 1$. The number of shares is in this way a deterministic function of the asset price. Further, the number of shares is often subjected to a minimum and maximum. In this case, we will limit the minimum number of shares to $N_{min} = \frac{X}{U(1-D)}$ if, the asset price at maturity is above a predefined level U at maturity. Similarly, we will reach the maximum number of shares $N_{max} = \frac{X}{L(1-D)}$ if the stock price at maturity is equal or lower than a predefined level L. Based on Handley (2001), we get the following closed-form solution:

$$\begin{aligned} c = {} & \frac{XD}{1-D}e^{-rT} + N_{min}[Se^{(b-r)T}N(d_1) - Ue^{-rT}N(d_2)] \\ & - N_{max}[Le^{-rT}N(-d_4) - Se^{(b-r)T}N(-d_3)] \\ & + N_{max}[L(1-D)e^{-rT}N(-d_6) - Se^{(b-r)T}N(-d_5)], \end{aligned} \tag{4.1}$$

where

$$d_1 = \frac{\ln(S/U) + (b+\sigma^2/2)T}{\sigma\sqrt{T}} \qquad d_2 = d_1 - \sigma\sqrt{T}$$

$$d_3 = \frac{\ln(S/L) + (b+\sigma^2/2)T}{\sigma\sqrt{T}} \qquad d_4 = d_3 - \sigma\sqrt{T}$$

$$d_5 = \frac{\ln(S/(L(1-D))) + (b+\sigma^2/2)T}{\sigma\sqrt{T}} \qquad d_6 = d_5 - \sigma\sqrt{T}$$

Table 4-1 shows values for variable purchase options, for two different discount values D, different choices of volatility σ, as well as cap and floor levels: L and U.

Example

Consider a variable purchase option with six months to expiration and a strike price of 101. The current stock price is 100, the risk-free rate is 5%, the volatility is 20%, and the discount is 10%. Further, the number of shares is capped at stock price 90 and has a floor at stock price of 110. $S = 100$, $X = 101$, $T = 0.5$, $r = 0.05$, $b = 0.05$, $\sigma = 0.2$, $D = 0.1$, $L = 90$, and $U = 110$. What is the value of the VPO? First, the

TABLE 4-1

Variable Purchase Options Values

($S = 100, X = 101, T = 0.5, r = 0.05, b = 0.05$)

		$D = 0\%$			$D = 20\%$		
L	U	$\sigma = 10\%$	$\sigma = 20\%$	$\sigma = 30\%$	$\sigma = 10\%$	$\sigma = 20\%$	$\sigma = 30\%$
101	101	3.6093	6.3735	9.1518	26.4945	26.7913	28.0316
95	105	1.7415	4.4071	7.1166	26.1568	26.8766	27.7651
90	110	0.5661	2.6687	5.1300	25.2155	26.2052	26.9532
85	115	0.1447	1.5471	3.6483	24.7956	25.7473	26.4874
	BSM:	3.6093	6.3735	9.1518	3.6093	6.3735	9.1518

maximum number of shares we can get if the asset price is 90 or lower at expiration is

$$N_{Max} = \frac{X}{L(1-D)} = \frac{101}{90(1-0.1)} = 1.2469$$

Similarly, the minimum number of shares we get if the stock is 110 or higher at maturity is

$$N_{Min} = \frac{X}{U(1-D)} = \frac{101}{110(1-0.1)} = 1.0202$$

Further

$$d_1 = \frac{\ln(100/110) + (0.05 + 0.2^2/2)0.5}{0.2\sqrt{0.5}} = -0.4265$$

$$d_2 = -0.4265 - 0.2\sqrt{0.5} = -0.5679$$

$$d_3 = \frac{\ln(100/90) + (0.05 + 0.2^2/2)0.5}{0.2\sqrt{0.5}} = 0.9925$$

$$d_4 = 0.9925 - 0.2\sqrt{0.5} = 0.8511$$

$$d_5 = \frac{\ln(100/(90(1-0.1))) + (0.05 + 0.2^2/2)0.5}{0.2\sqrt{0.5}} = 1.7375$$

$$d_6 = 1.7375 - 0.2\sqrt{0.5} = 1.5961$$

$$N(d_1) = N(-0.4265) = 0.3349 \qquad N(d_2) = N(-0.5679) = 0.2851$$

$$N(-d_3) = N(-0.9925) = 0.1605 \qquad N(-d_4) = N(-0.8511) = 0.1974$$

$$N(-d_5) = N(-1.7375) = 0.0411 \qquad N(-d_6) = N(-1.5961) = 0.0552$$

This gives us a VPO value of

$$c = \frac{101 \times 0.1}{1 - 0.1} e^{-0.05\times0.5} + 1.0202[100e^{(0.05-0.05)0.5}N(d_1) - 110e^{-0.05\times0.5}N(d_2)]$$
$$- 1.2469[90e^{-0.05\times0.5}N(-d_4) - 100e^{(0.05-0.05)0.5}N(-d_3)]$$
$$+ 1.2469[90(1 - 0.1)e^{-0.05\times0.5}N(-d_6) - 100e^{(0.05-0.05)0.5}N(-d_5)]$$
$$= 12.6288$$

4.2 EXECUTIVE STOCK OPTIONS

The Jennergren and Naslund (1993) formula takes into account that an employee or executive often loses her options if she has to leave the company before the option's expiration:

$$c = e^{-\lambda T}[Se^{(b-r)T}N(d_1) - Xe^{-rT}N(d_2)] \tag{4.2}$$

$$p = e^{-\lambda T}[Xe^{-rT}N(-d_2) - Se^{(b-r)T}N(-d_1)], \tag{4.3}$$

where

$$d_1 = \frac{\ln(S/X) + (b + \sigma^2/2)T}{\sigma\sqrt{T}} \qquad d_2 = d_1 - \sigma\sqrt{T}$$

(λ) is the jump rate per year. The value of the executive option equals the ordinary Black-Scholes option price multiplied by the probability $e^{-\lambda T}$ that the executive will stay with the firm until the option expires.

Example

What is the value of an executive call option when the stock price is 60, the strike price is 64, the time to maturity is two years, the risk-free rate is 7%, the dividend yield is 3%, the stock volatility is 38%, and the jump rate per year is 15%? $S = 60$, $X = 64$, $T = 2$, $r = 0.07$, $b = 0.07 - 0.03 = 0.04$, $\sigma = 0.38$, $\lambda = 0.15$, and

$$d_1 = \frac{\ln(60/64) + (0.04 + 0.38^2/2)2}{0.38\sqrt{2}} = 0.2975$$

$$d_2 = d_1 - 0.38\sqrt{2} = -0.2399$$

$$N(d_1) = N(0.2975) = 0.6169, \quad N(d_2) = N(-0.2399) = 0.4052$$

$$c = e^{-0.15\times2}[60e^{(0.04-0.07)2}N(d_1) - 64e^{-0.07\times2}N(d_2)] = 9.1244$$

4.3 MONEYNESS OPTIONS

A moneyness option is basically a plain vanilla option where the strike is set to a percentage of the future/forward price. For example, a 120%

moneyness call would have a strike equal to 120% of the forward price. A 120% moneyness put would have a spot equal to 120% of the strike. The value of this option is given in percent of the forward. The value of a moneyness call or put is thus given by

$$c = p = e^{-rT}[N(d_1) - LN(d_2)], \tag{4.4}$$

where $L = X/F$ for a call and $L = F/X$ for a put, and

$$d_1 = \frac{-\ln(L) + \sigma^2 T/2}{\sigma\sqrt{T}}, \qquad d_2 = d_1 - \sigma\sqrt{T}$$

Example

What is the value of a call option that is 120% out-of-the-money, with nine months to maturity, risk-free rate of 8%, and volatility of 30%? $L = X/F = 1.2$, $T = 0.75$, $r = 0.08$, $\sigma = 0.3$, and thus

$$d_1 = \frac{-\ln(1.2) + 0.3^2 \times 0.75/2}{0.3\sqrt{0.75}} = -0.5719$$

$$d_2 = d_1 - 0.3\sqrt{0.75} = -0.8317$$

$$N(d_1) = N(-0.5719) = 0.2837$$

$$N(d_2) = N(-0.8317) = 0.2028$$

$$c = e^{-0.08\times 0.75}[N(d_1) - 1.2N(d_2)] = 0.0380$$

A call that is 120% out-of the money relative to the forward price will thus have a value equal to 3.8% of the forward price.

4.4 POWER CONTRACTS AND POWER OPTIONS

There are two main categories of power options. Standard power options' payoff depends on the price of the underlying asset raised to some power. For powered options, the "standard" payoff (stock price in excess of the exercise price) is raised to some power. While it is possible to come up with lots of variants within these two categories, we cover the most common types. Additional types of powered options are covered in Chapter 7.

4.4.1 Power Contracts

A power contract is a simple derivative instrument paying $(S/X)^i$ at maturity, where i is some fixed power. The value of such a power contract is given by Shaw (1998) as

$$V_{\text{Power}} = \left(\frac{S}{X}\right)^i e^{[(b-\sigma^2/2)i - r + i^2\sigma^2/2]T} \tag{4.5}$$

Example

What is the current value of a power contract with a power of 2, six months to expiration, on a stock index trading at 400, with strike price of 450, risk-free rate of 8%, dividend yield of 2%, and volatility of 25%? $S = 400$, $X = 450$, $T = 0.5$, $r = 0.08$, $b = 0.08 - 0.02 = 0.06$, $\sigma = 0.25$, $i = 2$ yields

$$V_{\text{Power}} = \left(\frac{400}{450}\right)^2 e^{[(0.06-0.25^2/2)2-0.08+2^2\times0.25^2/2]0.5} = 0.8317$$

4.4.2 Standard Power Option

Standard power options (aka asymmetric power options) have nonlinear payoff at maturity. For a call, the payoff is $\max[S^i - X, 0]$, and for a put, it is $\max[X - S^i, 0]$, where i is some power ($i > 0$). The value of this power call is given by (see Heynen and Kat, 1996c; Zhang, 1998; and Esser, 2003)

$$c = S^i e^{[(i-1)(r+i\sigma^2/2)-i(r-b)]T} N(d_1) - Xe^{-rT} N(d_2), \tag{4.6}$$

while the value of a put is

$$p = Xe^{-rT} N(-d_2) - S^i e^{[(i-1)(r+i\sigma^2/2)-i(r-b)]T} N(-d_1), \tag{4.7}$$

where

$$d_1 = \frac{\ln(S/X^{1/i}) + (b + (i - \frac{1}{2})\sigma^2)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - i\sigma\sqrt{T}$$

Table 4-2 shows values for call and put power options, for different values of power i, and volatility σ.

Example

Consider a standard power option with three months to expiration, current stock price of 10, power of 2, strike price of 100, risk-free interest rate of 8%, continuous dividend yield of 6%, and expected volatility of the stock of 30%. With $S = 10$, $i = 2$, $X = 100$, $T = 0.5$, $r = 0.08$, $b = 0.08 - 0.06 = 0.02$, and $\sigma = 0.3$, we get

$$d_1 = \frac{\ln(10/100^{1/2}) + (0.02 + (2 - \frac{1}{2})0.3^2)0.5}{0.3\sqrt{0.5}} = 0.3653$$

$$d_2 = 0.3653 - 2 \times 0.3\sqrt{0.5} = -0.0589$$

$$N(d_1) = N(0.3653) = 0.6426, \quad N(d_2) = N(-0.0589) = 0.4765$$

$$c = 10^2 e^{[(2-1)(0.08+2\times0.3^2/2)-2(0.08-0.02)]0.5} N(d_1)$$

$$- 100e^{-0.08\times0.5} N(d_2) = 20.1016$$

TABLE 4-2

Examples of Power Option Values

($S = 10$, $X = 100$, $T = 0.5$, $r = 0.08$, $b = 0.02$)

Call Power Option Values					
i	$\sigma = 10\%$	$\sigma = 15\%$	$\sigma = 20\%$	$\sigma = 25\%$	$\sigma = 30\%$
1.90	0.3102	1.4522	3.2047	5.3446	7.7621
1.95	1.9320	4.2990	6.9724	9.8596	12.9351
2.00	6.7862	9.8585	13.0957	16.5057	20.1016
2.05	15.8587	18.6126	21.8980	25.5429	29.4939
2.10	28.4341	30.4628	33.4555	37.1126	41.2849
Put Power Option Values					
1.90	18.2738	18.9972	20.1600	21.5351	23.0079
1.95	10.2890	12.1467	14.1021	16.0575	17.9810
2.00	4.3539	6.8086	9.1746	11.4533	13.6490
2.05	1.3089	3.3161	5.5476	7.8230	10.0774
2.10	0.2745	1.4031	3.1247	5.1286	7.2508

4.4.3 Capped Power Option

Power options can lead to very high leverage and thus entail potentially very large losses for short positions in these options. It is therefore common to cap the payoff. The maximum payoff is set to some predefined level $\bar{C}$. The payoff at maturity for a capped power call is $\min[\max(S^i - X, 0), \bar{C}]$. Esser (2003) gives the closed-form solution:

$$c = S^i e^{[(i-1)(r+i\sigma^2/2)-i(r-b)]T}[N(e_1) - N(e_3)] \\ - e^{-rT}[XN(e_2) - (\bar{C} + X)N(e_4)], \tag{4.8}$$

where

$$e_1 = \frac{\ln(S/X^{1/i}) + (b + (i - \frac{1}{2})\sigma^2)T}{\sigma\sqrt{T}}$$

$$e_2 = e_1 - i\sigma\sqrt{T}$$

$$e_3 = \frac{\ln(S/(\bar{C} + X)^{1/i}) + (b + (i - \frac{1}{2})\sigma^2)T}{\sigma\sqrt{T}}$$

$$e_4 = e_3 - i\sigma\sqrt{T}$$

In the case of a capped power put, we have

$$p = e^{-rT}[XN(-e_2) - (X - \bar{C})N(-e_4)] \\ - S^i e^{[(i-1)(r+i\sigma^2/2)-i(r-b)]T}[N(-e_1) - N(-e_3)], \tag{4.9}$$

where e_1 and e_2 is as before. e_3 and e_4 has to be changed to

$$e_3 = \frac{\ln(S/(X - \bar{C})^{1/i}) + (b + (i - \frac{1}{2})\sigma^2)T}{\sigma\sqrt{T}}$$

$$e_4 = e_3 - i\sigma\sqrt{T}$$

4.4.4 Powered Option

At maturity, a powered call option pays off $\max[S - X, 0]^i$ and a put pays off $\max[X - S, 0]^i$. Esser (2003) describes how to value these options (see also Jarrow and Turnbull, 1996, Brockhaus, Ferraris, Gallus, Long, Martin, and Overhaus, 1999)

$$c = \sum_{j=0}^{i} \frac{i!}{j!(i-j)!} S^{i-j}(-X)^j e^{(i-j-1)(r+(i-j)\sigma^2/2)T-(i-j)(r-b)T} N(d_{i,j}) \tag{4.10}$$

and

$$p = \sum_{j=0}^{i} \frac{i!}{j!(i-j)!} (-S)^{i-j} X^j e^{(i-j-1)(r+(i-j)\sigma^2/2)T-(i-j)(r-b)T} N(-d_{i,j}) \tag{4.11}$$

where

$$d_{i,j} = \frac{\ln(S/X) + (b + (i - j - \frac{1}{2})\sigma^2)T}{\sigma\sqrt{T}}$$

In this formula, i can take integer values only; for noninteger values, the binomial tree in Chapter 7 can be used.

Table 4-3 shows values for powered call and put options, for different values of power i, and volatility σ.

TABLE 4-3

Powered Option Values

($S = 100$, $X = 100$, $T = 0.5$, $r = 0.1$, $b = 0.07$)

	Call			Put		
	$i = 1$	$i = 2$	$i = 3$	$i = 1$	$i = 2$	$i = 3$
$\sigma = 10\%$	4.7524	53.4487	758.8427	1.3641	9.7580	89.6287
$\sigma = 20\%$	7.3179	160.2955	4,608.7213	3.9296	57.8677	1,061.2120
$\sigma = 30\%$	9.9829	339.3731	15,624.1041	6.5946	142.2726	3,745.1853

Computer algorithm

The computer code returns the value of a powered call or put option.

```
Function PoweredOption(CallPutFlag As String, S As Double, _
  X As Double, T As Double, r As Double, b As Double, _
  v As Double, i As Double) As Double
```

```
Dim d1 As Double, sum As Double
Dim j As Integer

If CallPutFlag = "c" Then
    sum = 0
    For j = 0 To i Step 1
        d1 = (Log(S/X) + (b + (i - j - 0.5) * v^2)*T) / (v*Sqr(T))
        sum = sum + Application.Combin(i, j) _
         * S^(i - j) * (-X)^j * Exp((i - j - 1) _
         * (r + (i - j) * v^2 / 2) * T - (i - j) * (r - b) * T) _
         * CND(d1)
    Next
        PoweredOption = sum
ElseIf CallPutFlag = "p" Then
    sum = 0
    For j = 0 To i Step 1
        d1 = (Log(S/X) + (b + (i - j - 0.5) * v^2) * T) / (v * Sqr(T))
        sum = sum + Application.Combin(i, j) _
       * (-S)^(i - j) * X^j * Exp((i - j - 1) _
       * (r + (i - j) * v^2 / 2) * T - (i - j) * (r - b) * T) _
       * CND(-d1)
    Next
        PoweredOption = sum
End If

End Function
```

Special Case Powered to the Second

In the special case of a powered option with a power of 2, the formulas above simplify to Crack (1997, 2004)

$$c = S^2 e^{(2b-r+\sigma^2)T} N(d_0) - 2XSe^{(b-r)T} N(d_1) + X^2 e^{-rT} N(d_2), \qquad (4.12)$$

where

$$d_0 = \frac{\ln(S/X) + (b + \frac{3}{2}\sigma^2)T}{\sigma\sqrt{T}}$$

$$d_1 = \frac{\ln(S/X) + (b + \sigma^2/2)T}{\sigma\sqrt{T}} \qquad d_2 = d_1 - \sigma\sqrt{T}$$

and the put value is given by

$$p = S^2 e^{(2b-r+\sigma^2)T} N(-d_0) - 2XSe^{(b-r)T} N(-d_1) + X^2 e^{-rT} N(-d_2) \qquad (4.13)$$

4.5 LOG CONTRACTS

A log contract, first introduced by Neuberger (1994) and Neuberger (1996), is not strictly an option. It is, however, an important building block in volatility derivatives (see Chapter 6 as well as Demeterfi, Derman, Kamal, and Zou, 1999). The payoff from a log contract at maturity T is simply the natural logarithm of the underlying asset divided by the strike price, $\ln(S/X)$. The payoff is thus nonlinear and has many similarities with options. The value of this contract is

$$L = e^{-rT}[\ln(S/X) + (b - \sigma^2/2)T] \tag{4.14}$$

The delta of a log contract is

$$\Delta = \frac{e^{-rT}}{S}$$

and the gamma is

$$\Gamma = -\frac{e^{-rT}}{S^2}$$

Example
What is the current value of a log contract with three months to expiration on a stock trading at 90, with strike 80, risk-free rate 8%, and volatility 35%? $S = 90$, $X = 80$, $T = 0.25$, $r = 0.08$, $b = 0.08$, and $\sigma = 0.35$

$$L = e^{-0.08\times 0.25}[\ln(90/80) + (0.08 - 0.35^2/2)0.25] = 0.1200$$

4.5.1 Log(S) Contract

An even simpler version of the log contract is when the payoff simply is $\ln(S)$. The payoff is clearly still nonlinear in the underlying asset. It follows that the value of this contract is

$$L = e^{-rT}[\ln(S) + (b - \sigma^2/2)T] \tag{4.15}$$

The theta/time decay of a log contract is

$$\theta = -\frac{1}{T}\sigma^2$$

and its exposure to the stock price, delta, is

$$\Delta = -\frac{2}{T}\frac{1}{S}$$

This basically tells you that you need to be long $\frac{2}{T}$ stocks to be delta-neutral at any time. Moreover, the gamma is

$$\Gamma = \frac{2}{TS^2}$$

Example
What is the current value of a Log(S) contract with the same parameters used in the previous log contract example?

$$L = e^{-0.08\times 0.25}[\ln(90) + (0.08 - 0.35^2/2)0.5] = 4.4153$$

TABLE 4-4

Log Option Values

($S = 100, T = 0.75, r = 0.08, b = 0.04$)

σ	$X = 70$	$X = 80$	$X = 90$	$X = 100$	$X = 110$	$X = 120$	$X = 130$
20%	0.3510	0.2306	0.1369	0.0724	0.0341	0.0145	0.0056
30%	0.3422	0.2338	0.1528	0.0959	0.0580	0.0340	0.0195
40%	0.3379	0.2408	0.1687	0.1165	0.0796	0.0539	0.0363
50%	0.3365	0.2486	0.1830	0.1344	0.0986	0.0724	0.0532
60%	0.3362	0.2559	0.1954	0.1498	0.1152	0.0890	0.0691

4.5.2 Log Option

A log option introduced by Wilmott (2000) has a payoff at maturity equal to $\max[\ln(S/X), 0]$, which is basically an option on the rate of return on the underlying asset with strike $\ln(X)$. The value of a log option is given by[1]

$$c = e^{-rT} n(d_2)\sigma\sqrt{T} + e^{-rT}\left[\ln(S/X) + (b - \sigma^2/2)T\right] N(d_2), \tag{4.16}$$

where $N(\cdot)$ is the cumulative normal distribution function, $n(\cdot)$ is the normal density function, and

$$d_2 = \frac{\ln(S/X) + (b - \sigma^2/2)T}{\sigma\sqrt{T}}$$

Table 4-4 shows values of log options for various strikes and volatilities.

4.6 FORWARD START OPTIONS

A forward start option with time to maturity T starts at-the-money or proportionally in- or out-of-the-money after a known elapsed time t in the future. The strike is set equal to a positive constant α times the asset price S after the known time t. If α is less than unity, the call (put) will start $1 - \alpha$ percent in-the-money (out-of-the-money); if α is unity, the option will start at-the-money; and if α is larger than unity, the call (put) will start $\alpha - 1$ percentage out-of-the-money (in-the-money).A forward start option can be priced using the

[1]The formula presented here is slightly different due to a minor typo in the formula in Paul Wilmott's brilliant book on *Quantitative Finance*.

Rubinstein (1990) formula:

$$c = Se^{(b-r)t}[e^{(b-r)(T-t)}N(d_1) - \alpha e^{-r(T-t)}N(d_2)] \tag{4.17}$$

$$p = Se^{(b-r)t}[\alpha e^{-r(T-t)}N(-d_2) - e^{(b-r)(T-t)}N(-d_1)], \tag{4.18}$$

where

$$d_1 = \frac{\ln(1/\alpha) + (b + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}} \qquad d_2 = d_1 - \sigma\sqrt{T-t}$$

Application
Employee options are often of the forward starting type. Ratchet options (aka cliquet options) consist of a series of forward starting options.

Example
Consider an employee who receives a call option with forward start three months from today. The option starts 10% out-of-the-money, time to maturity is one year from today, the stock price is 60, the risk-free interest rate is 8%, the continuous dividend yield is 4%, and the expected volatility of the stock is 30%. $S = 60$, $\alpha = 1.1$, $t = 0.25$, $T = 1, r = 0.08, b = 0.08 - 0.04 = 0.04, \sigma = 0.3$.

$$d_1 = \frac{\ln(1/1.1) + (0.04 + 0.3^2/2)(1-0.25)}{0.3\sqrt{(1-0.25)}} = -0.1215$$

$$d_2 = d_1 - 0.3\sqrt{(1-0.25)} = -0.3813$$

$$N(d_1) = N(-0.1215) = 0.4517 \qquad N(d_2) = N(-0.3813) = 0.3515$$

$$c = 60e^{(0.04-0.08)0.25}[e^{(0.04-0.08)(1-0.25)}N(d_1) - 1.1e^{-0.08(1-0.25)}N(d_2)] = 4.4064$$

4.7 FADE-IN OPTION

A fade-in call has the same payoff as a standard call except the size of the payoff is weighted by how many fixings the asset price were inside a predefined range (L, U). If the asset price is inside the range for every fixing, the payoff will be identical to a plain vanilla option. More precisely, for a call option, the payoff will be $\max(S_T - X, 0) \times \frac{1}{n}\sum_{i=1}^{n} \eta(i)$, where n is the total number of fixings and $\eta(i) = 1$ if at fixing i the asset price is inside the range, and $\eta(i) = 0$ otherwise. Similarly, for a put, the payoff is $\max[X - S_T, 0] \times \frac{1}{n}\sum_{i=1}^{n} \eta(i)$.

Brockhaus, Ferraris, Gallus, Long, Martin, and Overhaus (1999)[2] describe a closed-form formula for fade-in options. For a call the value

[2]But I think they have a typo in their formula, so here is my version.

is given by

$$c = \frac{1}{n}\sum_{i=1}^{n} S^{(b-r)T}[M(-d_5, d_1; -\rho) - M(-d_3, d_1; -\rho)]$$
$$- Xe^{-rT}[M(-d_6, d_2; -\rho) - M(-d_4, d_2; -\rho)], \qquad (4.19)$$

where n is the number of fixings, $\rho = \frac{\sqrt{t_1}}{\sqrt{T}}$, $t_1 = \frac{iT}{n}$

$$d_1 = \frac{\ln(S/X) + (b + \sigma^2/2)T}{\sigma\sqrt{T}} \qquad d_2 = d_1 - \sigma\sqrt{T}$$

$$d_3 = \frac{\ln(S/L) + (b + \sigma^2/2)t_1}{\sigma\sqrt{t_1}} \qquad d_4 = d_3 - \sigma\sqrt{t_1}$$

$$d_5 = \frac{\ln(S/U) + (b + \sigma^2/2)t_1}{\sigma\sqrt{t_1}} \qquad d_6 = d_5 - \sigma\sqrt{t_1}$$

The value of a put is similarly

$$p = \frac{1}{n}\sum_{i=1}^{n} Xe^{-rT}[M(-d_6, -d_2; \rho) - M(-d_4, -d_2; \rho)]$$
$$- S^{(b-r)T}[M(-d_5, -d_1; \rho) - M(-d_3, -d_1; \rho)] \qquad (4.20)$$

Table 4-5 shows some possible values of fade-in call options. When we have a wide range far away from the current asset price and the volatility is relatively low, the fade-in value converges to the value of a plain vanilla option. The latter is given in the last row (BSM: Black-Scholes-Merton values). For simplicity, we are assuming a leap year, as this makes 183 fixings equal to half a year. We moreover ignore adjustments for the market being closed during weekends.

TABLE 4-5

Examples of Fade-in Call Option Values

($S = 100$, $X = 100$, $T = 0.5$, $r = 0.1$, $b = 0$, $n = 183$)

L	U	$\sigma = 0.1$	$\sigma = 0.15$	$\sigma = 0.2$	$\sigma = 0.25$	$\sigma = 0.3$	$\sigma = 0.35$	$\sigma = 0.4$
95	105	1.5427	1.7228	1.8353	1.9118	1.9663	2.0064	2.0364
90	110	2.2929	2.7573	3.0387	3.2410	3.3968	3.5206	3.6207
85	115	2.5864	3.4034	3.8948	4.2314	4.4887	4.6974	4.8716
80	120	2.6649	3.7549	4.4901	4.9887	5.3564	5.6489	5.8931
75	125	2.6802	3.9189	4.8735	5.5533	6.0497	6.4344	6.7493
50	150	2.6828	4.0229	5.3479	6.5956	7.6934	8.6153	9.3749
BSM:		2.6828	4.0232	5.3623	6.6997	8.0350	9.3679	10.6978

4.8 RATCHET OPTIONS

A ratchet option (aka moving strike option or cliquet option) consists of a series of forward starting options. The strike price for the next exercise date equals a positive constant times the asset price as of the previous exercise date. For instance, a one-year ratchet call option with quarterly payments will normally have four payments (exercise dates) equal to the difference between the asset price and the strike price fixed at the previous exercise date. The strike price of the first option is usually set equal to the asset price at the time the contract is initiated. A ratchet option can be priced as the sum of forward starting options.

$$c = \sum_{i=1}^{n} Se^{(b-r)t_i}[e^{(b-r)(T_i-t_i)}N(d_1) - \alpha e^{-r(T_i-t_i)}N(d_2)], \tag{4.21}$$

where n is the number of settlements, t_i is the time to the forward start or strike fixing, and T_i is the time to maturity of the forward starting option. A ratchet put is similar to a sum of forward starting puts.

4.9 RESET STRIKE OPTIONS—TYPE 1

In a reset call (put) option, the strike is reset to the asset price at a predetermined future time, if the asset price is below (above) the initial strike price. This makes the strike path-dependent. The payoff for a call at maturity is equal to $\max[\frac{S-\hat{X}}{\hat{X}}, 0]$, where $\hat{X}$ is equal to the original strike X if not reset, and equal to the reset strike if reset. Similarly, for a put, the payoff is $\max[\frac{\hat{X}-S}{\hat{X}}, 0]$. Gray and Whaley (1997) have derived a closed-form solution for such an option. For a call, we have

$$c = e^{(b-r)(T-\tau)}N(-a_2)N(z_1)e^{-r\tau} - e^{-rT}N(-a_2)N(z_2) - e^{-rT}M(a_2, y_2; \rho) + \left(\frac{S}{X}\right)e^{(b-r)T}M(a_1, y_1; \rho) \tag{4.22}$$

and for a put,

$$p = e^{-rT}N(a_2)N(-z_2) - e^{(b-r)(T-\tau)}N(a_2)N(-z_1)e^{-r\tau} + e^{-rT}M(-a_2, -y_2; \rho) - \left(\frac{S}{X}\right)e^{(b-r)T}M(-a_1, -y_1; \rho) \tag{4.23}$$

where b is the cost-of-carry of the underlying asset, σ is the volatility of the relative price changes in the asset, and r is the risk-free interest rate. X is the strike price of the option, τ the time to reset (in

TABLE 4-6

Examples of Reset Strike Option Type 1 Values

($S = 100, X = 100, T = 1, r = 0.1, b = 0.1$)

σ	$\tau = 0.1$	$\tau = 0.2$	$\tau = 0.3$	$\tau = 0.4$	$\tau = 0.5$	$\tau = 0.6$	$\tau = 0.7$	$\tau = 0.8$	$\tau = 0.9$
Call Option Values									
10%	0.1095	0.1102	0.1100	0.1095	0.1089	0.1082	0.1073	0.1064	0.1053
20%	0.1459	0.1484	0.1493	0.1492	0.1485	0.1473	0.1457	0.1435	0.1404
30%	0.1871	0.1916	0.1935	0.1940	0.1934	0.1919	0.1896	0.1862	0.1811
40%	0.2300	0.2365	0.2395	0.2405	0.2400	0.2382	0.2352	0.2306	0.2235
Put Option Values									
10%	0.0100	0.0110	0.0118	0.0125	0.0132	0.0137	0.0141	0.0142	0.0136
20%	0.0441	0.0466	0.0484	0.0497	0.0505	0.0510	0.0510	0.0503	0.0482
30%	0.0830	0.0867	0.0891	0.0907	0.0916	0.0920	0.0916	0.0901	0.0868
40%	0.1225	0.1272	0.1301	0.1320	0.1330	0.1332	0.1325	0.1305	0.1261

years), and T is its time to expiration. $N(x)$ and $M(a, b; \rho)$ are, respectively, the univariate and bivariate cumulative normal distribution functions. The remaining parameters are $\rho = \sqrt{\tau/T}$ and

$$a_1 = \frac{\ln(S/X) + (b + \sigma^2/2)\tau}{\sigma\sqrt{\tau}} \qquad a_2 = a_1 - \sigma\sqrt{\tau}$$

$$z_1 = \frac{(b + \sigma^2/2)(T - \tau)}{\sigma\sqrt{T - \tau}} \qquad z_2 = z_1 - \sigma\sqrt{T - \tau}$$

$$y_1 = \frac{\ln(S/X) + (b + \sigma^2/2)T}{\sigma\sqrt{T}} \qquad y_2 = y_1 - \sigma\sqrt{T}$$

Table 4-6 shows reset strike type 1 option values for various time to reset τ and volatilities σ.

4.10 RESET STRIKE OPTIONS—TYPE 2

For a reset option type 2, the strike is reset in a similar way as a reset option 1. That is, the strike is reset to the asset price at a predetermined future time, if the asset price is below (above) the initial strike price for a call (put). The payoff for such a reset call is $\max[S - \hat{X}, 0]$, and $\max[\hat{X} - S, 0]$ for a put, where $\hat{X}$ is equal to the original strike X if not reset, and equal to the reset strike if reset. Gray and Whaley (1999) have derived a closed-form solution for the price of European reset strike options. The price of the call option is then given by

$$c = Se^{(b-r)T}M(a_1, y_1; \rho) - Xe^{-rT}M(a_2, y_2; \rho)$$
$$- Se^{(b-r)\tau}N(-a_1)N(z_2)e^{-r(T-\tau)} + Se^{(b-r)T}N(-a_1)N(z_1),$$

while the price of the put option is given by

$$p = Se^{(b-r)\tau} N(a_1)N(-z_2)e^{-r(T-\tau)} - Se^{(b-r)T} N(a_1)N(-z_1)$$
$$+ Xe^{-rT} M(-a_2, -y_2; \rho) - Se^{(b-r)T} M(-a_1, -y_1; \rho),$$

where b is the cost-of-carry of the underlying asset, σ is the volatility of the relative price changes in the asset, and r is the risk-free interest rate. X is the strike price of the option, τ the time to reset (in years), and T is its time to expiration. $N(x)$ and $M(a, b; \rho)$ are, respectively, the univariate and bivariate cumulative normal distribution functions. Further

$$a_1 = \frac{\ln(S/X) + (b + \sigma^2/2)\tau}{\sigma\sqrt{\tau}} \qquad a_2 = a_1 - \sigma\sqrt{\tau}$$

$$z_1 = \frac{(b + \sigma^2/2)(T - \tau)}{\sigma\sqrt{T - \tau}} \qquad z_2 = z_1 - \sigma\sqrt{T - \tau}$$

$$y_1 = \frac{\ln(S/X) + (b + \sigma^2/2)T}{\sigma\sqrt{T}} \qquad y_2 = y_1 - \sigma\sqrt{T}$$

and $\rho = \sqrt{\tau/T}$. For reset options with multiple reset rights, see Dai, Kwok, and Wu (2003) and Liao and Wang (2003).

Table 4-7 shows reset strike type 2 option values for various time to reset τ, and volatilities σ.

TABLE 4-7

Examples of Reset Strike Option Type 2 Values

($S = 100, X = 100, T = 1, r = 0.1, b = 0.1$)

σ	$\tau = 0.1$	$\tau = 0.2$	$\tau = 0.3$	$\tau = 0.4$	$\tau = 0.5$	$\tau = 0.6$	$\tau = 0.7$	$\tau = 0.8$	$\tau = 0.9$
Call Option Values									
10%	10.8771	10.9340	10.9248	10.8870	10.8332	10.7690	10.6960	10.6132	10.5135
20%	14.3348	14.5434	14.6162	14.6176	14.5700	14.4822	14.3551	14.1810	13.9311
30%	18.1993	18.5294	18.6722	18.7122	18.6788	18.5822	18.4215	18.1817	17.8121
40%	22.1289	22.5615	22.7635	22.8372	22.8175	22.7159	22.5294	22.2364	21.7663
Put Option Values									
10%	1.0123	1.1224	1.2156	1.2991	1.3739	1.4378	1.4845	1.4991	1.4382
20%	4.5280	4.8344	5.0537	5.2180	5.3353	5.4032	5.4098	5.3261	5.0706
30%	8.5981	9.1012	9.4388	9.6728	9.8202	9.8801	9.8367	9.6484	9.1981
40%	12.8305	13.5408	14.0016	14.3066	14.4832	14.5324	14.4344	14.1348	13.4791

4.11 TIME-SWITCH OPTIONS

In a discrete time-switch call option, introduced by Pechtl (1995),[3] the investor receives an amount $A\,\Delta t$ at maturity T for each time interval Δt the corresponding asset price $S_{i\Delta t}$ has exceeded the strike price X. The discrete time-switch put option gives a similar payoff $A\,\Delta t$ at maturity for each time interval the asset price $S_{i\Delta t}$ has been below the strike price X.

$$c = Ae^{-rT}\sum_{i=1}^{n} N\left(\frac{\ln(S/X) + (b - \sigma^2/2)i\,\Delta t}{\sigma\sqrt{i\,\Delta t}}\right)\Delta t \tag{4.24}$$

$$p = Ae^{-rT}\sum_{i=1}^{n} N\left(\frac{-\ln(S/X) - (b - \sigma^2/2)i\,\Delta t}{\sigma\sqrt{i\,\Delta t}}\right)\Delta t \tag{4.25}$$

where $n = T/\Delta t$. If some of the option's total lifetime has already passed, it is necessary to add a fixed amount $\Delta t A e^{-rT} m$ to the option-pricing formula, where m is the number of time units where the option already has fulfilled its condition.

Example

What is the price of a call time-switch option with one year to expiration, where the investor accumulates $5 \times 1/365$ for each day the stock price exceeds the strike price of 110? The stock price is currently 100, the risk-free rate is 6%, and the volatility is 26%. $S = 100$, $A = 5$, $X = 110$, $T = 1$, $\Delta t = 1/365$, $r = b = 0.06$, $\sigma = 0.26$, $n = T/\Delta t = 365$, $m = 0$ yields

$$c = 5e^{-0.06\times 1}\sum_{i=1}^{365} N\left(\frac{\ln(100/110) + (0.05 - 0.26^2/2)i \times 1/365}{0.26\sqrt{i \times 1/365}}\right)\frac{1}{365} = 1.3750$$

Computer algorithm

```
Function TimeSwitchOption(CallPutFlag As String, S As Double, _
  X As Double, a As Double, T As Double, m As Integer, dt As Double, _
  r As Double, b As Double, v As Double) As Double

    Dim sum As Double, d As Double
    Dim i As Integer, n As Integer, Z As Integer

    n = T / dt
    sum = 0
    If CallPutFlag = "c" Then
        Z = 1
```

[3] In the same paper Pechtl (1995) shows how to value continuous time-switch options.

```
    ElseIf CallPutFlag = "p" Then
        Z = -1
    End If
    For i = 1 To n
        d = (Log(S/X) + (b - v^2 / 2) * i * dt) / (v * Sqr(i * dt))
        sum = sum + CND(Z * d) * dt
    Next
    TimeSwitchOption = a * Exp(-r * T) * sum + dt * a * Exp(-r * T) * m

End Function
```

Application

Discrete time-switch options have recently become quite popular in the interest rate markets in the form of so-called accrual swaps. In a typical accrual swap, the fixed-rate receiver accumulates an amount equal to the notional of the swap times the fixed rate only for days when the floating rate is below or above a certain level. Accrual swaps can easily be priced as the sum of discrete time-switch options under the assumptions of lognormally distributed forward rates with zero drift.

4.12 CHOOSER OPTIONS

4.12.1 Simple Chooser Options

A simple chooser option gives the holder the right to choose whether the option is to be a standard call or put after a time t_1, with strike X and time to maturity T_2. The payoff from a simple chooser option at time t_1 ($t_1 < T_2$) is

$$w(S, X, t_1, T_2) = \max[c_{BSM}(S, X, T_2), p_{BSM}(S, X, T_2)],$$

where $c_{BSM}(S, X, T_2)$ and $p_{BSM}(S, X, T_2)$ are the general Black-Scholes-Merton call and put formulas. A simple chooser option can be priced using the formula originally published by Rubinstein (1991c):

$$w = Se^{(b-r)T_2}N(d) - Xe^{-rT_2}N(d - \sigma\sqrt{T_2}) - Se^{(b-r)T_2}N(-y) + Xe^{-rT_2}N(-y + \sigma\sqrt{t_1}), \tag{4.26}$$

where

$$d = \frac{\ln(S/X) + (b + \sigma^2/2)T_2}{\sigma\sqrt{T_2}} \qquad y = \frac{\ln(S/X) + bT_2 + \sigma^2 t_1/2}{\sigma\sqrt{t_1}}$$

Example

Consider a simple chooser option with six months to expiration and three months to choose between a put or call. The underlying stock price is 50, the strike price is 50, the risk-free interest rate is 8%

per year, and the volatility per year is 25%. $S = 50$, $X = 50$, $T_2 = 0.5$, $t_1 = 0.25$, $r = 0.08$, $b = 0.08$, and $\sigma = 0.25$.

$$d = \frac{\ln(50/50) + (0.08 + 0.25^2/2)0.5}{0.25\sqrt{0.5}} = 0.3147$$

$$y = \frac{\ln(50/50) + 0.08 \times 0.5 + 0.25^2 \times 0.25/2}{0.25\sqrt{0.25}} = 0.3825$$

$$N(d) = N(0.3147) = 0.6235$$

$$N(d - \sigma\sqrt{T_2}) = N(0.3147 - 0.25\sqrt{0.5}) = 0.5548$$

$$N(-y) = N(-0.3825) = 0.3510$$

$$N(-y + \sigma\sqrt{t_1}) = N(-0.3825 + 0.25\sqrt{0.25}) = 0.3984$$

$$w = 50e^{(0.08-0.08)0.5}N(d) - 50e^{-0.08\times0.5}N(d - 0.25\sqrt{0.5})$$
$$- 50e^{(0.08-0.08)0.5}N(-y) + 50e^{-0.08\times0.5}N(-y + 0.25\sqrt{0.25}) = 6.1071$$

4.12.2 Complex Chooser Options

A complex chooser option, introduced by Rubinstein (1991c),[4] gives the holder the right to choose whether the option is to be a standard call option after a time t, with time to expiration T_c and strike X_c, or a put option with time to maturity T_p and strike X_p. The payoff from a complex chooser option at time t ($t < T_c, T_p$) is

$$w(S, X_c, X_p, t, T_c, T_p) = \max[c_{BSM}(S, X_c, T_c), p_{BSM}(S, X_p, T_p)],$$

where $c_{BSM}(S, X, T)$ and $p_{BSM}(S, X, T)$ are the BSM call and put formulas, respectively.

$$w = Se^{(b-r)T_c}M(d_1, y_1; \rho_1)$$
$$- X_c e^{-rT_c}M(d_2, y_1 - \sigma\sqrt{T_c}; \rho_1) - Se^{(b-r)T_p}M(-d_1, -y_2; \rho_2)$$
$$+ X_p e^{-rT_p}M(-d_2, -y_2 + \sigma\sqrt{T_p}; \rho_2), \quad (4.27)$$

[4]See also Nelken (1993).

where T_c is the time to maturity on the call, and T_p is the time to maturity on the put.

$$d_1 = \frac{\ln(S/I) + (b + \sigma^2/2)t}{\sigma\sqrt{t}} \qquad d_2 = d_1 - \sigma\sqrt{t}$$

$$y_1 = \frac{\ln(S/X_c) + (b + \sigma^2/2)T_c}{\sigma\sqrt{T_c}} \qquad y_2 = \frac{\ln(S/X_p) + (b + \sigma^2/2)T_p}{\sigma\sqrt{T_p}}$$

$$\rho_1 = \sqrt{t/T_c} \qquad \rho_2 = \sqrt{t/T_p},$$

and I is the solution to

$$Ie^{(b-r)(T_c-t)}N(z_1) - X_c e^{-r(T_c-t)}N(z_1 - \sigma\sqrt{T_c - t}) +$$

$$Ie^{(b-r)(T_p-t)}N(-z_2) - X_p e^{-r(T_p-t)}N(-z_2 + \sigma\sqrt{T_p - t}) = 0$$

$$z_1 = \frac{\ln(I/X_c) + (b + \sigma^2/2)(T_c - t)}{\sigma\sqrt{T_c - t}} \qquad z_2 = \frac{\ln(I/X_p) + (b + \sigma^2/2)(T_p - t)}{\sigma\sqrt{T_p - t}}$$

Example

Consider a complex chooser option that gives the holder the right to choose whether the option is to be a call with six months to expiration and strike price 55, or a put with seven months to expiration and strike price 48. The time to choose between a put or call is in three months, the underlying stock price is 50, the risk-free interest rate per year is 10%, the dividend yield is 5% per year, and the volatility per year is 35%. $S = 50$, $X_c = 55$, $X_p = 48$, $T_c = 0.5$, $T_p = 0.5833$, $t = 0.25$, $r = 0.1$, $b = 0.1 - 0.05 = 0.05$, and $\sigma = 0.35$.

$$d_1 = \frac{\ln(50/51.1158) + (0.05 + 0.35^2/2)0.25}{0.35\sqrt{0.25}} = 0.0328,$$

where a Newton-Raphson search gives the solution to the critical value $I = 51.1158$, and

$$d_2 = d_1 - 0.35\sqrt{0.25} = -0.1422$$

$$y_1 = \frac{\ln(50/55) + (0.05 + 0.35^2/2)0.5}{0.35\sqrt{0.5}} = -0.1604$$

$$y_2 = \frac{\ln(50/48) + (0.05 + 0.35^2/2)0.5833}{0.35\sqrt{0.5833}} = 0.3955$$

$$\rho_1 = \sqrt{0.25/0.5} = 0.7071 \qquad \rho_2 = \sqrt{0.25/0.5833} = 0.6547$$

$$M(d_1, y_1; \rho_1) = 0.3464 \qquad M(d_2, y_1 - 0.35\sqrt{0.5}; \rho_1) = 0.2660$$

$$M(-d_1, -y_2; \rho_2) = 0.2725 \qquad M(-d_2, -y_2 + 0.35\sqrt{0.5833}; \rho_2) = 0.3601$$

$$\begin{aligned} w = {} & 50e^{(0.05-0.1)0.5} M(d_1, y_1; \rho_1) \\ & - 55e^{-0.1\times 0.5} M(d_2, y_{\rm i} - \sigma\sqrt{T_c}; \rho_1) \\ & - 50e^{(0.05-0.1)0.5833} M(-d_1, -y_2; \rho_2) \\ & + 48e^{-0.05\times 0.5833} M(-d_2, -y_2 + \sigma\sqrt{T_p}; \rho_2) = 6.0508 \end{aligned}$$

Computer algorithm

The computer code returns the value of a complex chooser option.

```
Function ComplexChooser(S As Double, Xc As Double, Xp As Double, _
  T As Double, Tc As Double, Tp As Double, _
  r As Double, b As Double, v As Double) As Double

    Dim d1 As Double, d2 As Double, y1 As Double, y2 As Double
    Dim rho1 As Double, rho2 As Double, i As Double

    i = CriticalValueChooser(S, Xc, Xp, T, Tc, Tp, r, b, v)
    d1 = (Log(S/i) + (b + v^2 / 2) * T) / (v * Sqr(T))
    d2 = d1 - v * Sqr(T)
    y1 = (Log(S/Xc) + (b + v^2 / 2) * Tc) / (v * Sqr(Tc))
    y2 = (Log(S/Xp) + (b + v^2 / 2) * Tp) / (v * Sqr(Tp))
    rho1 = Sqr(T / Tc)
    rho2 = Sqr(T / Tp)

    ComplexChooser = S * Exp((b - r) * Tc) * CBND(d1, y1, rho1) _
    - Xc * Exp(-r * Tc) * CBND(d2, y1 - v * Sqr(Tc), rho1) _
    - S * Exp((b - r) * Tp) * CBND(-d1, -y2, rho2) _
    + Xp * Exp(-r * Tp) * CBND(-d2, -y2 + v * Sqr(Tp), rho2)

End Function
```

The critical stock value I is found by calling the function *CriticalValueChooser*($\cdot$) below, which is based on the Newton-Raphson algorithm.

```
Function CriticalValueChooser(S As Double, Xc As Double, _
Xp As Double, T As Double, Tc As Double, Tp As Double, _
r As Double, b As Double, v As Double) As Double

    Dim Sv As Double, ci As Double, Pi As Double, epsilon As Double
    Dim dc As Double, dp As Double, yi As Double, di As Double

    Sv = S

    ci = GBlackScholes("c", Sv, Xc, Tc - T, r, b, v)
    Pi = GBlackScholes("p", Sv, Xp, Tp - T, r, b, v)
    dc = GDelta("c", Sv, Xc, Tc - T, r, b, v)
    dp = GDelta("p", Sv, Xp, Tp - T, r, b, v)
    yi = ci - Pi
```

```
    di = dc - dp
    epsilon = 0.001

    'Newton-Raphson s¿keprosess
    While Abs(yi) > epsilon
        Sv = Sv - yi / di
        ci = GBlackScholes("c", Sv, Xc, Tc - T, r, b, v)
        Pi = GBlackScholes("p", Sv, Xp, Tp - T, r, b, v)
        dc = GDelta("c", Sv, Xc, Tc - T, r, b, v)
        dp = GDelta("p", Sv, Xp, Tp - T, r, b, v)
        yi = ci - Pi
        di = dc - dp
    Wend
    CriticalValueChooser = Sv

End Function
```

where $CND(\cdot)$ is the cumulative normal distribution function, and $CBND(\cdot)$ is the cumulative bivariate normal distribution function described in Chapter 13. Example: *ComplexChooser*(50, 55, 48, 0.25, 0.5, 0.5833, 0.1, 0.05, 0.35) returns a chooser value of 6.0508 as in the numerical example above.

4.13 OPTIONS ON OPTIONS

A model for pricing options on options was first published by Geske (1977). It was later extended and discussed by Geske (1979), Hodges and Selby (1987), and Rubinstein (1991a), among others.

Call on Call

Payoff: $\max[c_{BSM}(S, X_1, T_2) - X_2; 0]$, where X_1 is the strike price of the underlying option, X_2 is the strike price of the option on the option, and $c_{BSM}(S, X_1, T_2)$ is the generalized BSM call option formula with strike X_1 and time to maturity T_2. The value is

$$c_{call} = Se^{(b-r)T_2} M(z_1, y_1; \rho) - X_1 e^{-rT_2} M(z_2, y_2; \rho) - X_2 e^{-rt_1} N(y_2), \quad (4.28)$$

where

$$y_1 = \frac{\ln(S/I) + (b + \sigma^2/2)t_1}{\sigma\sqrt{t_1}} \qquad y_2 = y_1 - \sigma\sqrt{t_1}$$

$$z_1 = \frac{\ln(S/X_1) + (b + \sigma^2/2)T_2}{\sigma\sqrt{T_2}} \qquad z_2 = z_1 - \sigma\sqrt{T_2}$$

$$\rho = \sqrt{t_1/T_2},$$

where T_2 is the time to maturity on the underlying option, and t_1 is the time to maturity on the option on the option.

Put on Call
Payoff: $\max[X_2 - c_{BSM}(S, X_1, T_2); 0]$

$$p_{call} = X_1 e^{-rT_2} M(z_2, -y_2; -\rho) - S e^{(b-r)T_2} M(z_1, -y_1; -\rho) + X_2 e^{-rt_1} N(-y_2), \tag{4.29}$$

where the value of I is found by solving the equation

$$c_{BSM}(I, X_1, T_2 - t_1) = X_2$$

Call on Put
Payoff: $\max[p_{BSM}(S, X_1, T_2) - X_2; 0]$

$$c_{put} = X_1 e^{-rT_2} M(-z_2, -y_2; \rho) - S e^{(b-r)T_2} M(-z_1, -y_1; \rho) - X_2 e^{-rt_1} N(-y_2) \tag{4.30}$$

Put on Put
Payoff: $\max[X_2 - p_{BSM}(S, X_1, T_2); 0]$

$$p_{put} = S e^{(b-r)T_2} M(-z_1, y_1; -\rho) - X_1 e^{-rT_2} M(-z_2, y_2; -\rho) + X_2 e^{-rt_1} N(y_2), \tag{4.31}$$

where the value of I is found by solving the equation

$$p_{BSM}(I, X_1, T_2 - t_1) = X_2$$

Example
Consider a put-on-call option that gives the option holder the right to sell a call option for 50, three months from today. The strike on the underlying call option is 520, the time to maturity on the call is six months from today, the price on the underlying stock index is 500, the risk-free interest rate is 8%, and the stock index pays dividends at a rate of 3% annually and has a volatility of 35%. $S = 500$, $X_1 = 520$, $X_2 = 50$, $t_1 = 0.25$, $T_2 = 0.5$, $r = 0.08$, $b = 0.08 - 0.03 = 0.05$, and $\sigma = 0.35$.

The critical value I is

$$c_{BSM}(I, X_1, T_2 - t_1) = X_2$$

$$c_{BSM}(I, 520, 0.5 - 0.25) = 50$$

$$I = 538.3165$$

$$y_1 = \frac{\ln(500/538.3165) + (0.05 + 0.35^2/2)0.25}{0.35\sqrt{0.25}} = -0.2630$$

$$y_2 = y_1 - 0.35\sqrt{0.25} = -0.4380$$

$$z_1 = \frac{\ln(500/520) + (0.05 + 0.35^2/2)0.5}{0.35\sqrt{0.5}} = 0.0663$$

$$z_2 = z_1 - 0.35\sqrt{0.5} = -0.1812$$

$$\rho = \sqrt{0.25/0.5} = 0.7071$$

$$M(z_2, -y_2; -\rho) = 0.1736 \qquad M(z_1, -y_1; -\rho) = 0.1996$$

$$N(-y_2) = 0.6693$$

$$p_{call} = 520e^{-0.08\times0.5} M(z_2, -y_2; -\rho) - 500e^{(0.05-0.08)0.5} M(z_1, -y_1; -\rho) + 50e^{-0.08\times0.25} N(-y_2) = 21.1965$$

Computer algorithm

The computer algorithm *OptionsOnOptions*$(\cdot)$ returns the value of an option on an option. Setting the *TypeFlag* equal to "*cc*" gives a call on call, "*cp*" gives a call on a put, "*pp*" gives a put on a put, and "*pc*" gives a put on a call.

```
Function OptionsOnOptions(TypeFlag As String, S As Double, X1 As Double, _
  X2 As Double, t1 As Double, T2 As Double, r As Double, _
  b As Double, v As Double) As Double

    Dim y1 As Double, y2 As Double, z1 As Double, z2 As Double
    Dim i As Double, rho As Double, CallPutFlag As String

    If TypeFlag = "cc" Or TypeFlag = "pc" Then
        CallPutFlag = "c"
    Else
        CallPutFlag = "p"
    End If

    i = CriticalValueOptionsOnOptions(CallPutFlag, X1, X2, T2 - t1, r, b, v)

    rho = Sqr(t1 / T2)
    y1 = (Log(S/i) + (b + v^2 / 2) * t1) / (v * Sqr(t1))
    y2 = y1 - v * Sqr(t1)
    z1 = (Log(S/X1) + (b + v^2 / 2) * T2) / (v * Sqr(T2))
    z2 = z1 - v * Sqr(T2)

    If TypeFlag = "cc" Then
        OptionsOnOptions = S * Exp((b - r) * T2) * CBND(z1, y1, rho)
         - X1 * Exp(-r * T2) * CBND(z2, y2, rho) _
         - X2 * Exp(-r * t1) * CND(y2)
    ElseIf TypeFlag = "pc" Then
        OptionsOnOptions = X1 * Exp(-r * T2) * CBND(z2, -y2, -rho) _
         - S * Exp((b - r) * T2) * CBND(z1, -y1, -rho) _
         + X2 * Exp(-r * t1) * CND(-y2)
    ElseIf TypeFlag = "cp" Then
        OptionsOnOptions = X1 * Exp(-r * T2) * CBND(-z2, -y2, rho) _
         - S * Exp((b - r) * T2) * CBND(-z1, -y1, rho) _
         - X2 * Exp(-r * t1) * CND(-y2)
    ElseIf TypeFlag = "pp" Then
        OptionsOnOptions = S * Exp((b - r) * T2) * CBND(-z1, y1, -rho) _
         - X1 * Exp(-r * T2) * CBND(-z2, y2, -rho) _
```

```
        + Exp(-r * t1) * X2 * CND(y2)
    End If

End Function


Function CriticalValueOptionsOnOptions(CallPutFlag As String, X1 As Double, _
X2 As Double, T As Double, r As Double, b As Double, v As Double) As Double

    Dim Si As Double, ci As Double, di As Double, epsilon As Double

    Si = X1
    ci = GBlackScholes(CallPutFlag, Si, X1, T, r, b, v)
    di = GDelta(CallPutFlag, Si, X1, T, r, b, v)
    epsilon = 1e-06

    '// Newton-Raphson algorithm
    While Abs(ci - X2) > epsilon
        Si = Si - (ci - X2) / di
        ci = GBlackScholes(CallPutFlag, Si, X1, T, r, b, v)
        di = GDelta(CallPutFlag, Si, X1, T, r, b, v)
    Wend
    CriticalValueOptionsOnOptions = Si

End Function
```

where $CND(\cdot)$ is the cumulative normal distribution function and $CBND(\cdot)$ is the cumulative bivariate normal distribution function described in Chapter 13. Example: *OptionsOnOptions* ("*pc*", 500, 520, 50, 0.25, 0.5, 0.08, 0.05, 0.35) returns a put-on-call price of 21.1965, as in the numerical example above.

4.13.1 Put–Call Parity Compound Options

Shilling (2001) gives the put-call parity between options on options

$$c_{call}(S, X_1, X_2, t_1, T_2, r, b, \sigma) + X_2 e^{-rt_1} = p_{call}(S, X_1, X_2, t_1, T_2, r, b, \sigma) + c_{BSM}(S, X_1, T_2, r, b, \sigma) \qquad (4.32)$$

That is, a call on a call plus the discounted strike price of the compound options is equal in value to a put on a call plus a standard call with strike X_1 and time to maturity T_2. Similar we have

$$c_{put}(S, X_1, X_2, t_1, T_2, r, b, \sigma) + X_2 e^{-rt_1} = p_{put}(S, X_1, X_2, t_1, T_2, r, b, \sigma) + p_{BSM}(S, X_1, T_2, r, b, \sigma). \qquad (4.33)$$

That is, a call on a put plus the discounted strike price of the compound options is equal in value to a put on a put plus a standard put with strike X_1 and time to maturity T_2.

4.13.2 Compound Option Approximation

Bensoussan, Crouhy, and Galai (1995), show how to approximate the value of a call on call option using the Black-Scholes-Merton formula; see also Bensoussan, Crouhy, and Galai (1997). The approximate value is given by

$$c_{call} \approx c_{BSM} N(d_1) - X_2 e^{-rt_1} N(d_2) \tag{4.34}$$

$$d_1 = \frac{\ln(c_{BSM}/X_2) + (b + \hat{\sigma}^2/2)t_1}{\hat{\sigma}\sqrt{t_1}}$$

$$d_2 = d_1 - \sigma\sqrt{t_1},$$

and the volatility of the underlying option can be approximated by

$$\hat{\sigma} \approx \sigma \frac{|\Delta_{BSM}|S}{c_{BSM}},$$

where

$$c_{BSM} = c_{BSM}(S, X_1, T_2, r, b, \sigma) \qquad \Delta_{BSM} = \Delta_{BSM}(S, X_1, T_2, r, b, \sigma)$$

For a put on a call, the approximate value is

$$p_{call} \approx X_2 e^{-rt_1} N(d_2) - c_{BSM} N(d_1)$$

The approximation is quite accurate for at- and in-the-money options. It is less accurate for out-of-the-money options. The formula approximates the volatility of the underlying call option by $\hat{\sigma}$. The formulas would have been exact closed-form solutions if the volatility of the underlying option was constant through time. We know from a simple application of Ito's lemma, however, that the volatility of the underlying option will change through time, even if the volatility of the underlying asset is assumed to be constant. The $\hat{\sigma}$ is thus an approximating average volatility of the underlying option.

Example

Consider the same example as used to illustrate the exact compound option formula: a put-on-call option that gives the option holder the right to sell a call option for 50, three months from today. The strike on the underlying call option is 520, the time to maturity on the call is six months from today, the price on the underlying stock index is 500, the risk-free interest rate is 8%, and the stock index pays dividends at a rate of 3% annually and has a volatility of 35%. $S = 500$, $X_1 = 520$, $X_2 = 50$, $t_1 = 0.25$, $T_2 = 0.5$, $r = 0.08$, $b = 0.08 - 0.03 = 0.05$, $\sigma = 0.35$. First we have to calculate the value of the call option

$c_{BSM}(S, X_1, T_2, r, b, \sigma)$:

$$d_1 = \frac{\ln(500/520) + (0.05 + 0.35^2/2)0.5}{0.35\sqrt{0.5}} = 0.0663$$

$$d_2 = d_1 - \sigma\sqrt{T_2} = 0.0663 - 0.35\sqrt{0.5} = -0.1812$$

$$N(d_1) = N(0.0663) = 0.5264 \qquad N(d_2) = N(-0.1812) = 0.4281$$

$$c = 500e^{(0.05-0.08)0.5}N(d_1) - 520e^{-0.08\times0.5}N(d_2) = 45.4081$$

Next we calculate the volatility of the underlying option:

$$\hat{\sigma} \approx \sigma\frac{|\Delta_{BSM}(S, X_1, T_2, r, b, \sigma)|S}{c_{BSM}(S, X_1, T_2, r, b, \sigma)} = 0.35 \times \frac{e^{(0.05-0.08)0.5}N(d_1) \times 500}{45.4081} = 1.9986$$

Now we can calculate the option on option value:

$$d_1 = \frac{\ln(45.4081/50) + (0.05 + 1.9986^2/2)0.25}{1.9986\sqrt{0.25}} = 0.4158$$

$$d_2 = d_1 - \sigma\sqrt{t_1} = 0.4158 - 1.9986\sqrt{0.25} = -0.5835$$

$$N(-d_1) = N(-0.4158) = 0.3388 \qquad N(-d_2) = N(0.5835) = 0.7202$$

This gives us a put on call value of

$$p_{call} \approx 45.4081 \times N(-d_1) - Xe^{-0.08\times0.25}N(-d_2) = 19.9147$$

The exact value is 21.1965, so in this case, the approximation is not very good. However, this approach at least gives us quite good intuition behind the basics of compound option valuation.

ATM-ATM Approximate Compound Option

In the special case of an at-the-money compound option on an at-the money forward underlying option, the value can be approximated by

$$c_{call} \approx p_{call} \approx S\sigma\sqrt{t_1}(0.08\sigma\sqrt{T_2} + 0.2)$$

or

$$S0.4\sigma\sqrt{t_1}(0.2\sigma\sqrt{T_2} + 0.5) = BS(t_1)_{ATMF} \times \Delta(T_2)_{ATMF}$$

The value is thus approximately equal to Black-Scholes at-the-money forward with time to maturity t_1 ($BS(t_1)_{ATMF}$), multiplied by the Black-Scholes delta for an option at-the-money forward with time to maturity T_2 ($\Delta(T_2)_{ATMF}$). Similarly, for a call on a put or a put on a put, we have

$$c_{put} \approx p_{put} \approx S0.4\sigma\sqrt{t_1}(0.5 - 0.2\sigma\sqrt{T_2}) = BS(t_1)_{ATM} \times (-\Delta(T_2)_{ATMF})$$

4.14 OPTIONS WITH EXTENDIBLE MATURITIES

Valuation of extendible options was introduced by Longstaff (1990). Extendible options can be found embedded in several financial contracts. For example, corporate warrants have frequently given the corporate issuer the right to extend the life of the warrants. Firms involved in leveraged buyouts from time to time issue extendible bonds where maturity can be extended at the firm's discretion. Another example is options on real estate where the holder can extend the expiration by paying an additional fee.

4.14.1 Options That Can Be Extended by the Holder

These are options that can be exercised at their maturity date t_1 but that also allow the holder at that time to extend the life of the option until T_2 by paying an additional premium A to the writer of the option. The strike price of the option can be adjusted from X_1 to X_2 at the time of the extension. The payoff from options that can be extended by the holder at time t_1 ($t_1 < T_2$) is

$$c(S, X_1, X_2, t_1, T_2) = \max[S - X_1; c_{BSM}(S, X_2, T_2 - t_1) - A; 0]$$

$$p(S, X_1, X_2, t_1, T_2) = \max[X_1 - S; p_{BSM}(S, X_2, T_2 - t_1) - A; 0],$$

where $c_{BSM}(S, X_2, T_2 - t_1)$ is the general Black-Scholes-Merton call formula, and $p_{BSM}(S, X_2, T_2 - t_1)$ is the general Black-Scholes-Merton put formula.

Extendible Call

$$\begin{aligned} c = {} & c_{BSM}(S, X_1, t_1) + Se^{(b-r)T_2} M_2(y_1, y_2, -\infty, z_1; \rho) \\ & - X_2 e^{-rT_2} M_2(y_1 - \sigma\sqrt{t_1}, y_2 - \sigma\sqrt{t_1}, -\infty, z_1 - \sigma\sqrt{T_2}; \rho) \\ & - Se^{(b-r)t_1} N_2(y_1, z_2) + X_1 e^{-rt_1} N_2(y_1 - \sigma\sqrt{t_1}, z_2 - \sigma\sqrt{t_1}) \\ & - Ae^{-rt_1} N_2(y_1 - \sigma\sqrt{t_1}, y_2 - \sigma\sqrt{t_1}), \end{aligned} \tag{4.35}$$

where

$$y_1 = \frac{\ln(S/I_2) + (b + \sigma^2/2)t_1}{\sigma\sqrt{t_1}} \qquad y_2 = \frac{\ln(S/I_1) + (b + \sigma^2/2)t_1}{\sigma\sqrt{t_1}}$$

$$z_1 = \frac{\ln(S/X_2) + (b + \sigma^2/2)T_2}{\sigma\sqrt{T_2}} \qquad z_2 = \frac{\ln(S/X_1) + (b + \sigma^2/2)t_1}{\sigma\sqrt{t_1}}$$

$$\rho = \sqrt{t_1/T_2},$$

where I_1 is the critical value of S at time t_1, below which the option is not extended. I_2 is the critical value of S at time t_1, where the option will be exercised rather than extended. The critical values I_1 and I_2 are the respective solutions to

$$c_{BSM}(I_1, X_2, T_2 - t_1) = A \qquad c_{BSM}(I_2, X_2, T_2 - t_1) = I_2 - X_1 + A$$

If $A = 0$ then $I_1 = 0$, and if $A < X_1 - X_2 e^{-r(T_2 - t_1)}$ then $I_2 = \infty$. The call is extended only if $I_1 < S < I_2$. If $S < I_1$ at t_1, the option expires out-of-the-money, and if $S > I_2$ at t_1 it is optimal to exercise the option rather than extend it. The extendible call has several special cases:

- If $I_1 = 0$ and $I_2 = \infty$, the call will always be extended.
- If $I_1 > 0$ and $I_2 = \infty$, the value of the extendible call reduces to a standard call on a call with strike equal to A. The underlying call has strike X_2 and time to maturity $(T_2 - t_1)$.
- If $I_1 \geq X_1$, the extendible call will never be extended.

The probability $M_2(a, b, c, d; \rho)$ and $N_2(a, b)$ can be determined directly from the standard bivariate normal distribution and the standard normal distribution

$$M_2(a, b, c, d; \rho) = M(b, d; \rho) - M(a, d; \rho) - M(b, c; \rho) + M(a, c; \rho)$$

$$N_2(a, b) = N(b) - N(a)$$

Extendible Put

$$\begin{aligned} p = {} & p_{BSM}(S, X_1, t_1) - Se^{(b-r)T_2} M_2(y_1, y_2, -\infty, -z_1; \rho) \\ & + X_2 e^{-rT_2} M_2(y_1 - \sigma\sqrt{t_1}, y_2 - \sigma\sqrt{t_1}, -\infty, -z_1 + \sigma\sqrt{T_2}; \rho) \\ & + Se^{(b-r)t_1} N_2(z_2, y_2) - X_1 e^{-rt_1} N_2(z_2 - \sigma\sqrt{t_1}, y_2 - \sigma\sqrt{t_1}) \\ & - Ae^{-rt_1} N_2(y_1 - \sigma\sqrt{t_1}, y_2 - \sigma\sqrt{t_1}), \end{aligned} \tag{4.36}$$

where the variables I_1 and I_2 are solutions to

$$p_{BSM}(I_1, X_2, T_2 - t_1) = X_1 - I_1 + A, \qquad p_{BSM}(I_2, X_2, T_2 - t_1) = A$$

If $A = 0$, then $I_2 = \infty$. The put is extended only if $I_1 < S < I_2$. The extendible put has several special cases:

- If $A = 0$ and $I_1 = 0$, the put will always be extended.
- If $A > 0$ and $I_1 = 0$, the value of the extendible put reduces to a standard call on a put with strike A. The underlying put has strike X_2 and time to maturity $(T_2 - t_1)$.
- If $I_2 < X_1$ or $I_1 = X_1$, the extendible put will never be extended.

Example

Consider an extendible call with initial time to maturity six months, extendible for an additional three months. The stock price is 100, the initial strike price is 100, the extended strike price is 105, the risk-free interest rate is 8% per year, the volatility is 25% per year, and the extension fee is 1. $S = 100$, $X_1 = 100$, $X_2 = 105$, $t_1 = 0.5$, $T_2 = 0.75$, $r = 0.08$, $b = 0.08$, $\sigma = 0.25$, and $A = 1$.

$$y_1 = \frac{\ln(100/105.7138) + (0.08 + 0.25^2/2)0.5}{0.25\sqrt{0.5}} = 0.0003$$

The critical values I_1 and I_2 can easily be found by the Newton-Raphson algorithm. It gives $I_1 = 86.7406$ and $I_2 = 105.7138$.

$$y_2 = \frac{\ln(100/86.7406) + (0.08 + 0.25^2/2)0.5}{0.25\sqrt{0.5}} = 1.1193$$

$$z_1 = \frac{\ln(100/105) + (0.08 + 0.25^2/2)0.75}{0.25\sqrt{0.75}} = 0.1600$$

$$z_2 = \frac{\ln(100/100) + (0.08 + 0.25^2/2)0.5}{0.25\sqrt{0.5}} = 0.3147$$

$$\rho = \sqrt{0.5/0.75} = 0.8165$$

$$M_2(y_1, y_2, -10, z_1; \rho) = 0.1277$$

$$M_2(y_1 - 0.25\sqrt{0.5}, y_2 - 0.25\sqrt{0.5}, -10, z_1 - 0.25\sqrt{0.75}; \rho) = 0.1181$$

$$N_2(y_1, z_2) = 0.1234, \quad N_2(y_1 - 0.25\sqrt{0.5}, y_2 - 0.25\sqrt{0.5}) = 0.3971$$

$$N_2(y_1 - 0.25\sqrt{0.5}, z_2 - 0.25\sqrt{0.5}) = 0.1249$$

$$c_{BSM}(100, 100, 0.5) = 9.0412$$

$$c = 9.0412 + 100e^{(0.08-0.08)0.75} M_2(y_1, y_2, -\infty, z_1; \rho)$$
$$- 105e^{-0.08\times0.75} M_2(y_1 - \sigma\sqrt{t_1}, y_2 - \sigma\sqrt{t_1}, -\infty, z_1 - \sigma\sqrt{T_2}; \rho)$$
$$+ 100e^{(0.08-0.08)0.5} N_2(y_1, z_2) - 100e^{-0.08\times0.5} N_2(y_1 - \sigma\sqrt{t_1}, z_2 - \sigma\sqrt{t_1})$$
$$- 1e^{-0.08\times0.5} N_2(y_1 - \sigma\sqrt{t_1}, y_2 - \sigma\sqrt{t_1}) = 9.4029$$

4.14.2 Writer-Extendible Options

These options can be exercised at their initial maturity date t_1 but are extended to T_2 if the option is out-of-the-money at t_1. The payoff from a writer-extendible call option at time t_1 ($t_1 < T_2$) is

$$c(S, X_1, X_2, t_1, T_2) = (S - X_1) \quad \text{if} \quad S \geq X_1 \quad \text{else} \quad c_{BSM}(S, X_2, T_2 - t_1),$$

and for a writer-extendible put is

$$p(S, X_1, X_2, t_1, T_2) = (X_1 - S) \quad \text{if} \quad S < X_1 \quad \text{else} \quad p_{BSM}(S, X_2, T_2 - t_1)$$

Writer-Extendible Call

$$c = c_{BSM}(S, X_1, t_1) + Se^{(b-r)T_2} M(z_1, -z_2; -\rho) - X_2 e^{-rT_2} M(z_1 - \sigma\sqrt{T_2}, -z_2 + \sigma\sqrt{t_1}; -\rho) \tag{4.37}$$

Writer-Extendible Put

$$p = p_{BSM}(S, X_1, t_1) + X_2 e^{-rT_2} M(-z_1 + \sigma\sqrt{T_2}, z_2 - \sigma\sqrt{t_1}; -\rho) - Se^{(b-r)T_2} M(-z_1, z_2; -\rho) \tag{4.38}$$

Example
Consider a writer-extendible call on a stock with original time to maturity six months, that will be extended three months if the option is out-of-the-money at t_1. The stock price is 80, and the initial strike price is 90. If the option is extended, the strike price is adjusted to 82. The risk-free interest rate is 10%, and the volatility is 30%. $S = 80$, $X_1 = 90$, $X_2 = 82$, $t_1 = 0.5$, $T_2 = 0.75$, $r = 0.1$, $b = 0.1$, and $\sigma = 0.3$.

$$z_1 = \frac{\ln(80/82) + (0.1 + 0.3^2/2)0.75}{0.3\sqrt{0.75}} = 0.3235$$

$$z_2 = \frac{\ln(80/90) + (0.1 + 0.3^2/2)0.5}{0.3\sqrt{0.5}} = -0.2135$$

$$\rho = \sqrt{0.5/0.75} = 0.8165$$

$$M(z_1, -z_2; -\rho) = 0.2369, \quad M(z_1 - 0.3\sqrt{0.75}, -z_2 + 0.3\sqrt{0.5}; -\rho) = 0.2192$$

$$c_{BSM}(80, 90, 0.5) = 4.5418$$

$$c = 4.5418 + 80e^{(0.1-0.1)0.75} M(z_1, -z_2; -\rho) - 82e^{-0.1\times 0.75} M(z_1 - 0.3\sqrt{0.75}, -z_2 + 0.3\sqrt{0.5}; -\rho) = 6.8238$$

4.15 LOOKBACK OPTIONS

4.15.1 Floating-Strike Lookback Options

A floating-strike lookback call gives the holder of the option the right to buy the underlying security at the lowest price observed, S_{min}, during the option's lifetime. Similarly, a floating-strike lookback put gives the option holder the right to sell the underlying security at the highest price observed, S_{max}, during the option's lifetime. The payoff from a standard floating-strike lookback call option is

$$c(S, S_{min}, T) = \max(S - S_{min}; 0) = S - S_{min},$$

and for a put it is

$$p(S, S_{max}, T) = \max(S_{max} - S; 0) = S_{max} - S$$

Floating-strike lookback options were originally introduced by Goldman, Sosin, and Gatto (1979).[5]

Floating-Strike Lookback Call
If $b \neq 0$ then

$$\begin{aligned} c &= Se^{(b-r)T}N(a_1) - S_{min}e^{-rT}N(a_2) \\ &+ Se^{-rT}\frac{\sigma^2}{2b}\left[\left(\frac{S}{S_{min}}\right)^{-\frac{2b}{\sigma^2}} N\left(-a_1 + \frac{2b}{\sigma}\sqrt{T}\right) - e^{bT}N(-a_1)\right] \end{aligned} \tag{4.39}$$

and if $b = 0$ we have

$$\begin{aligned} c &= Se^{-rT}N(a_1) - S_{min}e^{-rT}N(a_2) \\ &+ Se^{-rT}\sigma\sqrt{T}\,[n(a_1) + a_1(N(a_1) - 1)], \end{aligned} \tag{4.40}$$

where

$$a_1 = \frac{\ln(S/S_{min}) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$a_2 = a_1 - \sigma\sqrt{T}$$

Floating-Strike Lookback Put
If $b \neq 0$ then

$$\begin{aligned} p &= S_{max}e^{-rT}N(-b_2) - Se^{(b-r)T}N(-b_1) \\ &+ Se^{-rT}\frac{\sigma^2}{2b}\left[-\left(\frac{S}{S_{max}}\right)^{-\frac{2b}{\sigma^2}} N\left(b_1 - \frac{2b}{\sigma}\sqrt{T}\right) + e^{bT}N(b_1)\right] \end{aligned} \tag{4.41}$$

and if $b = 0$ we have

$$\begin{aligned} p &= S_{max}e^{-rT}N(-b_2) - Se^{(b-r)T}N(-b_1) \\ &+ Se^{-rT}\sigma\sqrt{T}[n(b_1) + N(b_1)b_1], \end{aligned} \tag{4.42}$$

where

$$b_1 = \frac{\ln(S/S_{max}) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$b_2 = b_1 - \sigma\sqrt{T}$$

[5]See also Garman (1989).

Example

Consider a lookback call option with six months left to expiration. Assume it gives the right to buy the underlying stock index at the lowest price recorded during the life of the option and that the minimum stock index price observed so far is 100, the stock price is 120, the risk-free interest rate is 10%, the dividend yield is 6%, and the volatility is 30%. $S = 120$, $S_{min} = 100$, $T = 0.5$, $r = 0.1$, $b = 0.1 - 0.06 = 0.04$, $\sigma = 0.3$.

$$a_1 = \frac{\ln(120/100) + (0.04 + 0.3^2/2)0.5}{0.3\sqrt{0.5}} = 1.0598$$

$$a_2 = a_1 - 0.3\sqrt{0.5} = 0.8477$$

$$N(a_1) = N(1.0598) = 0.8554 \qquad N(-a_1) = N(-1.0598) = 0.1446$$

$$N(a_2) = N(0.8477) = 0.8017 \qquad N\left(-a_1 + \frac{2 \times 0.04}{0.3}\sqrt{0.5}\right) = 0.1918$$

$$c = 120e^{(0.04-0.1)0.5}N(a_1) - 100e^{-0.1\times 0.5}N(a_2) + 120e^{-0.1\times 0.5}\frac{0.3^2}{2 \times 0.04}$$
$$\times \left[\left(\frac{120}{100}\right)^{-2\times 0.04/0.3^2} N\left(-a_1 + \frac{2 \times 0.04}{0.3}\sqrt{0.5}\right) - e^{0.04\times 0.5}N(-a_1)\right] = 25.3533$$

Application

Floating-strike lookback options can be used to construct what is sometimes marketed as a range or hi–low option. A range option guarantees a payout equal to the observed range of the underlying asset, through the life of the option. This is actually just a lookback straddle: a long lookback call plus a long lookback put with the same time to maturity.

4.15.2 Fixed-Strike Lookback Options

In a fixed-strike lookback call, the strike is fixed in advance. At expiration, the option pays out the maximum of the difference between the highest observed price during the option's lifetime, S_{max}, and the strike X, and 0. Similarly, a put at expiration pays out the maximum of the difference between the fixed-strike X and the minimum observed price, S_{min}, and 0. Fixed-strike lookback options can be priced using the Conze and Viswanathan (1991) formula.

Fixed-Strike Lookback Call

$$c = Se^{(b-r)T}N(d_1) - Xe^{-rT}N(d_2)$$
$$+ Se^{-rT}\frac{\sigma^2}{2b}\left[-\left(\frac{S}{X}\right)^{-\frac{2b}{\sigma^2}} N\left(d_1 - \frac{2b}{\sigma}\sqrt{T}\right) + e^{bT}N(d_1)\right], \qquad (4.43)$$

where

$$d_1 = \frac{\ln(S/X) + (b + \sigma^2/2)T}{\sigma\sqrt{T}} \qquad d_2 = d_1 - \sigma\sqrt{T}$$

When $X \le S_{max}$

$$c = e^{-rT}(S_{max} - X) + Se^{(b-r)T}N(e_1) - S_{max}e^{-rT}N(e_2)$$
$$+ Se^{-rT}\frac{\sigma^2}{2b}\left[-\left(\frac{S}{S_{max}}\right)^{-\frac{2b}{\sigma^2}} N\left(e_1 - \frac{2b}{\sigma}\sqrt{T}\right) + e^{bT}N(e_1)\right],$$

where

$$e_1 = \frac{\ln(S/S_{max}) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}, \qquad e_2 = e_1 - \sigma\sqrt{T}$$

Fixed-Strike Lookback Put

$$p = Xe^{-rT}N(-d_2) - Se^{(b-r)T}N(-d_1)$$
$$+ Se^{-rT}\frac{\sigma^2}{2b}\left[\left(\frac{S}{X}\right)^{-\frac{2b}{\sigma^2}} N\left(-d_1 + \frac{2b}{\sigma}\sqrt{T}\right) - e^{bT}N(-d_1)\right] \qquad (4.44)$$

When $X \ge S_{min}$

$$p = e^{-rT}(X - S_{min}) - Se^{(b-r)T}N(-f_1) + S_{min}e^{-rT}N(-f_2)$$
$$+ Se^{-rT}\frac{\sigma^2}{2b}\left[\left(\frac{S}{S_{min}}\right)^{-\frac{2b}{\sigma^2}} N\left(-f_1 + \frac{2b}{\sigma}\sqrt{T}\right) - e^{bT}N(-f_1)\right],$$

where

$$f_1 = \frac{\ln(S/S_{min}) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}, \qquad f_2 = f_1 - \sigma\sqrt{T}$$

Table 4-8 shows values for fixed-strike lookback call and put options, for different values of time to maturity T, strike price X, and volatility σ.

4.15.3 Partial-Time Floating-Strike Lookback Options

In the partial-time floating-strike lookback options, the lookback period is at the beginning of the option's lifetime. Time to expiration is T_2, and time to the end of the lookback period is t_1 ($t_1 \le T_2$). Except for the partial lookback period, the partial-time floating-strike lookback option is similar to a standard floating-strike lookback option. However, a partial lookback option must naturally be cheaper than a similar standard floating-strike lookback option. Heynen and Kat (1994c) have developed formulas for pricing these options.

TABLE 4-8

Fixed-Strike Lookback Option Values

($S = S_{min} = S_{max} = 100, r = 0.1, b = 0.1$)

		Call			Put		
	X	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$
$T = 0.5$	95	13.2687	18.9263	24.9857	0.6899	4.4448	8.9213
	100	8.5126	14.1702	20.2296	3.3917	8.3177	13.1579
	105	4.3908	9.8905	15.8512	8.1478	13.0739	17.9140
$T = 1$	95	18.3241	26.0731	34.7116	1.0534	6.2813	12.2376
	100	13.8000	21.5489	30.1874	3.8079	10.1294	16.3889
	105	9.5445	17.2965	25.9002	8.3321	14.6536	20.9130

Partial-Time Floating-Strike Lookback Call

$$
\begin{aligned}
c = {} & Se^{(b-r)T_2}N(d_1 - g_1) - \lambda S_{min}e^{-rT_2}N(d_2 - g_1) \\
& + \lambda Se^{-rT_2}\frac{\sigma^2}{2b}\Bigg[\left(\frac{S}{S_{min}}\right)^{-\frac{2b}{\sigma^2}} M\Bigg(-f_1 + \frac{2b\sqrt{t_1}}{\sigma} \\
& - d_1 + \frac{2b\sqrt{T_2}}{\sigma} - g_1; \sqrt{t_1/T_2}\Bigg) \\
& - e^{bT_2}\lambda^{\frac{2b}{\sigma^2}} M(-d_1 - g_1, e_1 + g_2; -\sqrt{1 - t_1/T_2})\Bigg] \\
& + Se^{(b-r)T_2}M(-d_1 + g_1, e_1 - g_2; -\sqrt{1 - t_1/T_2}) \\
& + \lambda S_{min}e^{-rT_2}M(-f_2, d_2 - g_1; -\sqrt{t_1/T_2}) \\
& - e^{-b(T_2 - t_1)}\left(1 + \frac{\sigma^2}{2b}\right)\lambda Se^{(b-r)T_2}N(e_2 - g_2)N(-f_1)
\end{aligned}
\tag{4.45}
$$

The factor λ enables the creation of so-called "fractional" lookback options where the strike is fixed at some percentage above or below the actual extreme, $\lambda \geq 1$ for calls and $0 < \lambda \leq 1$ for puts.

$$d_1 = \frac{\ln(S/M_0) + (b + \sigma^2/2)T_2}{\sigma\sqrt{T_2}} \qquad d_2 = d_1 - \sigma\sqrt{T_2}$$

$$e_1 = \frac{(b + \sigma^2/2)(T_2 - t_1)}{\sigma\sqrt{T_2 - t_1}} \qquad e_2 = e_1 - \sigma\sqrt{T_2 - t_1}$$

$$f_1 = \frac{\ln(S/M_0) + (b + \sigma^2/2)t_1}{\sigma\sqrt{t_1}} \qquad f_2 = f_1 - \sigma\sqrt{t_1}$$

$$g_1 = \frac{\ln(\lambda)}{\sigma\sqrt{T_2}} \qquad g_2 = \frac{\ln(\lambda)}{\sigma\sqrt{T_2 - t_1}}$$

where

$$M_0 = \begin{cases} S_{min} & \text{if call} \\ S_{max} & \text{if put} \end{cases}$$

Partial-Time Floating-Strike Lookback Put

$$\begin{aligned} p = {} & \lambda S_{max} e^{-rT_2} N(-d_2 + g_1) - S e^{(b-r)T_2} N(-d_1 + g_1) \\ & + \lambda S e^{-rT_2} \frac{\sigma^2}{2b} \left[-\left(\frac{S}{S_{max}} \right)^{-\frac{2b}{\sigma^2}} M \left(f_1 - \frac{2b\sqrt{t_1}}{\sigma} \right. \right. \\ & \left. d_1 - \frac{2b\sqrt{T_2}}{\sigma} + g_1; \sqrt{t_1/T_2} \right) \\ & \left. + e^{bT_2} \lambda^{\frac{2b}{\sigma^2}} M(d_1 + g_1, -e_1 - g_2; -\sqrt{1 - t_1/T_2}) \right] \\ & - S e^{(b-r)T_2} M(d_1 - g_1, -e_1 + g_2; -\sqrt{1 - t_1/T_2}) \\ & - \lambda S_{max} e^{-rT_2} M(f_2, -d_2 + g_1; -\sqrt{t_1/T_2}) \\ & + e^{-b(T_2 - t_1)} \left(1 + \frac{\sigma^2}{2b} \right) \lambda S e^{(b-r)T_2} N(-e_2 + g_2) N(f_1) \end{aligned} \tag{4.46}$$

Table 4-9 shows values for partial-time floating-strike lookback call and put options. Different input parameters are used for the volatility σ, the asset price S, and the time to the end of the lookback period t_1.

TABLE 4-9

Partial-Time Floating-Strike Lookback Option Values

$(T_2 = 1, r = 0.06, b = 0.06, \lambda = 1)$

	$S = S_{min} = S_{max} = 90$			$S = S_{min} = S_{max} = 110$		
	$t_1 = 0.25$	$t_1 = 0.5$	$t_1 = 0.75$	$t_1 = 0.25$	$t_1 = 0.5$	$t_1 = 0.75$
call $\sigma = 0.1$	8.6524	9.2128	9.5567	10.5751	11.2601	11.6804
call $\sigma = 0.2$	13.3402	14.5121	15.3140	16.3047	17.7370	18.7171
call $\sigma = 0.3$	17.9831	19.6618	20.8493	21.9793	24.0311	25.4825
put $\sigma = 0.1$	2.7189	3.4639	4.1912	3.3231	4.2336	5.1226
put $\sigma = 0.2$	7.9153	9.5825	11.0362	9.6743	11.7119	13.4887
put $\sigma = 0.3$	13.4719	16.1495	18.4071	16.4657	19.7383	22.4976

4.15.4 Partial-Time Fixed-Strike Lookback Options

For the option described here the lookback period starts at a predetermined date t_1 after the option contract is initiated. The partial-time fixed-strike lookback call pays off the maximum of the highest observed price of the underlying asset in the lookback period, S_{max}, in excess of the strike price X, and 0. The put pays off the maximum of the fixed-strike price X minus the minimum observed asset price in the lookback period $(T_2 - t_1)$, S_{min}, and 0. This option is naturally cheaper than a similar standard fixed-strike lookback option. Heynen and Kat (1994c) have published closed-form solutions for these types of options.

Partial-Time Fixed-Strike Lookback Call

$$
\begin{aligned}
c &= Se^{(b-r)T_2} N(d_1) - Xe^{-rT_2} N(d_2) \\
&+ Se^{-rT_2} \frac{\sigma^2}{2b} \Bigg[- \left(\frac{S}{X}\right)^{-\frac{2b}{\sigma^2}} \\
&\times M\left(d_1 - \frac{2b\sqrt{T_2}}{\sigma} - f_1 + \frac{2b\sqrt{t_1}}{\sigma}, -\sqrt{t_1/T_2}\right) \\
&+ e^{bT_2} M(e_1, d_1; \sqrt{1 - t_1/T_2}) \Bigg] \\
&- Se^{(b-r)T_2} M(-e_1, d_1; -\sqrt{1 - t_1/T_2}) \\
&- Xe^{-rT_2} M(f_2, -d_2; -\sqrt{t_1/T_2}) \\
&+ e^{-b(T_2 - t_1)} \left(1 - \frac{\sigma^2}{2b}\right) Se^{(b-r)T_2} N(f_1) N(-e_2),
\end{aligned} \tag{4.47}
$$

where d_1, e_1, f_1 are defined under the floating-strike lookback options.

Partial-Time Fixed-Strike Lookback Put

$$
\begin{aligned}
p &= Xe^{-rT_2} N(-d_2) - Se^{(b-r)T_2} N(-d_1) + Se^{-rT_2} \frac{\sigma^2}{2b} \\
&\times \Bigg[\left(\frac{S}{X}\right)^{-\frac{2b}{\sigma^2}} M\left(-d_1 + \frac{2b\sqrt{T_2}}{\sigma}, f_1 - \frac{2b\sqrt{t_1}}{\sigma}; -\sqrt{t_1/T_2}\right) \\
&- e^{bT_2} M(-e_1, -d_1; \sqrt{1 - t_1/T_2}) \Bigg]
\end{aligned}
$$

TABLE 4-10

Partial-Time Fixed-Strike Lookback Option Values

($S = 100, T_2 = 1, r = 0.06, b = 0.06$)

	$X = 90$			$X = 110$		
	$t_1 = 0.25$	$t_1 = 0.5$	$t_1 = 0.75$	$t_1 = 0.25$	$t_1 = 0.5$	$t_1 = 0.75$
call $\sigma = 0.1$	20.2845	19.6239	18.6244	4.0432	3.9580	3.7015
call $\sigma = 0.2$	27.5385	25.8126	23.4957	11.4895	10.8995	9.8244
call $\sigma = 0.3$	35.4578	32.7172	29.1473	19.7250	18.4025	16.2976
put $\sigma = 0.1$	0.4973	0.4632	0.3863	12.6978	10.9492	9.1555
put $\sigma = 0.2$	4.5863	4.1925	3.5831	19.0255	16.9433	14.6505
put $\sigma = 0.3$	9.9348	9.1111	7.9267	25.2112	22.8217	20.0566

$$+ Se^{(b-r)T_2} M(e_1, -d_1; -\sqrt{1 - t_1/T_2})$$
$$+ Xe^{-rT_2} M(-f_2, d_2; -\sqrt{t_1/T_2})$$
$$- e^{-b(T_2-t_1)} \left(1 - \frac{\sigma^2}{2b}\right) Se^{(b-r)T_2} N(-f_1) N(e_2) \qquad (4.48)$$

Table 4-10 shows examples of partial-time fixed-strike lookback call and put option values. The input parameters vary as in Table 4–9.

4.15.5 Extreme-Spread Options

The time to maturity of an extreme-spread option is divided into two periods: one period starting today and ending at time t_1, and another period starting at t_1 and ending at the maturity of the option T_2. The payoff at maturity of a call (put) equals the positive part of the difference between the maximum (minimum) value of the underlying asset of the second (first) period, S_{max}, and the maximum (minimum) value of the underlying asset of the first (second) period. Likewise, the payoff at maturity of a reverse extreme spread call (put) equals the positive part of the difference between the minimum (maximum) of the underlying asset of the second (first) period, S_{min}, and the minimum (maximum) value of the underlying asset of the second (first) period. Formulas for valuation of these types of options were introduced by Bermin (1996b).

Extreme-Spread Option Values

$$
\begin{aligned}
w = \eta \Bigg[& Se^{(b-r)T_2}\left(1+\frac{\sigma^2}{2b}\right) N_\eta\left(\frac{-m+\mu_2 T_2}{\sigma\sqrt{T_2}}\right) - e^{-b(T_2-t_1)} Se^{(b-r)T_2} \\
& \times \left(1+\frac{\sigma^2}{2b}\right) N_\eta\left(\frac{-m+\mu_2 t_1}{\sigma\sqrt{t_1}}\right) + e^{-rT_2} M_0 N_\eta\left(\frac{m-\mu_1 T_2}{\sigma\sqrt{T_2}}\right) \\
& - e^{-rT_2} M_0 \frac{\sigma^2}{2b} e^{\frac{2\mu_1 m}{\sigma^2}} N_\eta\left(\frac{-m-\mu_1 T_2}{\sigma\sqrt{T_2}}\right) - e^{-rT_2} M_0 N_\eta\left(\frac{m-\mu_1 t_1}{\sigma\sqrt{t_1}}\right) \\
& + e^{-rT_2} M_0 \frac{\sigma^2}{2b} e^{\frac{2\mu_1 m}{\sigma^2}} N_\eta\left(\frac{-m-\mu_1 t_1}{\sigma\sqrt{t_1}}\right) \Bigg],
\end{aligned} \tag{4.49}
$$

where $N_\eta(x) = N(\eta x)$, and

$$
m = \ln(M_0/S) \qquad \mu_1 = b - \sigma^2/2 \qquad \mu_2 = b + \sigma^2/2
$$

$$
\eta = \begin{cases} 1 & \text{if call} \\ -1 & \text{if put} \end{cases} \qquad \phi = \begin{cases} 1 & \text{if extreme spread} \\ -1 & \text{if reverse extreme spread} \end{cases}
$$

$$
M_0 = \begin{cases} S_{max} & \text{if } \phi\eta = 1 \\ S_{min} & \text{if } \phi\eta = -1 \end{cases}
$$

where S_{min} is the observed minimum, and S_{max} is the observed maximum.

Reverse Extreme-Spread Option Values

$$
\begin{aligned}
w = -\eta \Bigg[& Se^{(b-r)T_2}\left(1+\frac{\sigma^2}{2b}\right) N_\eta\left(\frac{m-\mu_2 T_2}{\sigma\sqrt{T_2}}\right) \\
& + e^{-rT_2} M_0 N_\eta\left(\frac{-m+\mu_1 T_2}{\sigma\sqrt{T_2}}\right) \\
& - e^{-rT_2} M_0 \frac{\sigma^2}{2b} e^{\frac{2\mu_1 m}{\sigma^2}} N_\eta\left(\frac{m+\mu_1 T_2}{\sigma\sqrt{T_2}}\right) \\
& - Se^{(b-r)T_2}\left(1+\frac{\sigma^2}{2b}\right) N_\eta\left(\frac{-\mu_2(T_2-t_1)}{\sigma\sqrt{T_2-t_1}}\right) \\
& - e^{-b(T_2-t_1)} Se^{(b-r)T_2}\left(1-\frac{\sigma^2}{2b}\right) N_\eta\left(\frac{\mu_1(T_2-t_1)}{\sigma\sqrt{T_2-t_1}}\right) \Bigg]
\end{aligned} \tag{4.50}
$$

Table 4-11 shows values for extreme and reverse extreme call options for a range of input parameter values.

TABLE 4-11

Extreme and Reverse Extreme Call Option Values

$(S = 100, T_2 = 1, r = b = 0.1)$

		Initial time period t_1			
		0.00	0.25	0.5	0.75
	S_{max}	Extreme-Spread Values			
$\sigma = 0.15$	100	17.5212	10.6618	6.7967	3.3218
	110	9.6924	8.4878	5.8519	2.9676
	120	4.6135	4.5235	3.6613	2.0566
$\sigma = 0.3$	100	30.1874	17.4998	10.9444	5.2735
	110	22.0828	16.3674	10.4668	5.0942
	120	15.7847	13.5892	9.2051	4.6071
	S_{min}	Reverse Extreme-Spread Values			
$\sigma = 0.15$	100	0.0000	2.7046	5.7250	9.3347
	90	3.6267	6.3314	9.3517	12.9615
	80	11.3474	14.0521	17.0724	20.6821
$\sigma = 0.3$	100	0.0000	3.6120	7.8702	13.3404
	90	1.4769	5.0890	9.3471	14.8173
	80	5.7133	9.3253	13.5835	19.0537

4.16 MIRROR OPTIONS

Mirror options introduced by Manzano (2001) has the same payoff at maturity as a standard option, but in addition the owner has the right to mirror the future path of the underlying asset an undetermined number of times in the option's lifetime. To understand the mirroring, first assume the case of one mirror path. By mirroring the underlying asset S_t at a given time t_m, we will get a a mirror path S^* that is defined as $S_t^* = S_{t_m}^2/S_t$, where S_{t_m} is the asset price at the mirror time and S_t is the asset price thereafter. With discrete monitored prices, that would be $S_t^* = S_{t-1}^2/S_t$. After a second mirroring, the path would be just as for S_t but with a multiplicative factor. For several consecutive mirroring times $t_1 \leq t_2 \leq \cdots \leq t_m$, we will have

$$S_t^* = \begin{cases} S_{t_1}^2 S_{t_2}^{-2} \cdots S_{t_{2n}}^{-2} S_t & t_0 \leq t_1 \leq \cdots \leq t_{2n} \leq t \\ S_{t_1}^2 S_{t_2}^{-2} \cdots S_{t_{2n}}^2 S_t^{-1} & t_0 \leq t_1 \leq \cdots \leq t_{2n+1} \leq t \end{cases}$$

In general for an even number of mirrorings, the mirror path equals the path for S_t but with a multiplicative factor dependent on the historical value of the asset price at the mirroring times selected by the option holder. For an odd number of mirrorings, we also have a path-dependent factor, but now multiplying with the inverse value of the underlying S^{-1}.

The owner of the mirror option is not allowed to mirror the past history. Only future, nonpredictable history is therefore affected by the mirroring. Moreover, the value of a short position in a mirror option is not equivalent to the value of selling a mirror option, since only the owner of the option has the right to make the mirror decision. The value of a mirror call is given by

$$c = e^{-rT}[FN(d_1) - XN(d_2)], \tag{4.51}$$

where

$$d_1 = \frac{\ln(S/X) + T\sigma^2/2}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

$$F = Se^{(\sigma^2/2 \pm |b - \sigma^2/2|)T},$$

where $\pm$ is equal to plus for a long option and minus for a short option. For a put we have

$$p = e^{-rT}[XN(-d_2) - FN(-d_1)] \tag{4.52}$$

Table 4-12 shows values for mirror options for different choices of strike price X and volatility σ.

TABLE 4-12

Value of Mirror Options

($S = 100, T = 1, r = 0.1, b = 0$)

X	$\sigma = 2$	$\sigma = 3$	$\sigma = 4$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$
	Long Call Options			Long Put Options		
90	15.1251	21.7207	30.3377	3.2476	6.3455	9.5653
95	12.0145	18.7393	27.3809	4.9943	8.4092	11.8360
100	9.3471	16.0579	24.6489	7.2075	10.7889	14.3434
105	7.1263	13.6733	22.1377	9.8678	13.4655	17.0719
110	5.3293	11.5746	19.8406	12.9319	16.4147	20.0043
	Short Call Options			Short Put Options		
90	−12.2959	−15.3939	−18.6137	−2.3841	−4.1511	−5.5894
95	−9.5185	−12.9334	−16.3602	−3.7976	−5.6939	−7.1568
100	−7.2075	−10.7889	−14.3434	−5.6544	−7.5366	−8.9490
105	−5.3436	−8.9413	−12.5477	−7.9578	−9.6762	−10.9620
110	−3.8836	−7.3663	−10.9559	−10.6849	−12.1017	−13.1891

4.17 BARRIER OPTIONS

Barrier options on stocks are known to have been traded in the OTC market since 1967. Barrier options have become extremely popular and are certainly the most popular class of exotic options. The Chicago Board Option Exchange and the American Option Exchange now list up-and-out call options and down-and-out put options on stock indexes. Several barrier options are traded actively in the OTC market: currency, interest rate, and commodity options.

4.17.1 Standard Barrier Options

Merton (1973) and Reiner and Rubinstein (1991a) have developed formulas for pricing standard barrier options.[6] The different formulas use a common set of factors:

$$A = \phi S e^{(b-r)T} N(\phi x_1) - \phi X e^{-rT} N(\phi x_1 - \phi\sigma\sqrt{T})$$

$$B = \phi S e^{(b-r)T} N(\phi x_2) - \phi X e^{-rT} N(\phi x_2 - \phi\sigma\sqrt{T})$$

$$C = \phi S e^{(b-r)T} (H/S)^{2(\mu+1)} N(\eta y_1) - \phi X e^{-rT} (H/S)^{2\mu} N(\eta y_1 - \eta\sigma\sqrt{T})$$

$$D = \phi S e^{(b-r)T} (H/S)^{2(\mu+1)} N(\eta y_2) - \phi X e^{-rT} (H/S)^{2\mu} N(\eta y_2 - \eta\sigma\sqrt{T})$$

$$E = K e^{-rT} [N(\eta x_2 - \eta\sigma\sqrt{T}) - (H/S)^{2\mu} N(\eta y_2 - \eta\sigma\sqrt{T})]$$

$$F = K[(H/S)^{\mu+\lambda} N(\eta z) + (H/S)^{\mu-\lambda} N(\eta z - 2\eta\lambda\sigma\sqrt{T})]$$

where

$$x_1 = \frac{\ln(S/X)}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T} \qquad x_2 = \frac{\ln(S/H)}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T}$$

$$y_1 = \frac{\ln(H^2/(SX))}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T} \qquad y_2 = \frac{\ln(H/S)}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T}$$

$$z = \frac{\ln(H/S)}{\sigma\sqrt{T}} + \lambda\sigma\sqrt{T} \qquad \mu = \frac{b - \sigma^2/2}{\sigma^2} \qquad \lambda = \sqrt{\mu^2 + \frac{2r}{\sigma^2}}$$

"In" Barriers

In options are paid for today but first come into existence if the asset price S hits the barrier H before expiration. It is possible to include a prespecified cash rebate K, which is paid out at option expiration if the option has not been knocked in during its lifetime.

[6]See also Rich (1994).

Down-and-in call $S > H$
Payoff: $\max(S - X; 0)$ if $S \leq H$ before T else K at expiration.

$c_{di(X>H)} = C + E \qquad \eta = 1,\ \phi = 1$

$c_{di(X<H)} = A - B + D + E \qquad \eta = 1,\ \phi = 1$

Up-and-in call $S < H$
Payoff: $\max(S - X; 0)$ if $S \geq H$ before T else K at expiration.

$c_{ui(X>H)} = A + E \qquad \eta = -1,\ \phi = 1$

$c_{ui(X<H)} = B - C + D + E \qquad \eta = -1,\ \phi = 1$

Down-and-in put $S > H$
Payoff: $\max(X - S; 0)$ if $S \leq H$ before T else K at expiration.

$p_{di(X>H)} = B - C + D + E \qquad \eta = 1,\ \phi = -1$

$p_{di(X<H)} = A + E \qquad \eta = 1,\ \phi = -1$

Up-and-in put $S < H$
Payoff: $\max(X - S; 0)$ if $S \geq H$ before T else K at expiration.

$p_{ui(X>H)} = A - B + D + E \qquad \eta = -1,\ \phi = -1$

$p_{ui(X<H)} = C + E \qquad \eta = -1,\ \phi = -1$

"Out" Barriers

Out options are similar to standard options except that the option becomes worthless if the asset price S hits the barrier before expiration. It is possible to include a prespecified cash rebate K, which is paid out if the option is knocked out before expiration.

Down-and-out call $S > H$
Payoff: $\max(S - X; 0)$ if $S > H$ before T else K at hit.

$c_{do(X>H)} = A - C + F \qquad \eta = 1,\ \phi = 1$

$c_{do(X<H)} = B - D + F \qquad \eta = 1,\ \phi = 1$

Up-and-out call $S < H$
Payoff: $\max(S - X; 0)$ if $S < H$ before T else K at hit.

$c_{uo(X>H)} = F \qquad \eta = -1,\ \phi = 1$

$c_{uo(X<H)} = A - B + C - D + F \qquad \eta = -1,\ \phi = 1$

Down-and-out put $S > H$
Payoff: $\max(X - S; 0)$ if $S > H$ before T else K at hit.

$p_{do(X>H)} = A - B + C - D + F \qquad \eta = 1,\ \phi = -1$

$p_{do(X<H)} = F \qquad \eta = 1,\ \phi = -1$

Up-and-out put $S < H$
Payoff: $\max(X - S; 0)$ if $S < H$ before T else K at hit.

$p_{uo(X>H)} = B - D + F \qquad \eta = -1,\ \phi = -1$

$p_{uo(X<H)} = A - C + F \qquad \eta = -1,\ \phi = -1$

TABLE 4-13

Value of Standard Barrier Options

($S = 100$, $K = 3$, $T = 0.5$, $r = 0.08$, $b = 0.04$)

Type	X	H	$\sigma = 0.25$	$\sigma = 0.3$	Type	X	H	$\sigma = 0.25$	$\sigma = 0.3$
c_{do}	90	95	9.0246	8.8334	p_{do}	90	95	2.2798	2.4170
c_{do}	100	95	6.7924	7.0285	p_{do}	100	95	2.2947	2.4258
c_{do}	110	95	4.8759	5.4137	p_{do}	110	95	2.6252	2.6246
c_{do}	90	100	3.0000	3.0000	p_{do}	90	100	3.0000	3.0000
c_{do}	100	100	3.0000	3.0000	p_{do}	100	100	3.0000	3.0000
c_{do}	110	100	3.0000	3.0000	p_{do}	110	100	3.0000	3.0000
c_{uo}	90	105	2.6789	2.6341	p_{uo}	90	105	3.7760	4.2293
c_{uo}	100	105	2.3580	2.4389	p_{uo}	100	105	5.4932	5.8032
c_{uo}	110	105	2.3453	2.4315	p_{uo}	110	105	7.5187	7.5649
c_{di}	90	95	7.7627	9.0093	p_{di}	90	95	2.9586	3.8769
c_{di}	100	95	4.0109	5.1370	p_{di}	100	95	6.5677	7.7989
c_{di}	110	95	2.0576	2.8517	p_{di}	110	95	11.9752	13.3078
c_{di}	90	100	13.8333	14.8816	p_{di}	90	100	2.2845	3.3328
c_{di}	100	100	7.8494	9.2045	p_{di}	100	100	5.9085	7.2636
c_{di}	110	100	3.9795	5.3043	p_{di}	110	100	11.6465	12.9713
c_{ui}	90	105	14.1112	15.2098	p_{ui}	90	105	1.4653	2.0658
c_{ui}	100	105	8.4482	9.7278	p_{ui}	100	105	3.3721	4.4226
c_{ui}	110	105	4.5910	5.8350	p_{ui}	110	105	7.0846	8.3686

Table 4-13 shows values for standard barrier options. Values are tabulated with different values for the strike price X, barrier H, and volatility σ.

4.17.2 Standard American Barrier Options

Haug (2001a) utilizes the reflection principle to give closed-form solutions for an American call down-and-in option when the barrier H is below the strike price X. When $H \leq X$, the value is given by

$$C_{di}(S, X, H, T, r, b, \sigma) = \left(\frac{S}{H}\right)^{1-\frac{2b}{\sigma^2}} C\left(\frac{H^2}{S}, X, T, r, b, \sigma\right), \tag{4.53}$$

where $C(\cdot)$ is the value of a plain vanilla American call option. For instance, the Bjerksund-Stensland approximation can be used to value the American call, or any other valuation method for American options (tree models, finite difference). In the case of an American put, Haug (2001c) gives the solution, when $H \geq X$, as

$$P_{ui}(S, X, H, T, r, b, \sigma) = \left(\frac{S}{H}\right)^{1-\frac{2b}{\sigma^2}} P\left(\frac{H^2}{S}, X, T, r, b, \sigma\right) \tag{4.54}$$

Dai and Kwok (2004) generalize this result for a call when $H \leq \max\left(X, \frac{r}{r-b}X\right)$:

$$C_{di}(S, X, H, T, r, b, \sigma) = \left(\frac{S}{H}\right)^{1-\frac{2b}{\sigma^2}} \left[C\left(\frac{H^2}{S}, X, T, r, b, \sigma\right) - c_{BSM}\left(\frac{H^2}{S}, X, T, r, b, \sigma\right)\right] + c_{di}(S, X, H, T, r, b, \sigma) \quad (4.55)$$

The American down-and-in call can thus be decomposed into simpler options—in this case, plain vanilla American $C(\cdot)$ and European calls $c_{BSM}(\cdot)$ plus a European standard barrier option $c_{di}(\cdot)$. For a put when $H \geq \min\left(X, \frac{r}{r-b}X\right)$, we have

$$P_{ui}(S, X, H, T, r, b, \sigma) = \left(\frac{S}{H}\right)^{1-\frac{2b}{\sigma^2}} \left[P\left(\frac{H^2}{S}, X, T, r, b, \sigma\right) - p_{BSM}\left(\frac{H^2}{S}, X, T, r, b, \sigma\right)\right] + p_{ui}(S, X, H, T, b, r, \sigma) \quad (4.56)$$

Table 4-14 shows values for American down-and-in calls, for different values of barrier price H, and time to maturity T.

In-Out Parity for American Barrier Options

We have already presented the in-out barrier parity for European barrier options. For example, the sum of a European call and a European knock-in call with identical strikes is equal to a European knock-out call. As mentioned by Haug (2001a) and discussed in more detail by Dai and Kwok (2004), the in-out parity will, in general, not hold for American barrier options.

TABLE 4-14

Value of Knock-in American Call Options Using Bjerksund-Stensland (2002) Approximation

($S = 100.5, X = 100, r = 0.1, b = 0.02, \sigma = 0.3$)

H	$T = 0.25$	$T = 0.5$	$T = 0.75$	$T = 1$
95	2.1023	4.1636	5.8274	7.2314
99	4.8961	7.3529	9.1900	10.6908
100	5.8488	8.3314	10.1777	11.6823

4.17.3 Double-Barrier Options

A double-barrier option is knocked either in or out if the underlying price touches the lower boundary L or the upper boundary U prior to expiration. The formulas below pertain only to double knock-out options. The price of a double knock-in call is equal to the portfolio of a long standard call and a short double knock-out call, with identical strikes and time to expiration. Similarly, a double knock-in put is equal to a long standard put and a short double knock-out put. Double-barrier options can be priced using the Ikeda and Kuintomo (1992) formula.[7]

Call Up-and-Out-Down-and-Out

Payoff: $c(S, U, L, T) = \max(S - X; 0)$ if $L < S < U$ before T else 0.

$$
\begin{aligned}
c = {} & Se^{(b-r)T} \sum_{n=-\infty}^{\infty} \left\{ \left(\frac{U^n}{L^n}\right)^{\mu_1} \left(\frac{L}{S}\right)^{\mu_2} [N(d_1) - N(d_2)] \right. \\
& \left. - \left(\frac{L^{n+1}}{U^n S}\right)^{\mu_3} [N(d_3) - N(d_4)] \right\} \\
& - Xe^{-rT} \sum_{n=-\infty}^{\infty} \left\{ \left(\frac{U^n}{L^n}\right)^{\mu_1 - 2} \left(\frac{L}{S}\right)^{\mu_2} \right. \\
& \times [N(d_1 - \sigma\sqrt{T}) - N(d_2 - \sigma\sqrt{T})] \\
& \left. - \left(\frac{L^{n+1}}{UnS}\right)^{\mu_3 - 2} [N(d_3 - \sigma\sqrt{T}) - N(d_4 - \sigma\sqrt{T})] \right\}, \qquad (4.57)
\end{aligned}
$$

where

$$
d_1 = \frac{\ln(SU^{2n}/(XL^{2n})) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}
$$

$$
d_2 = \frac{\ln(SU^{2n}/(FL^{2n})) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}
$$

$$
d_3 = \frac{\ln(L^{2n+2}/(XSU^{2n})) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}
$$

$$
d_4 = \frac{\ln(L^{2n+2}/(FSU^{2n})) + (b + \sigma^2/2)T}{\sigma\sqrt{T}},
$$

[7] For the valuation of double-barrier options, see also Bhagavatula and Carr (1995) and Geman and Yor (1996).

$$\mu_1 = \frac{2[b - \delta_2 - n(\delta_1 - \delta_2)]}{\sigma^2} + 1 \qquad \mu_2 = 2n\frac{(\delta_1 - \delta_2)}{\sigma^2}$$

$$\mu_3 = \frac{2[b - \delta_2 + n(\delta_1 - \delta_2)]}{\sigma^2} + 1 \qquad F = Ue^{\delta_1 T},$$

where δ_1 and δ_2 determine the curvature L and U. The case of

1. $\delta_1 = \delta_2 = 0$ corresponds to two flat boundaries.
2. $\delta_1 < 0 < \delta_2$ corresponds to a lower boundary exponentially growing as time elapses, while the upper boundary will be exponentially decaying.
3. $\delta_1 > 0 > \delta_2$ corresponds to a convex downward lower boundary and a convex upward upper boundary.

Put Up-and-Out-Down-and-Out
Payoff: $p(S, U, L, T) = \max(X - S; 0)$ if $L < S < U$ before T else 0.

$$p = +Xe^{-rT} \sum_{n=-\infty}^{\infty} \left\{ \left(\frac{U^n}{L^n}\right)^{\mu_1 - 2} \left(\frac{L}{S}\right)^{\mu_2} \times [N(y_1 - \sigma\sqrt{T}) - N(y_2 - \sigma\sqrt{T})] - \left(\frac{L^{n+1}}{U^n S}\right)^{\mu_3 - 2} [N(y_3 - \sigma\sqrt{T}) - N(y_4 - \sigma\sqrt{T})] \right\}$$
$$- Se^{(b-r)T} \sum_{n=-\infty}^{\infty} \left\{ \left(\frac{U^n}{L^n}\right)^{\mu_1} \left(\frac{L}{S}\right)^{\mu_2} [N(y_1) - N(y_2)] - \left(\frac{L^{n+1}}{U^n S}\right)^{\mu_3} [N(y_3) - N(y_4)] \right\}, \tag{4.58}$$

where

$$y_1 = \frac{\ln(SU^{2n}/(EL^{2n})) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$y_2 = \frac{\ln(SU^{2n}/(XL^{2n})) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$y_3 = \frac{\ln(L^{2n+2}/(ESU^{2n})) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$y_4 = \frac{\ln(L^{2n+2}/(XSU^{2n})) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$E = Le^{\delta_2 T}$$

The double-barrier options are expressed as infinite series of weighted normal distribution functions. However, numerical studies

TABLE 4-15

Examples of Call Up-and-Out-Down-and-Out Values

($S = 100, X = 100, r = 0.1, b = 0.1$)

				$T = 0.25$			$T = 0.5$		
L	U	δ_1	δ_2	$\sigma = 0.15$	$\sigma = 0.25$	$\sigma = 0.35$	$\sigma = 0.15$	$\sigma = 0.25$	$\sigma = 0.35$
50	150	0	0	4.3515	6.1644	7.0373	6.9853	7.9336	6.5088
60	140	0	0	4.3505	5.8500	5.7726	6.8082	6.3383	4.3841
70	130	0	0	4.3139	4.8293	3.7765	5.9697	4.0004	2.2563
80	120	0	0	3.7516	2.6387	1.4903	3.5805	1.5098	0.5635
90	110	0	0	1.2055	0.3098	0.0477	0.5537	0.0441	0.0011
50	150	−0.1	0.1	4.3514	6.0997	6.6987	6.8974	6.9821	5.2107
60	140	−0.1	0.1	4.3478	5.6351	5.2463	6.4094	5.0199	3.1503
70	130	−0.1	0.1	4.2558	4.3291	3.1540	4.8182	2.6259	1.3424
80	120	−0.1	0.1	3.2953	1.9868	1.0351	1.9245	0.6455	0.1817
90	110	−0.1	0.1	0.5887	0.1016	0.0085	0.0398	0.0002	0.0000
50	150	0.1	−0.1	4.3515	6.2040	7.3151	7.0086	8.6080	7.7218
60	140	0.1	−0.1	4.3512	5.9998	6.2395	6.9572	7.4267	5.6620
70	130	0.1	−0.1	4.3382	5.2358	4.3859	6.6058	5.3761	3.3446
80	120	0.1	−0.1	4.0428	3.2872	2.0048	5.0718	2.6591	1.1871
90	110	0.1	−0.1	1.9229	0.6451	0.1441	1.7079	0.3038	0.0255

show that the convergence of the formulas is rapid. The numerical study of Ikeda and Kuintomo (1992) suggests that it suffices to calculate the leading two or three terms for most cases. The Ikeda and Kuntomo formula only holds when the strike price is inside the barrier range. For double barrier options when the strike is outside the barrier range see section on "Double-Barrier Option Using Barrier Symmetry."

Table 4-15 gives examples of call-up-and-out-down-and-out option values for different choices of lower L and upper U barrier, barrier curvatures δ_1 and δ_1, volatility σ, and time to maturity T.

Computer algorithm

The *DoubleBarrier*(·) function can be used to value four types of double barrier options:

1. *TypeFlag* set equal to "*co*" gives the value of an up-and-out-down-and-out call.
2. *TypeFlag* set equal to "*ci*" gives the value of an up-and-in-down-and-in call.
3. *TypeFlag* set equal to "*po*" gives the value of an up-and-out-down-and-out put.

4. *TypeFlag* set equal to "*pi*" gives the value of an up-and-in-down-and-in put.

```
Function DoubleBarrier(TypeFlag As String, S As Double, X As Double, L As Double, _
 U As Double, T As Double, r As Double, b As Double, v As Double, _
 delta1 As Double, delta2 As Double) As Double

    Dim E As Double, F As Double
    Dim Sum1 As Double, Sum2 As Double
    Dim d1 As Double, d2 As Double
    Dim d3 As Double, d4 As Double
    Dim mu1 As Double, mu2 As Double, mu3 As Double
    Dim OutValue As Double, n As Integer

    F = U * Exp(delta1 * T)
    E = L * Exp(delta2 * T)
    Sum1 = 0
    Sum2 = 0
    If TypeFlag = "co" Or TypeFlag = "ci" Then
        For n = -5 To 5
            d1 = (Log(S * U^(2 * n)/(X * L^(2 * n))) + (b + v^2 / 2) * T) / (v * Sqr(T))
            d2 = (Log(S * U^(2 * n)/(F * L^(2 * n))) + (b + v^2 / 2) * T) / (v * Sqr(T))
            d3 = (Log(L^(2 * n + 2)/(X * S * U^(2 * n))) _
            + (b + v^2 / 2) * T) / (v * Sqr(T))
            d4 = (Log(L^(2 * n + 2)/(F * S * U^(2 * n))) _
            + (b + v^2 / 2) * T) / (v * Sqr(T))
            mu1 = 2 * (b - delta2 - n * (delta1 - delta2)) / v^2 + 1
            mu2 = 2 * n * (delta1 - delta2) / v^2
            mu3 = 2 * (b - delta2 + n * (delta1 - delta2)) / v^2 + 1
            Sum1 = Sum1 + (U^n / L^n)^mu1 * (L / S)^mu2 * (CND(d1) - CND(d2)) _
             - (L^(n + 1) / (U^n * S))^mu3 * (CND(d3) - CND(d4))
            Sum2 = Sum2 + (U^n / L^n)^(mu1 - 2) * (L / S)^mu2 _
             * (CND(d1 - v * Sqr(T)) - CND(d2 - v * Sqr(T))) - (L^(n + 1) _
             / (U^n * S))^(mu3 - 2) * (CND(d3 - v * Sqr(T)) - CND(d4 - v * Sqr(T)))
        Next
        OutValue = S * Exp((b - r) * T) * Sum1 - X * Exp(-r * T) * Sum2
    ElseIf TypeFlag = "po" Or TypeFlag = "pi" Then
        For n = -5 To 5
            d1 = (Log(S * U^(2 * n)/(E * L^(2 * n))) _
            + (b + v^2 / 2) * T) / (v * Sqr(T))
            d2 = (Log(S * U^(2 * n)/(X * L^(2 * n))) _
            + (b + v^2 / 2) * T) / (v * Sqr(T))
            d3 = (Log(L^(2 * n + 2)/(E * S * U^(2 * n))) _
            + (b + v^2 / 2) * T) / (v * Sqr(T))
            d4 = (Log(L^(2 * n + 2)/(X * S * U^(2 * n))) _
            + (b + v^2 / 2) * T) / (v * Sqr(T))
            mu1 = 2 * (b - delta2 - n * (delta1 - delta2)) / v^2 + 1
            mu2 = 2 * n * (delta1 - delta2) / v^2
            mu3 = 2 * (b - delta2 + n * (delta1 - delta2)) / v^2 + 1
            Sum1 = Sum1 + (U^n / L^n)^mu1 * (L / S)^mu2 _
             * (CND(d1) - CND(d2)) - (L^(n + 1) / (U^n * S))^mu3 _
             * (CND(d3) - CND(d4))
            Sum2 = Sum2 + (U^n / L^n)^(mu1 - 2) * (L / S)^mu2 _
             * (CND(d1 - v * Sqr(T)) - CND(d2 - v * Sqr(T))) - (L^(n + 1) _
             / (U^n * S))^(mu3 - 2) _
             * (CND(d3 - v * Sqr(T)) - CND(d4 - v * Sqr(T)))
        Next
        OutValue = X * Exp(-r * T) * Sum2 - S * Exp((b - r) * T) * Sum1
```

```
    End If

    If TypeFlag = "co" Or TypeFlag = "po" Then
        DoubleBarrier = OutValue
    ElseIf TypeFlag = "ci" Then
        DoubleBarrier = GBlackScholes("c", S, X, T, r, b, v) - OutValue
    ElseIf TypeFlag = "pi" Then
        DoubleBarrier = GBlackScholes("p", S, X, T, r, b, v) - OutValue
    End If

End Function
```

The computer code calculates the value of a double-out barrier option. If the option is a double-in option, the computer code uses the barrier parity. An up-and-in-down-and-in call can, for instance, be constructed by going long a standard call option and going short an up-and-out-down-and-out call. The *GBlackScholes*(·) function at the end of the *DoubleBarrier*(·) function calls the generalized Black-Scholes-Merton function from Chapter 1.

4.17.4 Partial-Time Single-Asset Barrier Options

In single-asset partial-time barrier options, the period during which the underlying price is monitored for hitting the barrier is restricted to only a fraction of the option's lifetime. For partial-time-start barrier options, which we call type A options, the monitoring period starts when the option is initiated and ends at an arbitrary date t_1 before expiration. The monitoring period of partial-time-end-barrier options starts at an arbitrary date t_1 before expiration and ends at expiration T_2. Formulas for pricing this type of options were originally published by Heynen and Kat (1994b).

Partial-Time-Start-Out Options

$$c_A = Se^{(b-r)T_2}\left[M(d_1, \eta e_1; \eta\rho) - \left(\frac{H}{S}\right)^{2(\mu+1)} M(f_1, \eta e_3; \eta\rho)\right]$$

$$- Xe^{-rT_2}\left[M(d_2, \eta e_2; \eta\rho) - \left(\frac{H}{S}\right)^{2\mu} M(f_2, \eta e_4; \eta\rho)\right], \tag{4.59}$$

where $\eta = -1$ for an up-and-out call (c_{uoA}) and $\eta = 1$ for a down-and-out call (c_{doA}), and

$$d_1 = \frac{\ln(S/X) + (b + \sigma^2/2)T_2}{\sigma\sqrt{T_2}}, \quad d_2 = d_1 - \sigma\sqrt{T_2}$$

$$f_1 = \frac{\ln(S/X) + 2\ln(H/S) + (b + \sigma^2/2)T_2}{\sigma\sqrt{T_2}}, \quad f_2 = f_1 - \sigma\sqrt{T_2}$$

$$e_1 = \frac{\ln(S/H) + (b + \sigma^2/2)t_1}{\sigma\sqrt{t_1}}, \quad e_2 = e_1 - \sigma\sqrt{t_1}$$

$$e_3 = e_1 + \frac{2\ln(H/S)}{\sigma\sqrt{t_1}}, \quad e_4 = e_3 - \sigma\sqrt{t_1}$$

$$\mu = \frac{b - \sigma^2/2}{\sigma^2}, \quad \rho = \sqrt{t_1/T_2}$$

Partial-Time-Start-In Options

The price of "in" options of type A can be found using "out" options in combination with plain vanilla call options.

Up-and-In Call

$$c_{uiA} = call - c_{uoA}$$

Down-and-In Call

$$c_{diA} = call - c_{doA}$$

Partial-Time-End-Out Calls

There are two types of partial-time-end barrier options. Type B1 is defined such that only a barrier hit or crossing causes the option to be knocked out. We do not distinguish between the asset price hitting the barrier from above or below. In this case, there is no difference between up and down options.

When $X > H$, the knock-out call value is given by

$$c_{oB1} = Se^{(b-r)T_2}\left[M(d_1, e_1; \rho) - \left(\frac{H}{S}\right)^{2(\mu+1)} M(f_1, -e_3; -\rho)\right]$$
$$- Xe^{-rT_2}\left[M(d_2, e_2; \rho) - \left(\frac{H}{S}\right)^{2\mu} M(f_2, -e_4; -\rho)\right], \quad (4.60)$$

and when $X < H$, the knock-out call value is given by

$$c_{oB1} = Se^{(b-r)T_2}\left[M(-g_1, -e_1; \rho) - \left(\frac{H}{S}\right)^{2(\mu+1)} M(-g_3, e_3; -\rho)\right]$$
$$- Xe^{-rT_2}\left[M(-g_2, -e_2; \rho) - \left(\frac{H}{S}\right)^{2\mu} M(-g_4, e_4; -\rho)\right]$$
$$- Se^{(b-r)T_2}\left[M(-d_1, -e_1; \rho) - \left(\frac{H}{S}\right)^{2(\mu+1)} M(-f_1, e_3; -\rho)\right]$$
$$+ Xe^{-rT_2}\left[M(-d_2, -e_2; \rho) - \left(\frac{H}{S}\right)^{2\mu} M(-f_2, e_4; -\rho)\right]$$
$$+ Se^{(b-r)T_2}\left[M(g_1, e_1; \rho) - \left(\frac{H}{S}\right)^{2(\mu+1)} M(g_3, -e_3; -\rho)\right]$$
$$- Xe^{-rT_2}\left[M(g_2, e_2; \rho) - \left(\frac{H}{S}\right)^{2\mu} M(g_4, -e_4; -\rho)\right], \quad (4.61)$$

TABLE 4-16

Partial-Time-End Barrier Call Type B1 Option Values

($H = 100, r = b = 0.1, \sigma = 0.25, T_2 = 1$)

		Barrier Monitoring Time t_1				
S	X	0	0.25	0.5	0.75	1
95	90	0.0393	6.2747	10.3345	13.4342	17.1612
95	110	0.0000	3.7352	5.8712	7.1270	7.5763
105	90	9.8751	15.6324	19.2896	22.0753	25.4213
105	110	6.2303	9.6812	11.6055	12.7342	13.1376

where

$$g_1 = \frac{\ln(S/H) + (b + \sigma^2/2)T_2}{\sigma\sqrt{T_2}}, \quad g_2 = g_1 - \sigma\sqrt{T_2}$$

$$g_3 = g_1 + \frac{2\ln(H/S)}{\sigma\sqrt{T_2}}, \quad g_4 = g_3 - \sigma\sqrt{T_2}$$

Table 4-16 shows values for partial-time-end-barrier call options for a range of input parameters.

Type B2 partial-end barrier options are defined such that a down-and-out call is knocked out as soon as the underlying price is below the barrier. Similarly, an up-and-out call is knocked out as soon as the underlying price is above the barrier.

Down-and-Out Call $X < H$

$$\begin{aligned} c_{doB2} = {} & Se^{(b-r)T_2}\left[M(g_1, e_1; \rho) - \left(\frac{H}{S}\right)^{2(\mu+1)} M(g_3, -e_3; -\rho)\right] \\ & - Xe^{-rT_2}\left[M(g_2, e_2; \rho) - \left(\frac{H}{S}\right)^{2\mu} M(g_4, -e_4; -\rho)\right] \end{aligned} \tag{4.62}$$

Up-and-Out Call $X < H$

$$\begin{aligned} c_{uoB2} = {} & Se^{(b-r)T_2}\left[M(-g_1, -e_1; \rho) - \left(\frac{H}{S}\right)^{2(\mu+1)} M(-g_3, e_3; -\rho)\right] \\ & - Xe^{-rT_2}\left[M(-g_2, -e_2; \rho) - \left(\frac{H}{S}\right)^{2\mu} M(-g_4, e_4; -\rho)\right] \\ & - Se^{(b-r)T_2}\left[M(-d_1, -e_1; \rho) - \left(\frac{H}{S}\right)^{2(\mu+1)} M(e_3, -f_1; -\rho)\right] \\ & + Xe^{-rT_2}\left[M(-d_2, -e_2; \rho) - \left(\frac{H}{S}\right)^{2\mu} M(e_4, -f_2; -\rho)\right] \end{aligned} \tag{4.63}$$

4.17.5 Look-Barrier Options

Look-barrier options can be regarded as a combination of a partial-time barrier option and a forward starting fixed-strike lookback option (Bermin, 1996a). The option's barrier monitoring period starts at the option's starting date and ends at an arbitrary date t_1 before expiration. If the barrier is not hit during the monitoring period, the fixed-strike lookback option will be initiated at the same time the barrier ceases to exist. Because a knock-out barrier is introduced in the first part of the option's lifetime, the option will be cheaper than a standard partial-time fixed-strike lookback option.

$$
\begin{aligned}
w = \eta \Bigg\{ & Se^{(b-r)T_2}\left(1+\frac{\sigma^2}{2b}\right)\left[M_\eta\left(\frac{m-\mu_2 t_1}{\sigma\sqrt{t_1}}, \frac{-k+\mu_2 T_2}{\sigma\sqrt{T_2}}; -\rho\right)\right.\\
& \left. -e^{2\mu_2 h/\sigma^2} M_\eta\left(\frac{m-2h-\mu_2 t_1}{\sigma\sqrt{t_1}}, \frac{2h-k+\mu_2 T_2}{\sigma\sqrt{T_2}}; -\rho\right)\right]\\
& - e^{-rT_2} X\left[M_\eta\left(\frac{m-\mu_1 t_1}{\sigma\sqrt{t_1}}, \frac{-k+\mu_1 T_2}{\sigma\sqrt{T_2}}; -\rho\right)\right.\\
& \left. -e^{2\mu h/\sigma^2} M_\eta\left(\frac{m-2h-\mu_1 t_1}{\sigma\sqrt{t_1}}, \frac{2h-k+\mu_1 T_2}{\sigma\sqrt{T_2}}; -\rho\right)\right]\\
& - e^{-rT_2}\left(\frac{\sigma^2}{2b}\right)\left[S\left(\frac{S}{X}\right)^{-\frac{2b}{\sigma^2}} M_\eta\left(\frac{m+\mu_1 t_1}{\sigma\sqrt{t_1}}, \frac{-k-\mu_1 T_2}{\sigma\sqrt{T_2}}; -\rho\right)\right.\\
& \left. -H\left(\frac{H}{X}\right)^{-\frac{2b}{\sigma^2}} M_\eta\left(\frac{m-2h+\mu_1 t_1}{\sigma\sqrt{t_1}}, \frac{2h-k-\mu_1 T_2}{\sigma\sqrt{T_2}}; -\rho\right)\right]\\
& + Se^{(b-r)T_2}\left[\left(1+\frac{\sigma^2}{2b}\right) N_\eta\left(\frac{\mu_2(T_2-t_1)}{\sigma\sqrt{T_2-t_1}}\right) + e^{-b(T_2-t_1)}\left(1-\frac{\sigma^2}{2b}\right)\right.\\
& \left. \times N_\eta\left(\frac{-\mu_1(T_2-t_1)}{\sigma\sqrt{T_2-t_1}}\right)\right] g_1 - e^{-rT_2} X g_2 \Bigg\},
\end{aligned}
\tag{4.64}
$$

where $N_\eta(x) = N(\eta x)$, and $M_\eta(a, b; \rho) = M(\eta a, \eta b; \rho)$,

$$
\eta = \begin{cases} 1 & \text{if up-and-out call} \\ -1 & \text{if down-and-out put} \end{cases}
$$

$$
m = \begin{cases} \min(h, k) & \text{when } \eta = 1 \\ \max(h, k) & \text{when } \eta = -1 \end{cases}
$$

$$
h = \ln(H/S), \qquad k = \ln(X/S)
$$

$$
\mu_1 = b - \sigma^2/2, \qquad \mu_2 = b + \sigma^2/2, \qquad \rho = \sqrt{\frac{t_1}{T_2}}
$$

TABLE 4-17

Look-Barrier Up-and-Out Call Values

($S = X = 100, T_2 = 1, r = b = 0.1$)

		Barrier Monitoring Time t_1				
	H	0	0.25	0.5	0.75	1
	110	17.5212	9.6529	4.2419	1.7112	0.2388
$\sigma = 0.15$	120	17.5212	16.0504	11.0593	6.4404	2.1866
	130	17.5212	17.0597	14.9975	11.1547	5.5255
	110	30.1874	7.4146	2.7025	0.8896	0.0357
$\sigma = 0.3$	120	30.1874	16.4987	7.5509	3.1682	0.4259
	130	30.1874	23.1605	13.1118	6.6034	1.5180

$$g_1 = \left\{\left[N_\eta\left(\frac{h-\mu_2 t_1}{\sigma\sqrt{t_1}}\right) - e^{2\mu_2 h/\sigma^2} N_\eta\left(\frac{-h-\mu_2 t_1}{\sigma\sqrt{t_1}}\right)\right] - \left[N_\eta\left(\frac{m-\mu_2 t_1}{\sigma\sqrt{t_1}}\right) - e^{2\mu_2 h/\sigma^2} N_\eta\left(\frac{m-2h-\mu_2 t_1}{\sigma\sqrt{t_1}}\right)\right]\right\}$$

$$g_2 = \left\{\left[N_\eta\left(\frac{h-\mu_1 t_1}{\sigma\sqrt{t_1}}\right) - e^{2\mu_1 h/\sigma^2} N_\eta\left(\frac{-h-\mu_1 t_1}{\sigma\sqrt{t_1}}\right)\right] - \left[N_\eta\left(\frac{m-\mu_1 t_1}{\sigma\sqrt{t_1}}\right) - e^{2\mu_1 h/\sigma^2} N_\eta\left(\frac{m-2h-\mu_1 t_1}{\sigma\sqrt{t_1}}\right)\right]\right\}$$

Table 4-17 shows examples of look-barrier up-and-out call option values. It reports values for different choices of barrier H, barrier monitoring time t_1, and volatility σ.

4.17.6 Discrete-Barrier Options

All the barrier option pricing formulas presented so far assume continuous monitoring of the barrier. In practice, the barrier is normally monitored only at discrete points in time. An exception is the currency options market, where the barrier is frequently monitored almost continuously. For equity, commodity, and interest rate options, the barrier is typically monitored against an official daily closing price. Discrete monitoring will naturally lower the probability of barrier hits compared with continuous barrier monitoring. Broadie, Glasserman, and Kou (1995) have developed an approximation for a continuity correction for discrete barrier options. The correction shifts the barrier (or the barriers) away from the underlying asset. The probability of barrier hits is thus reduced in the model. To price any discrete barrier option, it is sufficient to replace the continuously monitored barrier

H in continuous barrier options formulas with a discrete barrier level H_D equal to

$$H_D = He^{\beta\sigma\sqrt{\Delta t}}$$

if the barrier is above the underlying security, and to

$$H_D = He^{-\beta\sigma\sqrt{\Delta t}}$$

if the barrier is below the underlying security. Δt is the time between monitoring events, and $\beta = \zeta\left(\frac{1}{2}\right)/\sqrt{2\pi} \approx 0.5826$, where ζ is the Riemann zeta function. Broadie, Glasserman, and Kou (1995) show both theoretically and through examples that discrete barrier options can be priced with remarkable accuracy using this simple correction. Other methods for pricing discrete barrier options have been published by Kat and Verdonk (1995), Reimer and Sandemann (1995), and Heynen and Kat (1996b).

4.17.7 Soft-Barrier Options

A soft-barrier option is similar to a standard barrier option, except that the barrier is no longer a single level. Rather, it's a soft range between an upper level U and a lower level L. Soft-barrier options are knocked in or out proportionally. For instance, consider a soft down-and-out call with a current asset price 100, with a soft-barrier range from $U = 90$ to $L = 80$. If the lowest asset price during the lifetime is 86, then 40% of the call will be knocked out. The valuation formula originally introduced by Hart and Ross (1994) can be used to price soft-down-and-in-call and soft-up-and-in-put options:

$$w = \frac{1}{U-L}\left\{\eta S e^{(b-r)T} S^{-2\mu}\frac{(SX)^{\mu+0.5}}{2(\mu+0.5)}\left[\left(\frac{U^2}{SX}\right)^{\mu+0.5}\right.\right.$$
$$\left. N(\eta d_1) - \lambda_1 N(\eta d_2) - \left(\frac{L^2}{SX}\right)^{\mu+0.5} N(\eta e_1) + \lambda_1 N(\eta e_2)\right]$$
$$- \eta X e^{-rT} S^{-2(\mu-1)}\frac{(SX)^{\mu-0.5}}{2(\mu-0.5)}\left[\left(\frac{U^2}{SX}\right)^{\mu-0.5} N(\eta d_3)\right.$$
$$\left.\left. -\lambda_2 N(\eta d_4) - \left(\frac{L^2}{SX}\right)^{\mu-0.5} N(\eta e_3) + \lambda_2 N(\eta e_4)\right]\right\}, \quad (4.65)$$

where η is set to 1 for a call and -1 for a put, and

$$d_1 = \frac{\ln(U^2/(SX))}{\sigma\sqrt{T}} + \mu\sigma\sqrt{T} \qquad d_2 = d_1 - (\mu + 0.5)\sigma\sqrt{T}$$

$$d_3 = \frac{\ln(U^2/(SX))}{\sigma\sqrt{T}} + (\mu - 1)\sigma\sqrt{T} \qquad d_4 = d_3 - (\mu - 0.5)\sigma\sqrt{T}$$

$$e_1 = \frac{\ln(L^2/(SX))}{\sigma\sqrt{T}} + \mu\sigma\sqrt{T} \qquad e_2 = e_1 - (\mu + 0.5)\sigma\sqrt{T}$$

$$e_3 = \frac{\ln(L^2/(SX))}{\sigma\sqrt{T}} + (\mu - 1)\sigma\sqrt{T} \qquad e_4 = e_3 - (\mu - 0.5)\sigma\sqrt{T}$$

$$\lambda_1 = e^{-0.5[\sigma^2 T(\mu+0.5)(\mu-0.5)]} \qquad \lambda_2 = e^{-0.5[\sigma^2 T(\mu-0.5)(\mu-1.5)]}$$

$$\mu = \frac{b + \sigma^2/2}{\sigma^2}$$

The value of a soft down-and-out call is equal to the value of a standard call minus the value of a soft down-and-in call. Similarly, the value of a soft up-and-out put is equal to the value of a standard put minus a soft up-and-in put.

Application

Standard barrier options are hard to delta hedge when the asset price is close to the barrier. The barrier option will then have a high gamma risk. Soft-barrier options will typically have a significantly lower gamma risk and will for that reason also be easier to hedge.

TABLE 4-18

Soft-Barrier Down-and-Out Call Values

($S = 100, X = 100, U = 95, r = 0.1, b = 0.05$)

	$T = 0.5$			$T = 1$		
L	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$
95	3.8075	4.5263	4.7297	5.4187	5.3614	5.2300
90	4.0175	5.5615	6.2595	6.0758	6.9776	7.2046
85	4.0529	6.0394	7.2496	6.2641	7.9662	8.7092
80	4.0648	6.2594	7.8567	6.3336	8.5432	9.8118
75	4.0708	6.3740	8.2253	6.3685	8.8822	10.5964
70	4.0744	6.4429	8.4578	6.3894	9.0931	11.1476
65	4.0768	6.4889	8.6142	6.4034	9.2343	11.5384
60	4.0785	6.5217	8.7260	6.4133	9.3353	11.8228
55	4.0798	6.5463	8.8099	6.4208	9.4110	12.0369
50	4.0808	6.5654	8.8751	6.4266	9.4698	12.2036

Table 4-18 shows values for soft-barrier down-and-out call options. Different choices for time to maturity T, lower barrier level L, and volatility σ are reported.

4.17.8 Use of Put-Call Symmetry for Barrier Options

The European put-call value symmetry first published by Bates (1991) and later applied for static replication of barrier options by Carr (1994), Carr and Bowie (1994), and Carr, Ellis, and Gupta (1998) states that a call with strike X when the spot is at H must have the same value as $\frac{X}{He^{bT}}$ number of puts with strike $\frac{(He^{bT})^2}{X}$.[8]

$$c(S, X, T, r, b, \sigma) = \frac{X}{Se^{bT}} \times p\left(S, \frac{(Se^{bT})^2}{X}, T, r, b, \sigma\right)$$

The put-call symmetry is useful to construct a static hedge consisting of standard options when one is hedging exotic options, such as barrier options. Barrier options often have complex and high gamma risk when the asset is close to the barrier. In that case, a static hedge consisting of plain vanilla options will cover the risk far better than dynamic delta hedging.

The put-call symmetry is based on assumptions that the volatility smile (implied volatilities for different strikes) is symmetric in the natural logarithm of $\frac{X}{Se^{bT}}$. As a result, a put with strike $\frac{(He^{bT})^2}{X}$ will have the same volatility as a call with strike X. Put options will have higher volatility than calls equidistant from the forward price.

Example

Consider a down-and-in call option on a futures contract with barrier $H = 120$ and strike $X = 140$. A static hedge can be constructed by using the put-call symmetry:

- Buy $\frac{140}{120e^{0\times T}} = 1.1667$ standard put options.
- Choose a strike of $\frac{(120e^{0\times T})^2}{140} = 102.8571$.

If the asset price closes above the barrier, both the down-and-out call and the standard put option will expire worthless. If the asset price hits the barrier before expiration, the puts will have the same value as one standard call option with strike 140. Because the put options are sold at a barrier hit and at the same time the call option is bought, the put-call symmetry will ensure a zero net cost.

---

[8]For more on static option replication, see Derman, Ergener, and Kani (1995).

4.18 BARRIER OPTION SYMMETRIES

Haug (1998) and Gao, Huang, and Subrahmanyam (2000) have described symmetries between put and call barrier options. As described by Haug (1998) and Haug (2001b), these symmetries can be useful to value many exotic barrier options—discussed in more detail below.

For standard barrier options, the symmetry between standard in-call and in-put options is

$$C_{di}(S,X,H,r,b,\sigma) = P_{ui}\left(X,S,\frac{SX}{H},r-b,-b,\sigma\right) \tag{4.66}$$
$$= \frac{X}{S_t}P_{ui}\left(S,\frac{S^2}{X},\frac{S^2}{H},r-b,-b,\sigma\right)$$

$$C_{ui}(S,X,H,r,b,\sigma) = P_{di}\left(X,S,\frac{SX}{H},r-b,-b,\sigma\right) \tag{4.67}$$
$$= \frac{X}{S}P_{di}\left(S,\frac{S^2}{X},\frac{S^2}{H},r-b,-b,\sigma\right),$$

where C_{di} is a down-and-in call, and C_{ui} is an up-and-in call (similarly for puts). The put-call symmetry between out barrier options is given by

$$C_{do}(S,X,H,r,b,\sigma) = P_{uo}\left(X,S,\frac{SX}{H},r-b,-b,\sigma\right) \tag{4.68}$$
$$= \frac{X}{S}P_{uo}\left(S,\frac{S^2}{X},\frac{S^2}{H},r-b,-b,\sigma\right)$$

$$C_{uo}(S,X,H,r,b,\sigma) = P_{do}\left(X,S,\frac{SX}{H},r-b,-b,\sigma\right) \tag{4.69}$$
$$= \frac{X}{S}P_{do}\left(S,\frac{S^2}{X},\frac{S^2}{H},r-b,-b,\sigma\right),$$

and for double barrier options, we have

$$C_o(S,X,L,U,r,b,\sigma) = P_o\left(X,S,\frac{SX}{U},\frac{SX}{L},r-b,-b,\sigma\right) \tag{4.70}$$
$$= \frac{X}{S}P_o\left(S,\frac{S^2}{X},\frac{S^2}{U},\frac{S^2}{L},r-b,-b,\sigma\right)$$

$$C_i(S,X,L,U,r,b,\sigma) = P_i\left(X,S,\frac{SX}{U},\frac{SX}{L},r-b,-b,\sigma\right) \tag{4.71}$$
$$= \frac{X}{S}P_i\left(S,\frac{S^2}{X},\frac{S^2}{U},\frac{S^2}{L},r-b,-b,\sigma\right),$$

where L is the lower barrier and U is the upper barrier. These transformations also hold for partial-time single- and double-barrier options described by Heynen and Kat (1994b) and Hui (1997).

4.18.1 First-Then-Barrier Options

Haug (1998) used put-call barrier symmetry to design first-then-barrier options on futures/forwards (cost-of-carry zero). An example is a first-down-then-up-and-in call $c_{dui}(S, X, L, U)$, for which the owner receives a standard up-and-in call with barrier $U(U > S)$ and strike X if the asset first hits a lower barrier $L(L < S < U)$. The asset price can naturally hit the lower barrier several times before it hits the upper barrier. To be knocked in, the asset price has to hit the lower barrier at least once before an upper barrier hit has any effect. Using the up-and-in call/down-and-in put barrier symmetry described above, we can simply construct a static hedge, and thereby a valuation formula:

$$c_{dui}(S, X, L, U, T, r, \sigma) = \frac{X}{L} p_{di}\left(S, \frac{L^2}{X}, \frac{L^2}{U}, T, r, \sigma\right) \tag{4.72}$$

Hence, to hedge a first-down-then-up-and-in barrier call option, we simply need to buy $\frac{X}{L}$ number of standard down-and-in puts with strike $\frac{L^2}{X}$ and barrier $\frac{L^2}{U}$.[9] Both the first-down-then-up-and-in call and the standard down-and-in put will expire worthless if the asset price never touches L. On the other hand, if the asset price hits the lower barrier L, the value of the $\frac{X}{L}$ down-and-in puts will exactly match the value of the up-and-in call. In this case, we just need to sell the down-and-in put while buying the up-and-in call. Evidently, we have created a "perfect" static hedge for this new barrier option using only standard barrier options and the barrier transformation principle.

One can easily construct static hedges and valuation formulas for a large class of first-then-barrier options in a similar fashion.

First-up-then-down-and-in call:

$$c_{udi}(S, X, L, U, T, r, \sigma) = \frac{X}{U} p_{ui}\left(S, \frac{U^2}{X}, \frac{U^2}{L}, T, r, \sigma\right) \tag{4.73}$$

First-down-then-up-and-in put:

$$p_{udi}(S, X, L, U, T, r, \sigma) = \frac{X}{L} c_{di}\left(S, \frac{L^2}{X}, \frac{L^2}{U}, T, r, \sigma\right) \tag{4.74}$$

[9] It is not necessary to be able to actually carry out such a static hedge. As long as the underlying asset price follows a geometric Brownian motion, we can still derive a valuation formula. From the formula we can easily derive the delta to dynamically replicate the option.

First-up-then-down-and-in put:

$$p_{udi}(S, X, L, U, T, r, \sigma) = \frac{X}{U} c_{ui}\left(S, \frac{U^2}{X}, \frac{U^2}{L}, T, r, \sigma\right) \tag{4.75}$$

For out options we can use the well-known in-out parity: out-barrier-option = long plain vanilla option + short in-barrier option. This yields

First-down-then-up-and-out call:

$$c_{duo}(S, X, L, U, T, r, \sigma) = c(S, X, T, r, \sigma) - c_{dui}(S, X, L, U, T, r, \sigma) \tag{4.76}$$

First-up-then-down-and-out call:

$$c_{udo}(S, X, L, U, T, r, \sigma) = c(S, X, T, r, \sigma) - c_{udi}(S, X, L, U, T, r, \sigma) \tag{4.77}$$

Table 4-19 compares plain vanilla options with a standard-up-and-out call and a first-down-then-up-and-out call, for different choices for volatility σ, and barrier levels L and U.

The value of the first-up-then-up-and-out call is evidently higher or equal to the standard barrier option. This is natural since the asset price has to hit the lower barrier before it hits the upper barrier for the option to be knocked out. The further the barriers are away from the current asset price, the less likely are barrier hits, and the option value converges to a plain vanilla call option. Low volatility has a similar effect on barrier hit probability. However, a first-then-out barrier option will always be less (or equal) in value to a similar plain vanilla option.

TABLE 4-19

Comparison of Plain Vanilla Call, Standard Down-and-Out Call, and First-Down-Then-Up-and-Out Call Option Values for Different Barrier Levels and Volatility

($S = 100, X = 100, T = 0.5, r = 0.1$)

		Plain Vanilla[1]		Standard Barrier[2]		First-Then-Barrier	
L	U	$\sigma = 10\%$	$\sigma = 30\%$	$\sigma = 10\%$	$\sigma = 30\%$	$\sigma = 10\%$	$\sigma = 30\%$
50	150	2.6828	8.0350	2.6828	8.0350	2.6828	8.0350
80	120	2.6828	8.0350	2.6828	7.9065	2.6828	7.9821
90	110	2.6828	8.0350	2.6801	6.3385	2.6827	6.4607
95	105	2.6828	8.0350	2.4626	3.9523	2.5284	3.9702

1. Using the Black (1976) formula.
2. Using the Reiner and Rubinstein (1991a) formula.

4.18.2 Double-Barrier Option Using Barrier Symmetry

Haug (1998) shows how to value a double-barrier option using only single-barrier option formulas in combination with put-call barrier symmetry (described in the end of the barrier option section). For options on an underlying asset with cost-of-carry zero (option on futures), we can accurately approximate the value of a double knock-in call barrier option with

$$c_i(S,X,L,U,T,r,\sigma)\approx \tag{4.78}$$

$$\min\Bigg[c(S,X,T);\ c_{ui}(S,X,U,T,r,\sigma)+c_{di}(S,X,L,T,r,\sigma)$$

$$-\frac{X}{U}p_{ui}\left(S,\frac{U^2}{X},\frac{U^2}{L},T,r,\sigma\right)-\frac{X}{L}p_{di}\left(S,\frac{L^2}{X},\frac{L^2}{U},T,r,\sigma\right)$$

$$+\frac{U}{L}c_{di}\left(S,\frac{L^2X}{U^2},\frac{L^3}{U^2},T,r,\sigma\right)+\frac{L}{U}c_{ui}\left(S,\frac{U^2X}{L^2},\frac{U^3}{L^2},T,r,\sigma\right)$$

$$-\frac{LX}{U^2}p_{ui}\left(S,\frac{U^4}{L^2X},\frac{U^4}{L^3},T,r,\sigma\right)-\frac{UX}{L^2}p_{di}\left(S,\frac{L^4}{U^2X},\frac{L^4}{U^3},T,r,\sigma\right)$$

$$+\frac{U^2}{L^2}c_{di}\left(S,\frac{L^4X}{U^4},\frac{L^5}{U^4},T,r,\sigma\right)+\frac{L^2}{U^2}c_{ui}\left(S,\frac{U^4X}{L^4},\frac{U^5}{L^4},T,r,\sigma\right)\Bigg],$$

where $c(S,X,T)$ is a standard call option and c_{ui} is a standard down-and-in barrier option etc. For a double knock-in put barrier option, we have

$$p_i(S,X,L,U,T,r,\sigma)\approx \tag{4.79}$$

$$\min\Bigg[p(S,X,T,r,\sigma);\ p_{ui}(S_t,X,U,T,r,\sigma)+p_{di}(S,X,L,T,r,\sigma)$$

$$-\frac{X}{U}c_{ui}\left(S,\frac{U^2}{X},\frac{U^2}{L},T,r,\sigma\right)-\frac{X}{L}c_{di}\left(S,\frac{L^2}{X},\frac{L^2}{U},T,r,\sigma\right)$$

$$+\frac{U}{L}p_{di}\left(S,\frac{L^2X}{U^2},\frac{L^3}{U^2},T,r,\sigma\right)+\frac{L}{U}p_{ui}\left(S,\frac{U^2X}{L^2},\frac{U^3}{L^2},T,r,\sigma\right)$$

$$-\frac{LX}{U^2}c_{ui}\left(S,\frac{U^4}{L^2X},\frac{U^4}{L^3},T,r,\sigma\right)-\frac{UX}{L^2}c_{di}\left(S,\frac{L^4}{U^2X},\frac{L^4}{U^3},T,r,\sigma\right)$$

$$+\frac{U^2}{L^2}p_{di}\left(S,\frac{L^4X}{U^4},\frac{L^5}{U^4},T,r,\sigma\right)+\frac{L^2}{U^2}p_{ui}\left(S,\frac{U^4X}{L^4},\frac{U^5}{L^4},T,r,\sigma\right)\Bigg]$$

TABLE 4-20

Examples of Up-and-Out-Down-and-Out Call Values

($S = 100, X = 100, r = 0.1, b = 0$)

		$T = 0.25$			$T = 1$		
L	U	$\sigma = 0.15$	$\sigma = 0.25$	$\sigma = 0.35$	$\sigma = 0.15$	$\sigma = 0.25$	$\sigma = 0.35$
			Haug Method Values				
50	150	2.9175	4.8136	5.9864	5.1564	5.1455	3.4284
60	140	2.9173	4.6265	5.0077	4.6500	3.5477	2.0222
70	130	2.9055	3.9425	3.3685	3.5014	1.8739	0.8131
80	120	2.6495	2.2807	1.3757	1.6735	0.4754	0.0778
90	110	1.0053	0.2882	0.0457	0.1150	0.0008	0.0000
			Ikeda Kunitomo Method Values				
50	150	2.9175	4.8136	5.9864	5.1564	5.1455	3.4284
60	140	2.9173	4.6265	5.0077	4.6500	3.5477	2.0222
70	130	2.9055	3.9425	3.3685	3.5014	1.8739	0.8131
80	120	2.6495	2.2807	1.3757	1.6735	0.4754	0.0778
90	110	1.0053	0.2882	0.0457	0.1150	0.0009	0.0000

Table 4-20 compares double-barrier option values calculated with the Haug formula with the Ikeda and Kunitomo formula. They are evidently close to identical for these parameters.

4.18.3 Dual Double-Barrier Options

The owner of a standard double knock-in call option receives a call if either the upper or the lower barrier is hit. It may be desirable, however, to get a put if the asset price hits the lower barrier and a call if the asset price hits the upper barrier—a knock-in double-barrier put-down-call-up. Using the technique described above, we can easily value the latter option. We limit ourselves also here to dual double-barrier options on futures or forwards. Haug (2005b) gives the following approximation:

$$c_{icp}(S, X, L, U, T, r, \sigma) \approx \min \Bigg[\max[c(S, X, T, r, \sigma); \qquad (4.80)$$

$$p(S, X, T, r, \sigma)]; c_{ui}(S, X, U, T, r, \sigma) + p_{di}(S, X, L, T, r, \sigma)$$

$$- \frac{X}{U} p_{ui}\left(S, \frac{U^2}{X}, \frac{U^2}{L}, T, r, \sigma\right) - \frac{X}{L} c_{di}\left(S, \frac{L^2}{X}, \frac{L^2}{U}, T, r, \sigma\right)$$

$$+ \frac{U}{L} c_{di}\left(S, \frac{L^2 X}{U^2}, \frac{L^3}{U^2}, T, r, \sigma\right) + \frac{L}{U} p_{ui}\left(S, \frac{U^2 X}{L^2}, \frac{U^3}{L^2}, T, r, \sigma\right)$$

$$- \frac{LX}{U^2} p_{ui}\left(S, \frac{U^4}{L^2X}, \frac{U^4}{L^3}, T, r, \sigma\right) - \frac{UX}{L^2} c_{di}\left(S, \frac{L^4}{U^2X}, \frac{L^4}{U^3}, T, r, \sigma\right)$$

$$+ \frac{U^2}{L^2} c_{di}\left(S, \frac{L^4X}{U^4}, \frac{L^5}{U^4}, T, r, \sigma\right) + \frac{L^2}{U^2} p_{ui}\left(S, \frac{U^4X}{L^4}, \frac{U^5}{L^4}, T, r, \sigma\right)\Bigg]$$

Similarly, the formula for a knock-in double barrier call-down-put-up is

$$c_{icd}(S, X, L, U, T, r, \sigma) \approx \min\Bigg[\max[c(S, X, T, r, \sigma); \quad (4.81)$$

$$p(S, X, T, r, \sigma)]; p_{ui}(S, X, U, T, r, \sigma) + c_{di}(S, X, L, T, r, \sigma)$$

$$- \frac{X}{U} c_{ui}\left(S, \frac{U^2}{X}, \frac{U^2}{L}, T, r, \sigma\right) - \frac{X}{L} p_{di}\left(S, \frac{L^2}{X}, \frac{L^2}{U}, T, r, \sigma\right)$$

$$+ \frac{U}{L} p_{di}\left(S, \frac{L^2X}{U^2}, \frac{L^3}{U^2}, T, r, \sigma\right) + \frac{L}{U} c_{ui}\left(S, \frac{U^2X}{L^2}, \frac{U^3}{L^2}, T, r, \sigma\right)$$

$$- \frac{LX}{U^2} c_{ui}\left(S, \frac{U^4}{L^2X}, \frac{U^4}{L^3}, T, r, \sigma\right) - \frac{UX}{L^2} p_{di}\left(S, \frac{L^4}{U^2X}, \frac{L^4}{U^3}, T, r, \sigma\right)$$

$$+ \frac{U^2}{L^2} p_{di}\left(S, \frac{L^4X}{U^4}, \frac{L^5}{U^4}, T, r, \sigma\right) + \frac{L^2}{U^2} c_{ui}\left(S, \frac{U^4X}{L^4}, \frac{U^5}{L^4}, T, r, \sigma\right)\Bigg]$$

A call-down-put-up knock-in option naturally has considerably lower value than a call-up-put-down knock-in option. The latter can be seen as a "poor man's" double-barrier option. These barrier options have traded in the OTC FX market.

Table 4-21 shows values for dual double-barrier options, with different choices for barriers L and U, and volatilities σ.

TABLE 4-21

Dual Double-Barrier Options

($S = 100$, $X = 100$, $T = 0.25$, $r = 0.08$, $b = 0$)

		call-up-put-down			put-up-call-down		
L	U	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$
70	130	0.0000	0.2368	2.6778	0.0000	0.0000	0.0009
75	125	0.0002	0.6753	4.5569	0.0000	0.0000	0.0063
80	120	0.0049	1.7827	5.8601	0.0000	0.0003	0.0400
85	115	0.0895	3.8906	5.8601	0.0000	0.0089	0.1791
90	110	0.8917	3.9088	5.8601	0.0001	0.1161	0.3405
95	105	1.9550	3.9088	5.8601	0.0570	0.2814	1.0719

4.19 BINARY OPTIONS

Binary options, aka digital options, are popular in the OTC markets for hedging and speculation. They are also important to financial engineers as building blocks for constructing more complex derivatives products.

4.19.1 Gap Options

The payoff from a call is 0 if $S \leq X_1$ and $S - X_2$ if $S > X_1$. Similarly, the payoff from a put is 0 if $S \geq X_1$ and $X_2 - S$ if $S < X_1$. The Reiner and Rubinstein (1991b) formula can be used to price these options:

$$c = Se^{(b-r)T}N(d_1) - X_2e^{-rT}N(d_2) \tag{4.82}$$

$$p = X_2e^{-rT}N(-d_2) - Se^{(b-r)T}N(-d_1), \tag{4.83}$$

where

$$d_1 = \frac{\ln(S/X_1) + (b + \sigma^2/2)T}{\sigma\sqrt{T}} \qquad d_2 = d_1 - \sigma\sqrt{T}$$

Notice that the payoff from this option can be negative, depending on the settings of X_1 and X_2. When the difference between X_1 and X_2 is such that the value of the option is zero, the option is often referred to as a pay-later option.

Example

Consider a gap call option with six months to expiration. The stock price is 50, the first strike is 50, the payoff strike is 57, the risk-free interest rate is 9% per year, and the volatility is 20% per year. $S = 50$, $X_1 = 50$, $X_2 = 57$, $T = 0.5$, $r = b = 0.09$, and $\sigma = 0.2$.

$$d_1 = \frac{\ln(50/50) + (0.09 + 0.2^2/2)0.5}{0.2\sqrt{0.5}} = 0.3889 \quad d_2 = d_1 - 0.2\sqrt{0.5} = 0.2475$$

$$N(d_1) = 0.6513, \quad N(d_2) = 0.5977$$

$$c = 50e^{(0.09-0.09)0.5}N(d_1) - 57e^{-0.09\times 0.5}N(d_2) = -0.0053$$

4.19.2 Cash-or-Nothing Options

The cash-or-nothing options pay an amount K at expiration if the option is in-the-money. The payoff from a call is 0 if $S \leq X$ and K if $S > X$. The payoff from a put is 0 if $S \geq X$ and K if $S < X$. Valuation of cash-or-nothing options can be made using the formula described by Reiner and Rubinstein (1991b):

$$c = Ke^{-rT}N(d) \tag{4.84}$$

$$p = Ke^{-rT}N(-d) \tag{4.85}$$

This is nothing but the last part of the Black-Scholes formula, where

$$d = \frac{\ln(S/X) + (b - \sigma^2/2)T}{\sigma\sqrt{T}}$$

Example
What is the value of a cash-or-nothing put option with nine months to expiration? The futures price is 100, the strike price is 80, the cash payout is 10, the risk-free interest rate is 6% per year, and the volatility is 35% per year. $S = 100$, $X = 80$, $K = 10$, $T = 0.75$, $r = 0.06$, $b = 0$, and $\sigma = 0.35$.

$$d = \frac{\ln(100/80) + (0 - 0.35^2/2)0.75}{0.35\sqrt{0.75}} = 0.5846$$

$$N(-d) = N(-0.5846) = 0.2794$$

$$p = 10e^{-0.06\times 0.75}N(-d) = 2.6710$$

4.19.3 Asset-or-Nothing Options

At expiration, the asset-or-nothing call option pays 0 if $S \leq X$ and S if $S > X$. Similarly, a put option pays 0 if $S \geq X$ and S if $S < X$. The option can be valued using the Cox and Rubinstein (1985) formula:[10]

$$c = Se^{(b-r)T}N(d) \tag{4.86}$$

$$p = Se^{(b-r)T}N(-d), \tag{4.87}$$

where

$$d = \frac{\ln(S/X) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$$

Example
Consider an asset-or-nothing put option with six months to expiration. The stock price is 70, the strike price is 65, the dividend yield is 5% per year, the risk-free interest rate is 7% per year, and the volatility is 27% per year. $S = 70$, $X = 65$, $T = 0.5$, $r = 0.07$, $b = 0.07 - 0.05 = 0.02$, and $\sigma = 0.27$.

$$d = \frac{\ln(70/65) + (0.02 + 0.27^2/2)0.5}{0.27\sqrt{0.5}} = 0.5360$$

$$N(-d) = N(-0.5360) = 0.2960$$

$$p = 70e^{(0.02-0.07)0.5}N(-d) = 20.2069$$

---

[10] See also Reiner and Rubinstein (1991b).

4.19.4 Supershare Options

A supershare option, originally introduced by Hakansson (1976), entitles its holder to a payoff of S/X_L if $X_L \le S < X_H$ and 0 otherwise.

$$w = (Se^{(b-r)T}/X_L)[N(d_1) - N(d_2)], \tag{4.88}$$

where

$$d_1 = \frac{\ln(S/X_L) + (b + \sigma^2/2)T}{\sigma\sqrt{T}} \qquad d_2 = \frac{\ln(S/X_H) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$$

Application
A portfolio of supershare options can be used to construct what is known as a superfund—traded at the American Stock Exchange under the name SuperUnits and at the Chicago Board Options Exchange as SuperShares.[11]

Example
Consider a supershare option with three months to expiration. The futures price is 100, the lower boundary X_L is 90, the upper boundary X_H is 110, the risk-free interest rate is 10% per year, and the volatility is 20% per year. $S = 100$, $X_L = 90$, $X_H = 110$, $T = 0.25$, $r = 0.1$, $b = 0$, and $\sigma = 0.2$.

$$d_1 = \frac{\ln(100/90) + (0 + 0.2^2/2)0.25}{0.2\sqrt{0.25}} = 1.1036$$

$$d_2 = \frac{\ln(100/110) + (0 + 0.2^2/2)0.25}{0.2\sqrt{0.25}} = -0.9031$$

$$N(d_1) = N(1.1036) = 0.8651 \quad N(d_2) = N(-0.9031) = 0.1832$$

$$w = (100e^{(0-0.1)0.25}/90)[N(d_1) - N(d_2)] = 0.7389$$

4.19.5 Binary Barrier Options

Reiner and Rubinstein (1991b) present a set of formulas that can be used to price 28 different types of so-called binary barrier options. The binary barrier options presented here can be divided into two main categories:

1. Cash-or-nothing barrier options. These pay out either a prespecified cash amount or nothing, depending on whether the asset price has hit the barrier or not.

[11]For more on supershares, see the articles by Cox and Rubinstein (1985), Hakansson (1991), and Rubinstein (1995b).

2. Asset-or-nothing barrier options. These pay out the value of the asset or nothing, depending on whether the asset price has hit the barrier or not.

We start by introducing nine factors:

$$A_1 = Se^{(b-r)T}N(\phi x_1)$$

$$B_1 = Ke^{-rT}N(\phi x_1 - \phi\sigma\sqrt{T})$$

$$A_2 = Se^{(b-r)T}N(\phi x_2)$$

$$B_2 = Ke^{-rT}N(\phi x_2 - \phi\sigma\sqrt{T})$$

$$A_3 = Se^{(b-r)T}(H/S)^{2(\mu+1)}N(\eta y_1)$$

$$B_3 = Ke^{-rT}(H/S)^{2\mu}N(\eta y_1 - \eta\sigma\sqrt{T})$$

$$A_4 = Se^{(b-r)T}(H/S)^{2(\mu+1)}N(\eta y_2)$$

$$B_4 = Ke^{-rT}(H/S)^{2\mu}N(\eta y_2 - \eta\sigma\sqrt{T})$$

$$A_5 = K[(H/S)^{\mu+\lambda}N(\eta z) + (H/S)^{\mu-\lambda}N(\eta z - 2\eta\lambda\sigma\sqrt{T})$$

where K is a prespecified cash amount. The binary variables η and ϕ each take the value 1 or -1. Moreover:

$$x_1 = \frac{\ln(S/X)}{\sigma\sqrt{T}} + (\mu+1)\sigma\sqrt{T} \qquad x_2 = \frac{\ln(S/H)}{\sigma\sqrt{T}} + (\mu+1)\sigma\sqrt{T}$$

$$y_1 = \frac{\ln(H^2/(SX))}{\sigma\sqrt{T}} + (\mu+1)\sigma\sqrt{T} \qquad y_2 = \frac{\ln(H/S)}{\sigma\sqrt{T}} + (\mu+1)\sigma\sqrt{T}$$

$$z = \frac{\ln(H/S)}{\sigma\sqrt{T}} + \lambda\sigma\sqrt{T} \qquad \mu = \frac{b-\sigma^2/2}{\sigma^2} \qquad \lambda = \sqrt{\mu^2 + \frac{2r}{\sigma^2}}$$

By using A_1 to A_5 and B_1 to B_4 in different combinations, one can price the 28 binary barrier options described below:

1. Down-and-in cash-(at-hit)-or-nothing ($S > H$):
   Value: A_5 $\quad \eta = 1$

2. Up-and-in cash-(at-hit)-or-nothing ($S < H$):
   Value: A_5 $\quad \eta = -1$

3. Down-and-in asset-(at-hit)-or-nothing ($S > H$):
   Value: $(K = H)$: A_5 $\quad \eta = 1$

4. Up-and-in asset-(at-hit)-or-nothing ($S < H$):
   Value: $(K = H)$: A_5 $\quad \eta = -1$

5. **Down-and-in cash-(at-expiration)-or-nothing** ($S > H$):
Value: $B_2 + B_4$ $\eta = 1,$ $\phi = -1$

6. **Up-and-in cash-(at-expiration)-or-nothing** ($S < H$):
Value: $B_2 + B_4$ $\eta = -1,$ $\phi = 1$

7. **Down-and-in asset-(at-expiration)-or-nothing** ($S > H$):
Value: $A_2 + A_4$ $\eta = 1,$ $\phi = -1$

8. **Up-and-in asset-(at-expiration)-or-nothing** ($S < H$):
Value: $A_2 + A_4$ $\eta = -1,$ $\phi = 1$

9. **Down-and-out cash-or-nothing** ($S > H$):
Value: $B_2 - B_4$ $\eta = 1,$ $\phi = 1$

10. **Up-and-out cash-or-nothing** ($S < H$):
Value: $B_2 - B_4$ $\eta = -1,$ $\phi = -1$

11. **Down-and-out asset-or-nothing** ($S > H$):
Value: $A_2 - A_4$ $\eta = 1,$ $\phi = 1$

12. **Up-and-out asset-or-nothing** ($S < H$):
Value: $A_2 - A_4$ $\eta = -1,$ $\phi = -1$

13. **Down-and-in cash-or-nothing call** ($S > H$):
Value: $(X > H)$: B_3 $\eta = 1$
Value: $(X < H)$: $B_1 - B_2 + B_4$ $\eta = 1,$ $\phi = 1$

14. **Up-and-in cash-or-nothing call** ($S < H$):
Value: $(X > H)$: B_1 $\phi = 1$
Value: $(X < H)$: $B_2 - B_3 + B_4$ $\eta = -1,$ $\phi = 1$

15. **Down-and-in asset-or-nothing call** ($S > H$):
Value: $(X > H)$: A_3 $\eta = 1$
Value: $(X < H)$: $A_1 - A_2 + A_4$ $\eta = 1,$ $\phi = 1$

16. **Up-and-in asset-or-nothing call** ($S < H$):
Value: $(X > H)$: A_1 $\phi = 1$
Value: $(X < H)$: $A_2 - A_3 + A_4$ $\eta = -1,$ $\phi = 1$

17. Down-and-in cash-or-nothing put ($S > H$):
Value: ($X > H$): $B_2 - B_3 + B_4$ $\eta = 1,$ $\phi = -1$
Value: ($X < H$): B_1 $\phi = -1$

18. Up-and-in cash-or-nothing put ($S < H$):
Value: ($X > H$): $B_1 - B_2 + B_4$ $\eta = -1,$ $\phi = -1$
Value: ($X < H$): B_3 $\phi = -1$

19. Down-and-in asset-or-nothing put ($S > H$):
Value: ($X > H$): $A_2 - A_3 + A_4$ $\eta = 1,$ $\phi = -1$
Value: ($X < H$): A_1 $\phi = -1$

20. Up-and-in asset-or-nothing put ($S < H$):
Value: ($X > H$): $A_1 - A_2 + A_3$ $\eta = -1,$ $\phi = -1$
Value: ($X < H$): A_3 $\phi = -1$

21. Down-and-out cash-or-nothing call ($S > H$):
Value: ($X > H$): $B_1 - B_3$ $\eta = 1,$ $\phi = 1$
Value: ($X < H$): $B_2 - B_4$ $\eta = 1,$ $\phi = 1$

22. Up-and-out cash-or-nothing call ($S < H$):
Value: ($X > H$): 0
Value: ($X < H$): $B_1 - B_2 + B_3 - B_4$ $\eta = -1,$ $\phi = 1$

23. Down-and-out asset-or-nothing call ($S > H$):
Value: ($X > H$): $A_1 - A_3$ $\eta = 1,$ $\phi = 1$
Value: ($X < H$): $A_2 - A_4$ $\eta = 1,$ $\phi = 1$

24. Up-and-out asset-or-nothing call ($S < H$):
Value: ($X > H$): 0
Value: ($X < H$): $A_1 - A_2 + A_3 - A_4$ $\eta = -1,$ $\phi = 1$

25. Down-and-out cash-or-nothing put ($S > H$):
Value: ($X > H$): $B_1 - B_2 + B_3 - B_4$ $\eta = 1,$ $\phi = -1$
Value: ($X < H$): 0

26. Up-and-out cash-or-nothing put ($S < H$):
Value: ($X > H$): $B_2 - B_4$ $\eta = -1,$ $\phi = -1$
Value: ($X < H$): $B_1 - B_3$ $\eta = -1,$ $\phi = -1$

TABLE 4-22

Binary Barrier Option Values

$(H = 100, T = 0.5, r = 0.1, b = 0.1, \sigma = 0.2)$
$K = 15$, except for option numbers (3) and (4) where $K = H$

Option #	S	$X = 102$	$X = 98$	Option #	S	$X = 102$	$X = 98$
(1)	105	9.7264	9.7264	(15)	105	37.2782	45.8530
(2)	95	11.6553	11.6553	(16)	95	44.5294	54.9262
(3)	105	68.0848	68.0848	(17)	105	4.4314	3.1454
(4)	95	11.6553	11.6553	(18)	95	5.3297	3.7704
(5)	105	9.3604	9.3604	(19)	105	27.5644	18.9896
(6)	95	11.2223	11.2223	(20)	95	38.7533	22.7755
(7)	105	64.8426	64.8426	(21)	105	4.8758	4.9081
(8)	95	77.7017	77.7017	(22)	95	0.0000	0.0407
(9)	105	4.9081	4.9081	(23)	105	39.9391	40.1574
(10)	95	3.0461	3.0461	(24)	95	0.0000	0.2676
(11)	105	40.1574	40.1574	(25)	105	0.0323	0.0000
(12)	95	17.2983	17.2983	(26)	95	3.0461	3.0054
(13)	105	4.9289	6.2150	(27)	105	0.2183	0.0000
(14)	95	5.3710	7.4519	(28)	95	17.2983	17.0306

27. Down-and-out asset-or-nothing put $(S > H)$:
Value: $(X > H)$: $A_1 - A_2 + A_3 - A_4$ $\eta = 1,$ $\phi = -1$
Value: $(X < H)$: 0

28. Up-and-out asset-or-nothing put $(S < H)$:
Value: $(X > H)$: $A_2 - A_4$ $\eta = -1,$ $\phi = -1$
Value: $(X < H)$: $A_1 - A_3$ $\eta = -1,$ $\phi = -1$

Table 4-22 gives examples of values for 28 different types of binary barrier options.

4.19.6 Double-Barrier Binary Options

Hui (1996) has published closed-form formulas for the valuation of one-touch double-barrier binary options. A knock-in one-touch double-barrier pays off a cash amount K at maturity if the asset price touches the lower L or upper U barrier before expiration. The option pays off zero if the barriers are not hit during the lifetime of the option. Similarly, a knock-out pays out a predefined cash amount K at maturity if the lower or upper barriers are not hit during the lifetime of the option. If the asset price touches any of the barriers, the option vanishes. The

TABLE 4-23

Double-Barrier Binary Option Values
($S = 100, T = 0.25, r = 0.05, b = 0.03, K = 10$)

L	U	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.5$
80	120	9.8716	8.9307	6.3272	1.9094
85	115	9.7961	7.2300	3.7100	0.4271
90	110	8.9054	3.6752	0.7960	0.0059
95	105	3.6323	0.0911	0.0002	0.0000

formula for the knock-out variant is

$$c = \sum_{i=1}^{\infty} \frac{2\pi i K}{Z^2} \left[\frac{\left(\frac{S}{L}\right)^{\alpha} - (-1)^i \left(\frac{S}{U}\right)^{\alpha}}{\alpha^2 + \left(\frac{i\pi}{Z}\right)^2} \right] \times \sin\left(\frac{i\pi}{Z}\ln(S/L)\right) e^{-\frac{1}{2}\left[\left(\frac{i\pi}{Z}\right)^2 - \beta\right]\sigma^2 T}, \tag{4.89}$$

where

$$Z = \ln(U/L), \qquad \alpha = -\frac{1}{2}\left(\frac{2b}{\sigma^2} - 1\right) \qquad \beta = -\frac{1}{4}\left(\frac{2b}{\sigma^2} - 1\right)^2 - 2\frac{r}{\sigma^2}$$

The option formula is an infinite series. Hui (1996) claims that the series converges very fast and that a few terms are sufficient in most situations. A knock-in can be valued as a short double-barrier binary option plus the cash amount Ke^{-rT}.

Table 4-23 gives examples of values for double-barrier binary options for different choices of barriers and volatility.

4.19.7 Double-Barrier Binary Asymmetrical

This double-barrier binary option is knocked out if the asset price hits the upper barrier U. If the asset price hits the lower barrier L, the option pays out at hit (immediately) the prespecified rebate R—thus the name. Hui (1996) shows that the value is

$$c = R\left(\frac{S}{L}\right)^{\alpha} \left[\sum_{i=1}^{\infty} \frac{2}{i\pi} \left(\frac{\beta - \left(\frac{i\pi}{Z}\right)^2 e^{-\frac{1}{2}\left[\left(\frac{i\pi}{Z}\right)^2 - \beta\right]\sigma^2 T}}{\left(\frac{i\pi}{Z}\right)^2 - \beta} \right) \times \sin\left(\frac{i\pi}{Z}\ln(S/L)\right) + \left(1 - \frac{\ln(S/L)}{Z}\right) \right], \tag{4.90}$$

TABLE 4-24

Double-Barrier Binary Option Values
($S = 100, T = 0.25, r = 0.05, b = 0.03, K = 10$)

L	U	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.5$
80	120	0.0000	0.2402	1.4076	3.8160
85	115	0.0075	0.9910	2.8098	4.6612
90	110	0.2656	2.7954	4.4024	4.9266
95	105	2.6285	4.7523	4.9096	4.9675

where

$$Z = \ln(U/L), \qquad \alpha = -\frac{1}{2}\left(\frac{2b}{\sigma^2} - 1\right) \qquad \beta = -\frac{1}{4}\left(\frac{2b}{\sigma^2} - 1\right)^2 - 2\frac{r}{\sigma^2}$$

Also, this formula contains an infinite series. It exhibits rapid convergence in most cases. Suppose, alternatively, the option is knocked out if the lower barrier is hit and pays out rebate at upper-barrier hit. In that case, the same formula can be used simply by exchanging the two barriers, setting $U = L$ and $L = U$.

Table 4-24 gives examples of values for double-barrier binary, cash at lower barrier, and knock out at upper barrier hit.

4.20 ASIAN OPTIONS

Asian options are especially popular in the energy over the counter (OTC) market and many commodity markets. An average is less volatile than the underlying asset itself and will lower the price of an average-rate option compared with a similar standard option.[12] If the option is based on an average, an attempt to manipulate the asset price just before expiration will normally have little or no effect on the option's value. Asian options should therefore be of particular interest in markets for thinly traded assets.

4.20.1 Geometric Average-Rate Options

If the underlying asset is assumed to be lognormally distributed, the geometric average ($(x_1 \cdots x_n)^{1/n}$) of the asset will itself be lognormally distributed.

[12]If the option is into the average period, the value of an average-rate option can naturally be higher than that of a similar standard option, depending on the realization of the asset price.

Geometric Continuous Average-Rate Options

As originally shown by Kemna and Vorst (1990), the geometric average option can be priced as a standard option by changing the volatility and cost-of-carry term:

$$c = Se^{(b_A - r)T} N(d_1) - Xe^{-rT} N(d_2) \tag{4.91}$$

$$p = Xe^{-rT} N(-d_2) - Se^{(b_A - r)T} N(-d_1), \tag{4.92}$$

where

$$d_1 = \frac{\ln(S/X) + (b_A + \sigma_A^2/2)T}{\sigma_A \sqrt{T}} \qquad d_2 = d_1 - \sigma_A \sqrt{T},$$

and the adjusted volatility is equal to

$$\sigma_A = \frac{\sigma}{\sqrt{3}}$$

Moreover, the adjusted cost-of-carry is set to

$$b_A = \frac{1}{2}\left(b - \frac{\sigma^2}{6}\right).$$

Example

What is the value of a geometric average-rate put option with three months to maturity? The strike is 85, the asset price is 80, the risk-free rate is 5%, the cost-of-carry is 8%, and the volatility is 20%. $S = 80$, $X = 85$, $T = 0.25$, $r = 0.05$, $b = 0.08$, and $\sigma = 0.2$.

$$\sigma_A = \frac{0.2}{\sqrt{3}} = 0.1155 \qquad b_A = \frac{1}{2}\left(0.08 - \frac{0.2^2}{6}\right) = 0.0366$$

$$d_1 = \frac{\ln(80/85) + (0.0366 + 0.1155^2/2)0.25}{0.1155\sqrt{0.25}} = -0.8624$$

$$d_2 = d_1 - 0.1155\sqrt{0.25} = -0.9201$$

$$N(-d_1) = N(0.8624) = 0.8058 \qquad N(-d_2) = N(0.9201) = 0.8213$$

$$p = 85e^{-0.05 \times 0.25} N(-d_2) - 80e^{(0.0366 - 0.05)0.25} N(-d_1) = 4.6922$$

The value of a similar standard European put option is 5.2186.

Geometric Discrete Average-Rate Options

In practice, all Asian options have discrete monitoring of the average. We now show how to value geometric average-rate options with discrete monitoring. We will present the more general case where we can calibrate the geometric average volatility to a term structure of implied volatilities from plain vanilla options. We are thus assuming a spot rate process with time dependent deterministic volatility:

$$dS_t = \mu S_t \, dt + v_t S_t \, dz_t$$

The formula for the geometric average volatility, described in detail by Haug, Haug, and Margrabe (2003), is

$$\sigma_G^2 = \frac{1}{n^3} \sum_{i=0}^{n-1} (n-i)^2 v_i^2 \tag{4.93}$$

where $v_i \equiv v_{t_i}$ is the local volatility between each fixing. For each time step, we need the local volatility. The local implied forward volatilities can be computed from global implied BSM volatilities by the formula

$$v_i = \sqrt{\frac{\sigma_i^2 t_i - \sigma_{i-1}^2 t_{i-1}}{t_i - t_{i-1}}},$$

where σ_i is the implied global volatility for an option expiring at time t_i, and σ_{i-1} is the implied volatility for an option expiring at time $t_{i-1} < t_i$.

Computer algorithm

```
Function GeometricVolFromLocalVolTermStructure(v As Object, n As Long) As Double

    Dim sum As Double
    Dim i As Long

    For i = 0 To n - 1
        sum = sum + v(i + 1)^2 * (n - i)^2
    Next

    GeometricVolFromLocalVolTermStructure = Sqr(sum / n^3)

End Function
```

Alternatively, we can find the Asian geometric volatility directly from the plain vanilla global volatilities shown by Levy (1997):

$$\sigma_G^2 = \frac{1}{n^2 T} \left[\sum_{i=1}^{n} \sigma_i^2 t_i + 2 \sum_{i=1}^{n-1} (n-i) \sigma_i^2 t_i \right], \tag{4.94}$$

where σ_i now is the implied BSM global volatility from an option that expires at t_i, and t_i is the time to fixing i.

Computer algorithm

```
Function GeometricVolFromGlobalVol(T As Double, v As Object, n As Long) As Double

    Dim sum As Double, dt As Double
    Dim i As Long

    dt = T / n

    For i = 1 To n - 1
        sum = sum + v(i)^2 * dt * i + 2 * (n - i) * v(i)^2 * dt * i
    Next
    sum = sum + v(n)^2 * T


    GeometricVolFromGlobalVol = Sqr(sum / (n^2 * T))

End Function
```

Formulas (4.93) and (4.94) both yield the same result; the only difference is that one of them takes local volatilities as input, while the other takes global volatilities as input. As global volatilities are the ones observable in the market, formula (4.94) seems to be the most practical—saving you some calculations.

The value of geometric average options that are calibrated to the term structure can now be computed with the BSM formula:

$$c = Se^{(b_G - r)T} N(d_1) - Xe^{-rT} N(d_2), \tag{4.95}$$

where X is the strike price, $N(\cdot)$ is the cumulative normal distribution,

$$d_1 = \frac{\ln(S/X) + (b_G + \sigma_G^2/2)T}{\sigma_G\sqrt{T}}$$

and

$$d_2 = d_1 - \sigma_G\sqrt{T}$$

This is the BSM formula where we have replaced the volatility with σ_G, and the cost-of-carry with

$$b_G = \frac{\sigma_G^2}{2} + \frac{1}{nT}\sum_{i=1}^{n}(b - \sigma_i^2/2)t_i$$

Here t_i is the time to each fixing, T is the time to maturity, and σ_i is the global BSM volatility for an option with expiration t_i.

Variable Time between Fixings

We have so far assumed equal time between fixings. In real applications the time between fixings can vary. Consider the case of daily fixings. Most markets are closed on weekends, which results in longer time periods over weekends. The formula for calculation of the variance for a geometric average with variable time between fixings is

$$\sigma_G^2 = \frac{1}{n^2 T} \sum_{i=0}^{n-1} (n-i)^2 \Delta t_i v_i^2 \tag{4.96}$$

We have simply assigned a variable time to each fixing, Δt_i.

4.20.2 Arithmetic Average-Rate Options

It is not possible (or very hard) to find a closed-form solution for the value of options on an arithmetic average $\left(\frac{x_1+\cdots+x_n}{n}\right)$.[13] The main reason is that when the asset is assumed to be lognormally distributed, the arithmetic average will not itself have a lognormal distribution. Arithmetic average rate options can be priced by analytical approximations, as presented below, or with Monte Carlo simulations, presented in Chapter 8.

The Turnbull and Wakeman Approximation

The approximation formula below is based on the work of Turnbull and Wakeman (1991). The approximation adjusts the mean and variance so that they are consistent with the exact moments of the arithmetic average. The adjusted mean, b_A, and variance, σ_A^2, are then used as input in the generalized BSM formula:

$$c \approx S e^{(b_A - r)T} N(d_1) - X e^{-rT} N(d_2) \tag{4.97}$$

$$p \approx X e^{-rT} N(d_2) - S e^{(b_A - r)T} N(d_1) \tag{4.98}$$

$$d_1 = \frac{\ln(S/X) + (b_A + \sigma_A^2/2)T}{\sigma_A \sqrt{T}} \qquad d_2 = d_1 - \sigma_A \sqrt{T},$$

where T is the time to maturity in years. The volatility and the cost-of-carry of the average are given by

$$\sigma_A = \sqrt{\frac{\ln(M_2)}{T} - 2b_A}$$

$$b_A = \frac{\ln(M_1)}{T}$$

[13]A possible exception is a recent paper by Linetsky (2004) with a "closed-form solution" for the continuously sampled arithmetic option. One of the solutions requires an infinite series of terms that involves Whittaker functions, and will by most practitioners not be considered "closed form." The method may still serve as a good benchmark to test simpler implementations against.

The exact first and second moments of the arithmetic average are

$$M_1 = \frac{e^{bT} - e^{bt_1}}{b(T - t_1)}$$

$$M_2 = \frac{2e^{(2b+\sigma^2)T}}{(b+\sigma^2)(2b+\sigma^2)(T-t_1)^2} + \frac{2e^{(2b+\sigma^2)t_1}}{b(T-t_1)^2}\left[\frac{1}{2b+\sigma^2} - \frac{e^{b(T-t_1)}}{b+\sigma^2}\right],$$

where in the case of $b = 0$ we have

$$M_1 = 1$$

$$M_2 = \frac{2e^{\sigma^2 T} - 2e^{\sigma^2 t_1}[1 + \sigma^2(T - t_1)]}{\sigma^4(T - t_1)^2},$$

where t_1 is the time to the beginning of the average period. If the option is into the average period, the strike price must be replaced by $\hat{X}$, and the option value must be multiplied by $\frac{T_2}{T}$, where

$$\hat{X} = \frac{T_2}{T}X - \frac{\tau}{T}S_A,$$

where S_A is the average asset price during the realized or observed time period so far. τ is the reminding time in the average period $\tau = T_2 - T$.

If we are into the average period, $\tau > 0$, and $\frac{T_2}{T}X - \frac{\tau}{T}S_A < 0$, then a call option will for certain be exercised and is equal to the expected value of the average at maturity minus the strike price $e^{-rT}(E[S_A] - X)$. The expected average at maturity is equal to[14]

$$E[S_A] = S_A\frac{T_2 - T}{T_2} + SM_1\frac{T}{T_2}$$

The put will in this case for certain not be in-the-money and will have value zero.

Computer algorithm

The computer code below calculates an adjusted cost-of-carry term, b_A, and volatility, v_A, and then calls the general BSM formula described in Chapter 1.

```
Function TurnbullWakemanAsian(CallPutFlag As String, S As Double, _
  SA As Double, X As Double, T As Double, T2 As Double, r As Double, _
  b As Double, v As Double) As Double
```

[14]In a risk-neutral world.

```
' // CallPutFlag = "c" for call and "p" for put option
' // S = Asset price
' // SA= Realized average so far
' // X = Strike price
' // t1 = Time to start of average period in years
' // T =  Time to maturity in years of option  T
' // T2 = Original time in average period in years,
 '// constant over life of option
' // r = risk-free rate
' // b = cost-of-carry underlying asset can be positive and negative
' // v = annualized volatility of asset price


Dim m1 As Double, m2 As Double, tau As Double, t1 As Double
Dim bA As Double, vA As Double

'//tau: reminding time of average perios

t1 = Max(0, T - T2)
tau = T2 - T

If b = 0 Then
    m1 = 1
Else

     m1 = (Exp(b * T) - Exp(b * t1)) / (b * (T - t1))
End If

'//Take into account when option wil  be exercised
If tau > 0 Then

    If T2 / T * X - tau / T * SA < 0 Then

        If CallPutFlag = "c" Then
            ' //Expected average at maturity:
             SA = SA * (T2 - T) / T2 + S * m1 * T / T2
            TurnbullWakemanAsian = Max(0, SA - X) * Exp(-r * T)
        Else
            TurnbullWakemanAsian = 0
        End If
        Exit Function

     End If
End If

'// Extended to hold for options on futures May 16, 1999 Espen G. Haug

If b = 0 Then

   m2 = 2 * Exp(v * v * T) / (v^4 * (T - t1)^2) _
            - 2 * Exp(v * v * t1) * (1 + v * v * (T - t1)) _
            / (v^4 * (T - t1)^2)

Else

    m2 = 2 * Exp((2 * b + v * v) * T) / ((b + v * v) _
             * (2 * b + v * v) * (T - t1)^2)
         + 2 * Exp((2 * b + v * v) * t1) / (b * (T - t1)^2) _
```

```
        * (1 / (2 * b + v * v) - Exp(b * (T - t1)) / (b + v * v))
End If
    bA = Log(m1) / T
    vA = Sqr(Log(m2) / T - 2 * bA)

If tau > 0 Then
    X = T2 / T * X - tau / T * SA

     TurnbullWakemanAsian = GBlackScholes(CallPutFlag, S, X, T, r, _
     bA, vA) * T / T2
Else
    TurnbullWakemanAsian = GBlackScholes(CallPutFlag, S, X, T, r, bA, vA)
End If

End Function
```

Example: *TurnbullWakemanAsian*("*p*", 90, 88, 95, 0, 0.25, 0.25, 0.07, 0.02, 0.25) will return an arithmetic average put value of 5.6093.

Asian Futures Options

In the case of Asian options on futures, only the formulas above can be simplified. If we assume the arithmetic average is approximately lognormally distributed, all we need to value an Asian futures option is to adjust the volatility of the Black-76 formula. This entails replacing the futures volatility σ with the volatility of the average on the futures σ_A:

$$c_A \approx e^{-rT}[FN(d_1) - XN(d_2)] \tag{4.99}$$

$$p_A \approx e^{-rT}[XN(-d_2) - FN(-d_1))], \tag{4.100}$$

where T is the time to maturity, r is the risk-free rate, F is the futures price, and X is the strike price.

$$d_1 = \frac{\ln(F/X) + T\sigma_A^2/2}{\sigma_A\sqrt{T}}, \qquad d_2 = d_1 - \sigma_A\sqrt{T},$$

where

$$\sigma_A = \sqrt{\frac{\ln(M)}{T}} \qquad M = \frac{2e^{\sigma^2 T} - 2e^{\sigma^2\tau}[1 + \sigma^2(T-\tau)]}{\sigma^4(T-\tau)^2},$$

where τ is the time to the beginning of the average period. If the option is into the average period, the strike price must be replaced by $\hat{X}$ and the option value must be multiplied by $\frac{T}{T_2}$, where

$$\hat{X} = X\frac{T_2}{T} - F_A\frac{(T_2 - T)}{T},$$

where T_2 is the original time in the average period and F_A is the average futures price during the realized or observed time period $T_2 - T$.

If $\hat{X}$ should be negative, the call option will for sure be exercised at maturity and the value becomes the discounted value of the expected average at maturity $E_Q[A]$ minus the strike price: $E_Q[A] - X$. The expected average is equal to

$$E_Q[A] = \frac{F_A(T_2 - T)}{T_2} + F\frac{T}{T_2}$$

For a put, the value will be 0 if $\hat{X}$ should be negative. This is basically the Turnbull-Wakeman formula extended to Asian options on futures.

Levy's Approximation
An alternative to the Turnbull and Wakeman formula is the Levy (1992) Asian option approximation:

$$c_{Asian} \approx S_E N(d_1) - X^* e^{-rT_2} N(d_2), \tag{4.101}$$

where

$$S_E = \frac{S}{Tb}(e^{(b-r)T_2} - e^{-rT_2})$$

$$d_1 = \frac{1}{\sqrt{V}}\left[\frac{\ln(D)}{2} - \ln(X^*)\right] \qquad d_2 = d_1 - \sqrt{V}$$

$$X^* = X - \frac{T - T_2}{T} S_A \qquad V = \ln(D) - 2[rT_2 + \ln(S_E)] \qquad D = \frac{M}{T^2}$$

$$M = \frac{2S^2}{b + \sigma^2}\left[\frac{e^{(2b+\sigma^2)T_2} - 1}{2b + \sigma^2} - \frac{e^{bT_2} - 1}{b}\right]$$

The Asian put value can be found by using the following put-call parity:

$$p_{Asian} = c_{Asian} - S_E + X^* e^{-rT_2}$$

where

S_A = Arithmetic average of the known asset price fixings.
S = Asset price.
X = Strike price of option.
r = Risk-free interest rate.
b = Cost-of-carry rate.
T_2 = Remaining time to maturity.
T = Original time to maturity.
σ = Volatility of natural logarithms of return of the underlying asset.

The formula does not allow for $b = 0$. Table 4-25 illustrates this.[15]

[15]For more comparisons between Asian option approximations, see Levy and Turnbull (1992).

TABLE 4-25

Examples of Arithmetic Average Call Option Values

($S = S_A = 100, T_2 = 0.75, r = 0.1, b = 0.05$)

	$\sigma = 0.15$			$\sigma = 0.35$		
X	$T = 0.75$	$T = 0.5$	$T = 0.25$	$T = 0.75$	$T = 0.5$	$T = 0.25$
	Turnbull and Wakeman Approximation					
95	7.0544	5.6731	5.0806	10.1213	6.9705	5.1411
100	3.7845	1.9964	0.6722	7.5038	4.0687	1.4222
105	1.6729	0.3565	0.0004	5.4071	2.1359	0.1552
	Levy's Approximation					
95	7.0544	5.6731	5.0806	10.1213	6.9705	5.1411
100	3.7845	1.9964	0.6722	7.5038	4.0687	1.4222
105	1.6729	0.3565	0.0004	5.4071	2.1359	0.1552

Example

Consider an arithmetic average currency option with a time to expiration of six months. The spot price is 6.80, the strike is 6.90, the domestic risk-free interest rate is 7% per year, the foreign interest rate is 9% per year, and the volatility of the spot rate is 14%. The option is on the average of the next six months. $S = 6.80, S_A = 6.80$, $X = 6.90$, $T = 0.5$, $T_2 = 0.5$, $r = 0.07$, $b = r - r_f = 0.07 - 0.09 = -0.02$, and $\sigma = 0.14$.

$$S_E = \frac{6.8}{0.5(-0.02)}(e^{(-0.02-0.07)\times 0.5} - e^{-0.07\times 0.5}) = 6.5334$$

$$X^* = 6.90 - \frac{0.5 - 0.5}{0.5} 6.80 = 6.9000$$

$$M = \frac{2 \times 6.80^2}{-0.02 + 0.14^2} \times \left[\frac{e^{(2(-0.02)+0.14^2)0.5} - 1}{2(-0.02) + 0.14^2} - \frac{e^{(-0.02)0.5} - 1}{-0.02}\right] = 11.4825$$

$$D = \frac{11.4825}{0.5^2} = 45.9298$$

$$V = \ln(45.9298) - 2[0.07 \times 0.5 + \ln(6.5334)] = 0.0033$$

$$d_1 = \frac{1}{\sqrt{0.0033}}\left[\frac{\ln(45.9298)}{2} - \ln(6.9000)\right] = -0.3146$$

$$d_2 = d_1 - \sqrt{0.0033} = -0.3717$$

$$N(d_1) = N(-0.3146) = 0.3765 \qquad N(d_2) = N(-0.3717) = 0.3551$$

$$c \approx 6.5334N(d_1) - 6.9000e^{-0.07\times 0.5}N(d_2) \approx 0.0944$$

$$p \approx 0.0944 - 6.5334 + 6.9000e^{-0.07\times 0.5} \approx 0.2237$$

4.20.3 Discrete Arithmetic Average-Rate Options

In practice, all traded Asian options have discrete fixings of their average, for instance, every day or week. We next present several approximations for discrete average Asian options. The first method is basically a discrete average version of the Turnbull-Wakeman formula. The next method is the Curran approximation. Both of these implementations assume a flat term structure of volatility for plain vanilla options. In practice, there is often an upward- or downward-sloping volatility term structure. The last method implements a volatility term structure.

Discrete Asian Approximation

The value of a Asian call can be valued as (see Levy, 1997, and Haug, Haug, and Margrabe, 2003)

$$c_A \approx e^{-rT}[F_A N(d_1) - XN(d_2)], \tag{4.102}$$

and the value of a Asian put as

$$p_A \approx e^{-rT} XN(-d_2) - [F_A N(-d_1)], \tag{4.103}$$

where

$$d_1 = \frac{\ln(F_A/X) + T\sigma_A^2/2}{\sigma_A\sqrt{T}}$$

$$d_2 = d_1 - \sigma_A\sqrt{T}$$

F_A is defined as $E[A_T]$, and

$$\sigma_A = \sqrt{\frac{\ln(E[A_T^2]) - 2\ln(E[A_T])}{T}}$$

$$E[A_T] = \frac{S}{n}e^{bt_1}\frac{1 - e^{bhn}}{1 - e^{bh}}$$

and

$$E[A_T^2] = \frac{S^2 e^{(2b+\sigma^2)t_1}}{n^2}\left[\frac{1 - e^{(2b+\sigma^2)hn}}{1 - e^{(2b+\sigma^2)h}} + \frac{2}{1 - e^{(b+\sigma^2)h}}\left(\frac{1 - e^{bhn}}{1 - e^{bh}} - \frac{1 - e^{(2b+\sigma^2)hn}}{1 - e^{(2b+\sigma^2)h}}\right)\right],$$

where $h = \frac{T-t_1}{n-1}$. In the case of $b = 0$ we have

$$E[A_T] = S$$

$$E[A_T^2] = \frac{S^2 e^{\sigma^2 t_1}}{n^2}\left[\frac{1-e^{\sigma^2 hn}}{1-e^{\sigma^2 h}} + \frac{2}{1-e^{\sigma^2 h}}\left(n - \frac{1-e^{\sigma^2 hn}}{1-e^{\sigma^2 h}}\right)\right]$$

If we are inside the average period, $m > 0$, then the strike price should be replaced by

$$X = \frac{nX - mS_A}{n-m}n - \frac{m}{n-m}$$

Moreover, if $S_A > \frac{n}{m}X$, then the exercise is certain for a call, and in the case of a put, it must end up out-of-the-money. So the value of the put must be zero, while the value of the call must be

$$c_A = e^{-rT}(\hat{S}_A - X),$$

where $\hat{S}_A = S_A\frac{m}{n} + E[A]\frac{n-m}{n}$.

If there is only one fixing left to maturity, then the value can be calculated using the BSM formula weighted with time left to maturity and an adjusted strike price. The value of an Asian call option is then

$$c_A = c_{BSM}(S, \hat{X}, T, r, b, \sigma)\frac{1}{n},$$

where c_{BSM} the generalized BSM call formula, and

$$\hat{X} = nX - (n-1)S_A,$$

and S_A is the realized average so far. Similarly, the value of a Asian put with one fixing left is

$$p_A = p_{BSM}(S, \hat{X}, T, r, b, \sigma)\frac{1}{n},$$

where p_{BSM} the generalized BSM put formula.

Table 4-26 gives values for discrete arithmetic average call options. Different choices for time to next average point t_1 and volatility σ are reported.

Computer algorithm

```
Function DiscreteAsianHHM(CallPutFlag As String, S As Double, SA As Double, _
  X As Double, t1 As Double, T As Double, n As Double, m As Double, _
  r As Double, b As Double, v As Double) As Double

    '// This is a modified version of the Levy formula published in
```

TABLE 4-26

Discrete Arithmetic Average Call Option Values
($X = 100, T = 0.5 + t_1, \Delta t = 1/52, r = 0.08, b = 0.03, n = 27, m = 0$)

t_1	S	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$	$\sigma = 0.5$
0 weeks	95	0.2719	1.4166	2.8005	4.2572	5.7480
	100	1.9484	3.4961	5.0557	6.6219	8.1951
	105	5.7150	6.7212	8.0874	9.5713	11.1094
10/52	95	0.8805	2.8800	5.0164	7.1874	9.3708
(10 weeks)	100	2.9570	5.1974	7.4551	9.7156	11.9757
	105	6.5087	8.2935	10.4171	12.6364	14.8936
20/52	95	1.4839	4.0658	6.7249	9.3973	12.0679
(20 weeks)	100	3.7669	6.4983	9.2546	12.0106	14.7590
	105	7.2363	9.5520	12.2008	14.9356	17.6981

```
'//     "Asian Pyramid Power" By Haug, Haug and Margrabe

Dim d1 As Double, d2 As Double, h As Double, EA As Double, EA2 As Double
Dim vA As Double, OptionValue As Double


h = (T - t1) / (n - 1)

If b = 0 Then
    EA = S
Else
    EA = S / n * Exp(b * t1) * (1 - Exp(b * h * n)) / (1 - Exp(b * h))
End If

If m > 0 Then
    '// Exercise is certain for call, put must be out-of-the-money
    If SA > n / m * X Then

        If CallPutFlag = "p" Then
            DiscreteAsianHHM = 0
        ElseIf CallPutFlag = "c" Then
            SA = SA * m / n + EA * (n - m) / n
            DiscreteAsianHHM = (SA - X) * Exp(-r * T)
        End If
        Exit Function

    End If
End If

' // Only one fix left use Black-Scholes weighted with time
If m = n - 1 Then
    X = n * X - (n - 1) * SA
    DiscreteAsianHHM = GBlackScholes(CallPutFlag, S, X, T, r, b, v) _
    * 1 / n
    Exit Function
End If
```

```
    If b = 0 Then

        EA2 = S * S * Exp(v * v * t1) / (n * n) _
           * ((1 - Exp(v * v * h * n)) / (1 - Exp(v * v * h)) _
          + 2 / (1 - Exp(v * v * h))  _
         * (n - (1 - Exp(v * v * h * n)) / (1 - Exp(v * v * h))))
    Else

        EA2 = S * S * Exp((2 * b + v * v) * t1) / (n * n) _
           * ((1 - Exp((2 * b + v * v) * h * n)) _
           / (1 - Exp((2 * b + v * v) * h)) _
           + 2 / (1 - Exp((b + v * v) * h)) * ((1 - Exp(b * h * n)) _
           / (1 - Exp(b * h)) - (1 - Exp((2 * b + v * v) * h * n)) _
           / (1 - Exp((2 * b + v * v) * h))))
    End If

    vA = Sqr((Log(EA2) - 2 * Log(EA)) / T)

    OptionValue = 0

    If m > 0 Then
        X = n / (n - m) * X - m / (n - m) * SA
    End If

    d1 = (Log(EA / X) + vA^2 / 2 * T) / (vA * Sqr(T))
    d2 = d1 - vA * Sqr(T)

    If CallPutFlag = "c" Then
        OptionValue = Exp(-r * T) * (EA * CND(d1) - X * CND(d2))
    ElseIf (CallPutFlag = "p") Then
        OptionValue = Exp(-r * T) * (X * CND(-d2) - EA * CND(-d1))
    End If

    DiscreteAsianHHM = OptionValue * (n - m) / n

End Function
```

Example: *DiscreteAsianHHM*("c",100,110,105,0,0.5,360,180,0.07,0.02,0.25) will return an arithmetic average call value of 2.0971.

Curran's Approximation

Curran (1992) has developed an approximation method for pricing Asian options based on the geometric conditioning approach.[16] Curran (1992) claims that this method is more accurate than other

[16]For more on Asian options valuation, see Geman and Yor (1993), Haykov (1993), Curran (1994), Bouaziz, Briys, and Grouhy (1994), Zhang (1994), Geman and Eydeland (1995), and Zhang (1995b).

closed-form approximations presented earlier.

$$c \approx e^{-rT}\left[\frac{1}{n}\sum_{i=1}^{n} e^{\mu_i+\sigma_i^2/2} N\left(\frac{\mu-\ln(\hat{X})}{\sigma_x}+\frac{\sigma_{xi}}{\sigma_x}\right)\right.$$
$$\left. -XN\left(\frac{\mu-\ln(\hat{X})}{\sigma_x}\right)\right], \qquad (4.104)$$

where

S = Initial asset price.
X = Strike price of option.
r = Risk-free interest rate.
b = Cost-of-carry.
T = Time to expiration in years.
t_1 = Time to first averaging point.
Δt = Time between averaging points.
n = Number of averaging points.
σ = Volatility of asset.
$N(x)$ = The cumulative normal distribution function.

$$\mu_i = \ln(S) + (b - \sigma^2/2)t_i$$

$$\sigma_i = \sqrt{\sigma^2[t_1 + (i-1)\Delta t]}$$

$$\sigma_{xi} = \sigma^2\{t_1 + \Delta t[(i-1) - i(i-1)/(2n)]\}$$

$$\mu = \ln(S) + (b - \sigma^2/2)[t_1 + (n-1)\Delta t/2]$$

$$\sigma_x = \sqrt{\sigma^2[t_1 + \Delta t(n-1)(2n-1)/6n]}$$

and

$$\hat{X} = 2X - \frac{1}{n}\sum_{i=1}^{n}\exp\left\{\mu_i + \frac{\sigma_{xi}[\ln(X)-\mu]}{\sigma_x^2} + \frac{\sigma_i^2 - \sigma_{xi}^2/\sigma_x^2}{2}\right\}$$

If we are inside the average period, $m > 0$, then the strike price should be replaced by

$$X = \frac{nX - mS_A}{n-m}n - \frac{m}{n-m}$$

Further, if $S_A > \frac{n}{m}X$, then exercise is certain for a call, and in the case of a put, it must end up out-of-the-money. So the value of the put must be zero, while the value of the call must be

$$c_A = e^{-rT}(\hat{S}_A - X),$$

where $\hat{S}_A = S_A\frac{m}{n} + E[A]\frac{n-m}{n}$.

TABLE 4-27

Asian Call Options Using the Geometric Conditioning Approach

($X = 100, T = 26$ weeks, $\Delta t = 1$ week, $r = 0.08, b = 0.03, n = 27$)

t_1	S	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$	$\sigma = 0.5$
0	95	0.2758	1.4262	2.8099	4.2581	5.7298
	100	1.9466	3.4899	5.0395	6.5878	8.1320
	105	5.7110	6.7024	8.0489	9.5053	11.0051
10/52	95	0.8819	2.8814	5.0139	7.1753	9.3417
(10 weeks)	100	2.9560	5.1934	7.4443	9.6923	11.9321
	105	6.5066	8.2852	10.3991	12.6029	14.8369
20/52	95	1.4844	4.0655	6.7207	9.3847	12.0409
(20 weeks)	100	3.7661	6.4952	9.2461	11.9920	14.7243
	105	7.2348	9.5466	12.1885	14.9116	17.6564

If there is only one fixing left to maturity, then the value can be calculated using the generalized BSM formula weighted with time left to maturity and an adjusted strike price. The value of an Asian call option is then

$$c_A = c_{BSM}(S, \hat{X}, T, r, b, \sigma)\frac{1}{n},$$

where c_{BSM} is the generalized BSM call formula

$$\hat{X} = nX - (n-1)S_A,$$

and S_A is the realized average so far. Similarly, the value of an Asian put with one fixing left is

$$p_A = p_{BSM}(S, \hat{X}, T, r, b, \sigma)\frac{1}{n},$$

where p_{BSM} the generalized BSM put formula.

Table 4-27 reports Asian option values based on Curran's approximation method.

Computer algorithm

The computer code below calculates the Asian option value using Curran's approximation.

```
Function AsianCurranApprox(CallPutFlag As String, S As Double, _
  SA As Double, X As Double, t1 As Double, T As Double, n As Long, _
  m As Long, r As Double, b As Double, v As Double) As Double

    Dim dt As Double, my As Double, myi As Double
```

```
Dim vxi As Double, vi As Double, vx As Double
Dim Km As Double, sum1 As Double, sum2 As Double
Dim ti As Double, EA As Double
Dim z As Integer, i As Long

z = 1
If CallPutFlag = "p" Then
    z = -1
End If

dt = (T - t1) / (n - 1)

 If b = 0 Then
    EA = S
 Else
    EA = S / n * Exp(b * t1) * (1 - Exp(b * dt * n)) / (1 - Exp(b * dt))
 End If

If m > 0 Then
     If SA > n / m * X Then
     '//Exercise is certain for call, put must be out-of-the-money:
         If CallPutFlag = "p" Then
             AsianCurranApprox = 0
         ElseIf CallPutFlag = "c" Then
             SA = SA * m / n + EA * (n - m) / n
             AsianCurranApprox = (SA - X) * Exp(-r * T)
         End If
         Exit Function
      End If
End If

 If m = n - 1 Then
 ' // Only one fix left use Black-Scholes weighted with time
         X = n * X - (n - 1) * SA
          AsianCurranApprox = GBlackScholes(CallPutFlag, S, X, T, r, b, v) _
          * 1 / n
           Exit Function
 End If

 If m > 0 Then
     X = n / (n - m) * X - m / (n - m) * SA
 End If

 vx = v * Sqr(t1 + dt * (n - 1) * (2 * n - 1) / (6 * n))
 my = Log(S) + (b - v * v * 0.5) * (t1 + (n - 1) * dt / 2)

 sum1 = 0
 For i = 1 To n Step 1

     ti = dt * i + t1 - dt
     vi = v * Sqr(t1 + (i - 1) * dt)
     vxi = v * v * (t1 + dt * ((i - 1) - i * (i - 1) / (2 * n)))
     myi = Log(S) + (b - v * v * 0.5) * ti
     sum1 = sum1 + Exp(myi + vxi / (vx * vx) * _
     (Log(X) - my) + (vi * vi - vxi * vxi / (vx * vx)) * 0.5)
 Next
 Km = 2 * X - 1 / n * sum1
 sum2 = 0
```

```
    For i = 1 To n Step 1

        ti = dt * i + t1 - dt
        vi = v * Sqr(t1 + (i - 1) * dt)
        vxi = v * v * (t1 + dt * ((i - 1) - i * (i - 1) / (2 * n)))
        myi = Log(S) + (b - v * v * 0.5) * ti
        sum2 = sum2 + Exp(myi + vi * vi * 0.5) _
            * CND(z * ((my - Log(Km)) / vx + vxi / vx))

    Next

    AsianCurranApprox = Exp(-r * T) * z * (1 / n * sum2 - X _
    * CND(z * (my - Log(Km)) / vx)) * (n - m) / n

End Function
```

Example: *AsianCurranApprox*("c",100,110,105,0,0.5,360,180,0.07,0.02,0.25) will return an arithmetic average call value of 2.0928.

4.20.4 Equivalence of Floating-Strike and Fixed-Strike Asian Options

We have mainly been looking at how to value what is known as fixed-strike Asian options. In a floating-strike Asian option, the strike is set equal to the average, and a floating-strike call option will at maturity pay out the maximum of the spot price minus the realized average and zero, $\max[S - A, 0]$. Similarly, a floating-strike put will at maturity pay out $\max[A - S, 0]$. One way to find the value of a floating-strike Asian option, or vice versa, is by using what is known as fixed-floating Asian value symmetry, aka fixed-floating Asian Symmetry. Henderson and Wojakowski (2001) describe how to go from the value of a fixed-strike Asian option to a floating-strike Asian option, and vice versa.

$$c_f = (S, 1, T, r, b, \sigma) = p_X(S, S, T, r - b, -b, \sigma), \tag{4.105}$$

where c_f stands for a floating-strike Asian call and p_X stands for fixed-strike Asian put. Similarly, we have

$$c_X = (X, S, T, r, b, \sigma) = p_f\left(S, \frac{X}{S}, T, r - b, -b, \sigma\right) \tag{4.106}$$

This result holds for arithmetic Asian options when we are still not in the average period.

4.20.5 Asian Options with Volatility Term-Structure

Plain vanilla options on the same security but with different time to maturity typically trade at different (implied) volatilities. In other words, we typically observe a nontrivial volatility term structure. The

Asian option formulas mentioned so far assume a flat term structure of volatility. We now describe a more realistic model that can be calibrated to the term structure of plain vanilla option volatilities as described by Haug, Haug, and Margrabe (2003); see also Levy (1997).

The volatility of an arithmetic discrete average, calibrated to the term structure of implied volatilities, can be found as

$$\sigma_A = \sqrt{\frac{\ln(E[A_T^2]) - 2\ln(E[A_T])}{T}}, \tag{4.107}$$

where

$$E[A_T] = \frac{1}{n}\sum_{i=1}^{n} F_i,$$

where F_i is the forward price at fixing i. Moreover,

$$E[A_T^2] = \frac{S^2}{n^2}\sum_{i=1}^{n} e^{(2b+\sigma_i^2)t_i} + 2\sum_{i=1}^{n}\sum_{j=i+1}^{n} e^{(b+\sigma_i^2)t_i}e^{bt_j}$$

σ_i is the plain vanilla BSM volatility for an option with expiration t_i, where t_i is the time to fixing i. Defining $F_A = E[A_T]$, we can now approximate the value of the arithmetic call option as[17]

$$c \approx e^{-rT}[F_A N(d_1) - XN(d_2)] \tag{4.108}$$

and a put option as

$$p \approx e^{-rT}[XN(-d_2) - F_A N(-d_1)], \tag{4.109}$$

where $N(\cdot)$ is the cumulative normal distribution function,

$$d_1 = \frac{\ln(F_A/X) + T\sigma_A^2/2}{\sigma_A\sqrt{T}}$$

and

$$d_2 = d_1 - \sigma_A\sqrt{T}$$

Even if this basically is the Black-76 formula with a modified asset price and volatility, it still holds for Asian options on stocks, stock indexes, and futures.

It is well known that this type of model works best for reasonably low volatilities—for instance, spot volatility less than 30%. However, it is in general far better to use a relatively simple approximation that takes into account the term structure of volatility than using a more accurate model that does not calibrate to the term structure.

[17]The Levy formula is unnecessarily complex. Haug, Haug, and Margrabe (2003) simplify it.

TABLE 4-28

Arithmetic Asian Options with Volatility Term Structure

$(S = 100, t_1 = 1/52, T = 0.5, r = 0.05, b = 0, \sigma = 0.2, n = 26, m = 0)$

	Call Values			Put Values		
X	Flat	Up +0.5%	Down −0.5%	Flat	Up +0.5%	Down −0.5%
80	19.5152	19.5063	19.5885	0.0090	0.0001	0.0823
90	10.1437	9.8313	10.7062	0.3906	0.0782	0.9531
100	3.2700	2.2819	4.3370	3.2700	2.2819	4.3370
110	0.5515	0.1314	1.2429	10.3046	9.8845	10.9960
120	0.0479	0.0016	0.2547	19.5541	19.5078	19.7609

Table 4-28 shows arithmetic Asian option values. The first column is values using 20% flat volatility term structure, while the next column is an upward-sloping term structure. We assume the plain vanilla implied volatility is increasing with 0.5% for every week to maturity. A six-month plain vanilla option thus trades for 20% implied Black–Scholes volatility, and a one-week option trades at 7.5% volatility. The third column is calibrated to a downward-sloping volatility term structure where the plain vanilla volatility for a one-week option trades at 32.5% volatility and a six-month option trades at 20% volatility.

Computer algorithm

```
Function AsianDiscreteTermStructure(CallPutFlag As String, S As Double, _
  SA As Double, X As Double, t1 As Double, T As Double, n As Long, _
  m As Long, r As Double, b As Double, v As Object) As Double

    Dim d1 As Double, d2 As Double, h As Double, EA As Double, EA2 As Double
    Dim vA As Double, OptionValue As Double
    Dim i As Long, j As Long
    Dim sum1 As Double, sum2 As Double


    h = (T - t1) / (n - 1)

   If b = 0 Then
        EA = S
   Else
        EA = S / n * Exp(b * t1) * (1 - Exp(b * h * n)) / (1 - Exp(b * h))
   End If

     If m > 0 Then
        If SA > n / m * X Then
        '// Exercise is certain for call, put must be out-of-the-money

            If CallPutFlag = "p" Then
```

```
                AsianDiscreteTermStructure = 0
            ElseIf CallPutFlag = "c" Then
                SA = SA * m / n + EA * (n - m) / n
                AsianDiscreteTermStructure = (SA - X) * Exp(-r * T)
            End If
            Exit Function
        End If
    End If

    If m = n - 1 Then
      ' // Only one fix left use Black-Scholes weighted with time:
        X = n * X - (n - 1) * SA
        AsianDiscreteTermStructure = GBlackScholes(CallPutFlag, S, X, _
        T, r, b, v(n)) * 1/n
        Exit Function
    End If



    sum1 = 0
    sum2 = 0

    For i = 1 To n - 1
        sum1 = sum1 + Exp((2 * b + v(i)^2) * (t1 + (i - 1) * h))
        For j = i + 1 To n
            sum2 = sum2 + Exp((b + v(i)^2) * (t1 + (i - 1) * h)) _
              * Exp(b * (t1 + (j - 1) * h))
        Next
    Next
    sum1 = sum1 + Exp((2 * b + v(n)^2) * (t1 + (n - 1) * h))


    EA2 = S^2 / (n^2) * (sum1 + 2 * sum2)
    vA = Sqr((Log(EA2) - 2 * Log(EA)) / T)


    If (m > 0) Then
        X = n / (n - m) * X - m / (n - m) * SA
    End If
    d1 = (Log(EA / X) + vA^2 / 2 * T) / (vA * Sqr(T))
    d2 = d1 - vA * Sqr(T)

    If CallPutFlag = "c" Then
        OptionValue = Exp(-r * T) * (EA * CND(d1) - X * CND(d2))
    ElseIf (CallPutFlag = "p") Then
        OptionValue = Exp(-r * T) * (X * CND(-d2) - EA * CND(-d1))
    End If

    AsianDiscreteTermStructure = OptionValue * (n - m) / n

End Function
```

CHAPTER

5

EXOTIC OPTIONS ON TWO ASSETS

Trading exotic options is the most fun you can have with your pants on.

Exotic options trader

In this chapter I present a large class of analytical formulas for exotic options on two or more assets. The exotic option pricing formulas presented here are based on the Black-Scholes-Merton (BSM) economy, meaning that the underlying asset prices are assumed to follow geometric Brownian motions

$$dS_1 = \mu_1 S_1 dt + \sigma_1 S_1 dz$$

$$dS_2 = \mu_2 S_2 dt + \sigma_2 S_2 dw,$$

where μ_1 and μ_2 are the expected instantaneous rates of return of the two assets, and σ_1 and σ_2 are the corresponding instantaneous volatilities. dz and dw are two correlated Wiener processes. The volatility and risk-free rate is assumed to be constant throughout the life of the option. This can in many cases be generalized to deterministic volatility and rates. Most of the formulas are written on a general form that include a cost-of-carry term, which makes it possible to use the same formula to price options on a wide variety of underlying assets: stocks, stock indexes paying a dividend yield, currencies, and futures. Since the formulas are closed-form solutions, they can in general only be used to price European-style options. See Chapter 7, "Trees and Finite Difference Methods," for information on how to value American-style exotic options.

5.1 RELATIVE OUTPERFORMANCE OPTIONS

Relative outperformance options, aka quotient options, are described by Derman (1992) and Zhang (1998). A relative outperformance call gives a payoff $\max\left[\frac{S_1}{S_2} - X, 0\right]$ at maturity. A put similarly pays off

$\max\left[X - \frac{S_1}{S_2}, 0\right]$. The value of a call is

$$c = e^{-rT}[FN(d_2) - XN(d_1)] \tag{5.1}$$

and for a put we have

$$p = e^{-rT}[XN(d_1) - FN(d_2)], \tag{5.2}$$

where $F = \frac{S_1}{S_2} e^{(b_1 - b_2 + \sigma_2^2 - \rho\sigma_1\sigma_2)T}$,

$$d_1 = \frac{\ln(F/X) + T\hat{\sigma}^2/2}{\hat{\sigma}\sqrt{T}}$$

$$d_2 = d_1 - \hat{\sigma}\sqrt{T}$$

$$\hat{\sigma} = \sqrt{\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2}$$

This is basically the Black-76 formula with the forward price and volatility replaced.

Table 5-1 gives examples of relative outperformance option values for different values of strike price X, time to maturity T, and correlation ρ.

TABLE 5-1

Examples of Relative Outperformance Option Values

($S_1 = 130$, $S_2 = 100$, $r = 0.07$, $b_1 = 0.05$, $b_2 = 0.03$, $\sigma_1 = 0.3$, $\sigma_2 = 0.4$)

	$T = 0.25$			$T = 0.5$		
X	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$
0.1	1.2582	1.2380	1.2181	1.3188	1.2769	1.2363
0.2	1.1599	1.1397	1.1198	1.2222	1.1804	1.1398
0.3	1.0616	1.0415	1.0216	1.1257	1.0838	1.0432
0.4	0.9634	0.9432	0.9233	1.0292	0.9873	0.9467
0.5	0.8651	0.8449	0.8250	0.9332	0.8908	0.8501
0.6	0.7671	0.7467	0.7268	0.8386	0.7949	0.7536
0.7	0.6699	0.6487	0.6285	0.7466	0.7004	0.6573
0.8	0.5748	0.5517	0.5303	0.6588	0.6089	0.5619
0.9	0.4840	0.4574	0.4329	0.5764	0.5221	0.4691
1.0	0.3997	0.3686	0.3382	0.5005	0.4417	0.3814
2.0	0.0266	0.0107	0.0013	0.0965	0.0509	0.0132
3.0	0.0010	0.0001	0.0000	0.0170	0.0044	0.0002

TABLE 5-2

Examples of Product Option Call Values

($S_1 = 100, S_2 = 105, X = 15000, r = 0.07, b_1 = 0.02, b_2 = 0.05$)

		$T = 0.1$			$T = 0.5$		
σ_1	σ_2	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$
0.2	0.3	0.0028	0.4289	3.2956	32.6132	154.3380	319.7141
0.3	0.3	0.0267	2.4026	13.2618	56.7733	266.1594	531.7894
0.4	0.3	0.3535	9.3273	35.4908	118.1504	425.9402	787.9742

5.2 PRODUCT OPTIONS

Zhang (1998) describes formulas for product options. A product call has payoff $\max[S_1 S_2 - X, 0]$, while a put pays off $\max[X - S_1 S_2, 0]$. The value of a call is given by

$$c = e^{-rT}[FN(d_2) - XN(d_1)] \tag{5.3}$$

and for a put we have

$$p = e^{-rT}[XN(d_1) - FN(d_2)], \tag{5.4}$$

where $F = S_1 S_2 e^{(b_1+b_2+\rho\sigma_1\sigma_2)T}$,

$$d_1 = \frac{\ln(F/X) + T\hat{\sigma}^2/2}{\hat{\sigma}\sqrt{T}}$$

$$d_2 = d_1 - \hat{\sigma}\sqrt{T}$$

$$\hat{\sigma} = \sqrt{\sigma_1^2 + \sigma_2^2 + 2\rho\sigma_1\sigma_2}$$

Table 5-2 shows product option values, for different values of volatility σ, time to maturity T, and correlation ρ.

5.3 TWO-ASSET CORRELATION OPTIONS

This call option pays off $\max(S_2 - X_2; 0)$ if $S_1 > X_1$ and 0 otherwise. The put pays off $\max(X_2 - S_2)$ if $S_1 < X_1$ and 0 otherwise. These options can be priced using the formulas of Zhang (1995a):

$$c = S_2 e^{(b_2-r)T} M(y_2 + \sigma_2\sqrt{T}, y_1 + \rho\sigma_2\sqrt{T}; \rho) - X_2 e^{-rT} M(y_2, y_1; \rho) \tag{5.5}$$

$$p = X_2 e^{-rT} M(-y_2, -y_1; \rho)$$
$$- S_2 e^{(b_2-r)T} M(-y_2 - \sigma_2\sqrt{T}, -y_1 - \rho\sigma_2\sqrt{T}; \rho), \tag{5.6}$$

where ρ is the correlation coefficient between the returns on the two assets and

$$y_1 = \frac{\ln(S_1/X_1) + (b_1 - \sigma_1^2/2)T}{\sigma_1\sqrt{T}} \qquad y_2 = \frac{\ln(S_2/X_2) + (b_2 - \sigma_2^2/2)T}{\sigma_2\sqrt{T}}$$

Example

Consider a call with six months to expiration. The price of stock A is 52, the price of stock B is 65, the strike price is 50, the payout level is 70, the volatility of stock A is 20%, the volatility of stock B is 30%, the risk-free interest rate is 10%, and the correlation between the two stocks is 0.75. With $S_1 = 52$, $S_2 = 65$, $T = 0.5$, $X_1 = 50$, $X_2 = 70$, $\sigma_1 = 0.2$, $\sigma_2 = 0.3$, $r = 0.1$, $b_1 = b_2 = 0.1$, and $\rho = 0.75$, we get

$$y_1 = \frac{\ln(52/50) + (0.1 - 0.2^2/2)0.5}{0.2\sqrt{0.5}} = 0.5602$$
$$y_2 = \frac{\ln(65/70) + (0.1 - 0.3^2/2)0.5}{0.3\sqrt{0.5}} = -0.2197$$
$$M(y_2 + \sigma_2\sqrt{T}, y_1 + \rho\sigma_2\sqrt{T}; \rho) = 0.4753 \qquad M(y_2, y_1; \rho) = 0.3933$$
$$c = 65e^{(0.1-0.1)0.5} M(y_2 + \sigma_2\sqrt{T}, y_1 + \rho\sigma_2\sqrt{T}; \rho)$$
$$- 70e^{-0.1\times 0.5} M(y_2, y_1; \rho) = 4.7073$$

5.4 EXCHANGE-ONE-ASSET-FOR-ANOTHER OPTIONS

An exchange-one-asset-for-another option, originally introduced by Margrabe (1978), gives the holder the right to exchange asset S_2 for S_1 at expiration. The payoff from an exchange-one-asset-for-another option is

$$\max(Q_1 S_1 - Q_2 S_2; 0),$$

where Q_1 is the quantity of asset S_1 and Q_2 is the quantity of asset S_2. The current value of the option is

$$c_{Exchange} = Q_1 S_1 e^{(b_1-r)T} N(d_1) - Q_2 S_2 e^{(b_2-r)T} N(d_2), \tag{5.7}$$

where

$$d_1 = \frac{\ln(Q_1 S_1/(Q_2 S_2)) + (b_1 - b_2 + \hat{\sigma}^2/2)T}{\hat{\sigma}\sqrt{T}}$$

$$d_2 = d_1 - \hat{\sigma}\sqrt{T}$$

$$\hat{\sigma} = \sqrt{\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2},$$

where ρ is the correlation between the two assets. Table 5-3 shows values for European exchange options. Different values are given for time to maturity T, the volatility of the second asset σ_2, and the correlation between the two assets ρ.

Application

Exchange options are embedded in several financial contracts. One example is when a corporation bids on another corporation by offering its own shares in exchange for the stocks in the takeover candidate. The owners of the takeover candidate receive an option to exchange their stocks for the stocks in the acquiring corporation.

There are also exchange options that are embedded in short bond or note futures contracts. If you are short a bond future, you have to deliver a bond to the counterparty that is long the future at the expiration of the future's contract. However, the counterparty that is short the future can normally choose from a whole class of bonds to deliver. Such bonds can be valued as exchange options.

Example

Consider a European option to exchange bond B for bond A with six months to expiration. Bond A is currently priced at 101, the coupon rate is 8%, and the volatility per year is 18%. Bond B is currently priced at 104, the coupon rate is 6%, and the volatility per year is 12%. The risk-free interest rate is 10%. The correlation between their rates of return is 0.8, $S_1 = 101$, $S_2 = 104$, $T = 0.5$, $r = 0.1$, $b_1 = 0.02$,

TABLE 5-3

Examples of European Exchange Options Values

($S_1 = 22, S_2 = 20, Q_1 = Q_2 = 1, r = 0.1, b_1 = 0.04, b_2 = 0.06, \sigma_1 = 0.2$)

	$T = 0.1$			$T = 0.5$		
σ_2	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$
0.15	2.1251	2.0446	1.9736	2.7619	2.4793	2.1378
0.20	2.1986	2.0913	1.9891	2.9881	2.6496	2.2306
0.25	2.2827	2.1520	2.0189	3.2272	2.8472	2.3736

$b_2 = 0.04$, $\sigma_1 = 0.18$, $\sigma_2 = 0.12$, $\rho = 0.8$. Thus:

$$\hat{\sigma} = \sqrt{0.18^2 + 0.12^2 - 2 \times 0.8 \times 0.18 \times 0.12} = 0.1106$$

$$d_1 = \frac{\ln(101/104) + (0.02 - 0.04 + 0.1106^2/2)0.5}{0.1106\sqrt{0.5}} = -0.4629$$

$$d_2 = d_1 - 0.1106\sqrt{0.5} = -0.5411$$

$$N(d_1) = N(-0.4629) = 0.3217 \qquad N(d_2) = N(-0.5411) = 0.2942$$

$$c = 101e^{(0.02-0.1)0.5}N(d_1) - 104e^{(0.04-0.1)0.5}N(d_2) = 1.5260$$

5.5 AMERICAN EXCHANGE-ONE-ASSET-FOR-ANOTHER OPTION

Bjerksund and Stensland (1993b) show that an American option to exchange asset S_2 for asset S_1 can be simplified to the problem of pricing a standard American call with underlying asset S_1 with risk-adjusted drift equal to $b_1 - b_2$, strike price S_2, time to maturity T, risk-free rate equal to $r - b_2$, and the volatility replaced by $\hat{\sigma}$.[1]

$$c_{Exchange} = C(Q_1S_1, Q_2S_2, T, r - b_2, b_1 - b_2, \hat{\sigma}), \qquad (5.8)$$

where $C(S, X, T, r, b, \sigma)$ is the value of a plain vanilla American call option—for example, the Bjerksund and Stensland (1993a) closed-form approximation. $\hat{\sigma}$ is defined as in the case of the European exchange option. Table 5-4 gives values of an American exchange option using different input parameters. The input parameters are as in Table 5-3 and illustrate that the American-style exchange option is more valuable than its European counterpart.

TABLE 5-4

Examples of American Exchange Options Values
($S_1 = 22$, $S_2 = 20$, $Q_1 = Q_2 = 1$, $r = 0.1$, $b_1 = 0.04$, $b_2 = 0.06$, $\sigma_1 = 0.2$)

	$T = 0.1$			$T = 0.5$		
σ_2	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$
0.15	2.1357	2.0592	2.0001	2.8051	2.5282	2.2053
0.20	2.2074	2.1032	2.0110	3.0288	2.6945	2.2906
0.25	2.2902	2.1618	2.0359	3.2664	2.8893	2.4261

[1] An alternative is to price the option in a binomial tree using the technique described by Rubinstein (1991b).

5.6 EXCHANGE OPTIONS ON EXCHANGE OPTIONS

Exchange options on exchange options can be found embedded in sequential exchange opportunities. An example described by Carr (1988) is a bond holder converting into a stock and later exchanging the shares received for stocks of an acquiring firm. Carr also introduces formulas for pricing this type of rather complex option.

The value of the option to exchange the option to exchange a fixed quantity Q of asset S_2 for the option to exchange asset S_2 for S_1 is

$$c = S_1 e^{(b_1-r)T_2} M(d_1, y_1; \sqrt{t_1/T_2}) - S_2 e^{(b_2-r)T_2} M(d_2, y_2; \sqrt{t_1/T_2}) - QS_2 e^{(b_2-r)t_1} N(d_2) \quad (5.9)$$

The value of the option to exchange asset S_2 for S_1 in return for a fixed quantity Q of asset S_2 is

$$c = S_2 e^{(b_2-r)T_2} M(d_3, y_2; -\sqrt{t_1/T_2}) - S_1 e^{(b_1-r)T_2} M(d_4, y_1; -\sqrt{t_1/T_2}) + QS_2 e^{(b_2-r)t_1} N(d_3), \quad (5.10)$$

where

$$d_1 = \frac{\ln(S_1/(IS_2)) + (b_1 - b_2 + \sigma^2/2)t_1}{\sigma\sqrt{t_1}}, \quad d_2 = d_1 - \sigma\sqrt{t_1}$$

$$d_3 = \frac{\ln(IS_2/S_1) + (b_2 - b_1 + \sigma^2/2)t_1}{\sigma\sqrt{t_1}}, \quad d_4 = d_3 - \sigma\sqrt{t_1}$$

$$y_1 = \frac{\ln(S_1/S_2) + (b_1 - b_2 + \sigma^2/2)T_2}{\sigma\sqrt{T_2}}, \quad y_2 = y_1 - \sigma\sqrt{T_2}$$

$$y_3 = \frac{\ln(S_2/S_1) + (b_2 - b_1 + \sigma^2/2)T_2}{\sigma\sqrt{T_2}}, \quad y_4 = y_3 - \sigma\sqrt{T_2}$$

$$\sigma = \sqrt{\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2}$$

I is the unique critical price ratio $I_1 = \frac{S_1 e^{(b_1-r)(T_2-t_1)}}{S_2 e^{(b_2-r)(T_2-t_1)}}$ solving

$$I_1 N(z_1) - N(z_2) = Q$$

$$z_1 = \frac{\ln(I_1) + (T_2 - t_1)\sigma^2/2}{\sigma\sqrt{T_2 - t_1}}, \quad z_2 = z_1 - \sigma\sqrt{T_2 - t_1},$$

where

S_1 = Asset 1.
S_2 = Asset 2.
t_1 = Time to expiration of the "original" option.

T_2 = Time to expiration of the underlying option ($T_2 > t_1$).
b_1 = Cost-of-carry rate, asset S_1.
b_2 = Cost-of-carry rate, asset S_2.
σ_1 = Volatility of asset S_1.
σ_2 = Volatility of asset S_2.
ρ = Correlation between asset S_1 and S_2.
Q = Quantity of asset delivered if option is exercised.

The value of the option to exchange a fixed quantity Q of asset S_2 for the option to exchange asset S_1 for S_2 is

$$c = S_2 e^{(b_2-r)T_2} M(d_3, y_3; \sqrt{t_1/T_2}) - S_1 e^{(b_1-r)T_2} M(d_4, y_4; \sqrt{t_1/T_2}) - QS_2 e^{(b_2-r)t_1} N(d_3) \tag{5.11}$$

The value of the option to exchange the option to exchange asset S_1 for S_2 in return for a fixed quantity Q of asset S_2 is

$$c = S_1 e^{(b_1-r)T_2} M(d_1, y_4; -\sqrt{t_1/T_2}) - S_2 e^{(b_2-r)T_2} M(d_2, y_3; -\sqrt{t_1/T_2}) + QS_2 e^{(b_2-r)t_1} N(d_2), \tag{5.12}$$

TABLE 5-5

Exchange Options on Exchange Options Values
($S_2 = 100, t_1 = 0.75, T_2 = 1, r = 0.1, b_1 = 0.1, b_2 = 0.1, Q = 0.1$)

		$\rho = -0.5$		$\rho = 0$		$\rho = 0.5$	
σ_1	σ_2	$S_1 = 100$	$S_1 = 105$	$S_1 = 100$	$S_1 = 105$	$S_1 = 100$	$S_1 = 105$
Option to exchange QS_2 for the option to exchange S_2 for S_1							
0.20	0.20	8.5403	10.9076	6.2333	8.4333	3.3923	5.2802
0.20	0.25	10.1756	12.6391	7.5787	9.8819	4.3605	6.3758
0.25	0.20	10.1756	12.6391	7.5787	9.8819	4.3605	6.3758
Option to exchange the option to exchange S_2 for S_1 in return for QS_2							
0.20	0.20	4.7893	4.1747	4.9870	4.2359	5.4267	4.3746
0.20	0.25	4.6948	4.1492	4.8607	4.1961	5.2395	4.3166
0.25	0.20	4.6948	4.1492	4.8607	4.1961	5.2395	4.3166
Option to exchange QS_2 for the option to exchange S_1 for S_2							
0.20	0.20	7.3679	5.9428	5.3005	3.9927	2.7895	1.7663
0.20	0.25	8.8426	7.3670	6.5040	5.1199	3.6392	2.4929
0.25	0.20	8.8426	7.3670	6.5040	5.1199	3.6392	2.4929
Option to exchange the option to exchange S_1 for S_2 in return for QS_2							
0.20	0.20	3.6169	4.2099	4.0542	4.7954	4.8239	5.8607
0.20	0.25	3.3619	3.8771	3.7859	4.4341	4.5182	5.4337
0.25	0.20	3.3619	3.8771	3.7859	4.4341	4.5182	5.4337

where I is now the unique critical price ratio $I_2 = \frac{S_2 e^{(b_2-r)(T_2-t_1)}}{S_1 e^{(b_1-r)(T_2-t_1)}}$ that solves

$$N(z_1) - I_2 N(z_2) = Q$$

$$z_1 = \frac{\ln(I_2) + (T_2 - t_1)\sigma^2/2}{\sigma\sqrt{T_2 - t_1}} \qquad z_2 = z_1 - \sigma\sqrt{T_2 - t_1}$$

Table 5-5 shows values of an exchange option on an exchange option. Values are given for a range of volatilities, correlations, and asset prices.

5.7 OPTIONS ON THE MAXIMUM OR THE MINIMUM OF TWO RISKY ASSETS

Formulas for pricing options on the minimum or maximum of two risky assets were originally introduced by Stulz (1982) and have later been extended and discussed by Johnson (1987); Boyle, Evnine, and Gibbs (1989); Boyle and Tse (1990); Rubinstein (1991d); Rich and Chance (1993); and others.

Call on the Minimum of Two Assets

Payoff: $\max[\min(S_1, S_2) - X, 0]$

$$\begin{aligned} c_{min}(S_1, S_2, X, T) &= S_1 e^{(b_1-r)T} M(y_1, -d; -\rho_1) \\ &+ S_2 e^{(b_2-r)T} M(y_2, d - \sigma\sqrt{T}; -\rho_2) \\ &- X e^{-rT} M(y_1 - \sigma_1\sqrt{T}, y_2 - \sigma_2\sqrt{T}; \rho), \end{aligned} \tag{5.13}$$

where

$$d = \frac{\ln(S_1/S_2) + (b_1 - b_2 + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$y_1 = \frac{\ln(S_1/X) + (b_1 + \sigma_1^2/2)T}{\sigma_1\sqrt{T}} \qquad y_2 = \frac{\ln(S_2/X) + (b_2 + \sigma_2^2/2)T}{\sigma_2\sqrt{T}}$$

$$\sigma = \sqrt{\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2} \qquad \rho_1 = \frac{\sigma_1 - \rho\sigma_2}{\sigma} \qquad \rho_2 = \frac{\sigma_2 - \rho\sigma_1}{\sigma}$$

Call on the Maximum of Two Assets

Payoff: $\max[\max(S_1, S_2) - X, 0]$

$$\begin{aligned} c_{max}(S_1, S_2, X, T) &= S_1 e^{(b_1-r)T} M(y_1, d; \rho_1) \\ &+ S_2 e^{(b_2-r)T} M(y_2, -d + \sigma\sqrt{T}; \rho_2) - X e^{-rT} \\ &\times [1 - M(-y_1 + \sigma_1\sqrt{T}, -y_2 + \sigma_2\sqrt{T}; \rho)] \end{aligned} \tag{5.14}$$

Put on the Minimum of Two Assets
Payoff: $\max[X - \min(S_1, S_2), 0]$

$$p_{min}(S_1, S_2, X, T) = Xe^{-rT} - c_{min}(S_1, S_2, 0, T) + c_{min}(S_1, S_2, X, T), \tag{5.15}$$

where

$$c_{min}(S_1, S_2, 0, T) = S_1 e^{(b_1-r)T} - S_1 e^{(b_1-r)T} N(d) + S_2 e^{(b_2-r)T} N(d - \sigma\sqrt{T})$$

Put on the Maximum of Two Assets
Payoff: $\max[X - \max(S_1, S_2), 0]$,

$$p_{max}(S_1, S_2, X, T) = Xe^{-rT} - c_{max}(S_1, S_2, 0, T) + c_{max}(S_1, S_2, X, T), \tag{5.16}$$

where

$$c_{max}(S_1, S_2, 0, T) = S_2 e^{(b_2-r)T} + S_1 e^{(b_1-r)T} N(d) - S_2 e^{(b_2-r)T} N(d - \sigma\sqrt{T})$$

Example
Consider a put option that gives the holder the right to sell the maximum of stock index A and stock index B at a strike price of 98. Time to maturity is six months, stock index A pays a dividend yield of 6%, stock index B pays a dividend yield of 9%, the price of index A is currently 100, the price of index B is 105, the volatility of index A is 11%, the volatility of index B is 16%, the risk-free interest rate is 5%, and the correlation between the return on the two stock indexes is 0.63. $S_1 = 100, S_2 = 105, X = 98, T = 0.5, r = 0.05$, $b_1 = 0.05 - 0.06 = -0.01, b_2 = 0.05 - 0.09 = -0.04, \sigma_1 = 0.11, \sigma_2 = 0.16$, and $\rho = 0.63$.

$$\sigma = \sqrt{0.11^2 + 0.16^2 - 2 \times 0.63 \times 0.11 \times 0.16} = 0.1246$$

$$\rho_1 = \frac{0.11 - 0.63 \times 0.16}{0.1246} = 0.0738 \qquad \rho_2 = \frac{0.16 - 0.63 \times 0.11}{0.1246} = 0.7280$$

$$d = \frac{\ln(100/105) + (-0.01 + 0.04 + 0.1246^2/2)0.5}{0.1246\sqrt{0.5}} = -0.3395$$

$$y_1 = \frac{\ln(100/98) + (-0.01 + 0.11^2/2)0.5}{0.11\sqrt{0.5}} = 0.2343$$

$$y_2 = \frac{\ln(105/98) + (-0.04 + 0.16^2/2)0.5}{0.16\sqrt{0.5}} = 0.4896$$

$$c_{max}(S_1, S_2, 0, T) = 105e^{(-0.04-0.05)0.5} + 100e^{(-0.01-0.05)0.5}N(d)$$
$$- 105e^{(-0.04-0.05)0.5}N(d - 0.1246\sqrt{0.5}) = 102.4324$$
$$c_{max}(S_1, S_2, X, T) = 100e^{(-0.01-0.05)0.5}M(y_1, d; \rho_1)$$
$$+ 105e^{(-0.04-0.05)0.5}M(y_2, -d + 0.1246\sqrt{0.5}; \rho_2)$$
$$- 98e^{-0.05\times 0.5}[1$$
$$- M(-y_1 + 0.11\sqrt{0.5}, -y_2 + 0.16\sqrt{0.5}; \rho)] = 8.0700$$
$$p_{max}(S_1, S_2, X, T) = 98e^{-0.05\times 0.5} - 102.4324 + 8.0701 = 1.2181$$

Options on the Maximum or the Minimum of Several Assets
For an extension of options on the maximum or the minimum on several assets, we refer to Johnson (1987).

5.8 SPREAD-OPTION APPROXIMATION

The payoff from a European call spread option on two underlying assets is $\max(S_1 - S_2 - X, 0)$. The payoff from a put option is similarly $\max(X - S_1 + S_2, 0)$. A European spread option can be valued using the standard Black and Scholes (1973) model by performing the following transformation, as originally shown by Kirk (1995) (here generalized to options on "any" asset):[2]

$$c = \max(S_1 - S_2 - X, 0) = \max\left(\frac{S_1}{S_2 + X} - 1, 0\right) \times (S_2 + X)$$
$$p = \max(X - S_1 + S_2, 0) = \max\left(1 - \frac{S_1}{S_2 + X}, 0\right) \times (S_2 + X)$$

The value of a call or put is given by

$$c \approx (Q_2 S_2 e^{(b_2-r)T} + Xe^{-rT})[SN(d_1) - N(d_2)] \quad (5.17)$$
$$p \approx (Q_2 S_2 e^{(b_2-r)T} + Xe^{-rT})[N(-d_2) - SN(-d_1)], \quad (5.18)$$

where

$$d_1 = \frac{\ln(S) + (\sigma^2/2)T}{\sigma\sqrt{T}}, \quad d_2 = d_1 - \sigma\sqrt{T}$$
$$S = \frac{Q_1 S_1 e^{(b_1-r)T}}{Q_2 S_2 e^{(b_2-r)T} + Xe^{-rT}},$$

[2]For valuation of spread options see also Boyle (1988), Wilcox (1991), Bjerksund and Stensland (1994), Rubinstein (1994b), Shimko (1994), and Pearson (1995).

and the volatility can be approximated by

$$\sigma \approx \sqrt{\sigma_1^2 + (\sigma_2 F)^2 - 2\rho\sigma_1\sigma_2 F},$$

where

$$F = \frac{Q_2 S_2 e^{(b_2-r)T}}{Q_2 S_2 e^{(b_2-r)T} + X e^{-rT}},$$

where

S_1 = Price on asset one.
S_2 = Price on asset two.
Q_1 = Quantity of asset one.
Q_2 = Quantity of asset two.
X = Strike price.
T = Time to expiration of the option in years.
b_1 = Cost-of-carry asset 1.
b_2 = Cost-of-carry asset 2.
r = Risk-free interest rate.
σ_1 = Volatility of asset 1.
σ_2 = Volatility of asset 2.
ρ = Correlation between the two assets.

Table 5-6 illustrates how call spread option values vary with changes in correlation ρ, volatilities σ_1 and σ_2, and time to maturity T.

Example
Consider a call option on the spread between two futures contracts, with three months to expiration. The price of futures contract 1 is 28, the price of futures contract 2 is 20, the strike is 7, the risk-free interest rate is 5% per year, the volatility of futures 1 is 29% per year, the volatility of futures 2 is 36%, and the correlation between the

TABLE 5-6

Examples of Call Spread Options on Futures Values
($S_1 = 122, S_2 = 120, X = 3, r = 0.1, b_1 = b_2 = 0$)

		$T = 0.1$			$T = 0.5$		
σ_1	σ_2	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$
0.20	0.20	4.7530	3.7970	2.5537	10.7517	8.7020	6.0257
0.25	0.20	5.4275	4.3712	3.0086	12.1941	9.9340	7.0067
0.20	0.25	5.4061	4.3451	2.9723	12.1483	9.8780	6.9284

instantaneous returns on the two futures is 0.42. $S_1 = 28$, $S_2 = 20$, $X = 7$, $T = 0.25$, $r = 0.05$, $b_1 = b_2 = 0$, $\sigma_1 = 0.29$, $\sigma_2 = 0.36$, and $\rho = 0.42$.

$$F = \frac{28e^{(0-0.05)0.25}}{20e^{(0-0.05)0.25} + 7e^{(0-0.05)0.25}} = 1.0370$$

$$\sigma \approx \sqrt{0.29^2 + \left[0.36\frac{20}{20+7}\right]^2 - 2\times 0.42\times 0.29\times 0.36\times\frac{20}{20+7}} = 0.3004$$

$$d_1 = \frac{\ln(1.0370) + (0.3004^2/2)0.25}{0.3004\sqrt{0.25}} = 0.3172$$

$$d_2 = d_1 - 0.3004\sqrt{0.25} = 0.1670$$

$$N(d_1) = N(0.3172) = 0.6245, \quad N(d_2) = N(0.1670) = 0.5663$$

$$c \approx (20+7)\{e^{-0.05\times 0.25}[1.0370N(d_1) - N(d_2)]\} = 2.1670$$

5.9 TWO-ASSET BARRIER OPTIONS

In a two-asset barrier option, one of the underlying assets, S_1, determines how much the option is in- or out-of-the-money, and the other asset, S_2, is linked to barrier hits. Heynen and Kat (1994b) have developed the following pricing formula:

$$w = \eta S_1 e^{(b_1-r)T_2}\left\{M(\eta d_1, \phi e_1; -\eta\phi\rho) - \exp\left[\frac{2(\mu_2+\rho\sigma_1\sigma_2)\ln(H/S_2)}{\sigma_2^2}\right]M(\eta d_3, \phi e_3; -\eta\phi\rho)\right\}$$
$$- \eta X e^{-rT}\left\{M(\eta d_2, \phi e_2; -\eta\phi\rho) - \exp\left[\frac{2\mu_2\ln(H/S_2)}{\sigma_2^2}\right]M(\eta d_4, \phi e_4; -\eta\phi\rho)\right\}, \quad (5.19)$$

where

$$d_1 = \frac{\ln(S_1/X) + (\mu_1 + \sigma_1^2)T}{\sigma_1\sqrt{T}} \qquad d_2 = d_1 - \sigma_1\sqrt{T}$$

$$d_3 = d_1 + \frac{2\rho\ln(H/S_2)}{\sigma_2\sqrt{T}} \qquad d_4 = d_2 + \frac{2\rho\ln(H/S_2)}{\sigma_2\sqrt{T}}$$

$$e_1 = \frac{\ln(H/S_2) - (\mu_2 + \rho\sigma_1\sigma_2)T}{\sigma_2\sqrt{T}} \qquad e_2 = e_1 + \rho\sigma_1\sqrt{T}$$

$$e_3 = e_1 - \frac{2\ln(H/S_2)}{\sigma_2\sqrt{T}} \qquad e_4 = e_2 - \frac{2\ln(H/S_2)}{\sigma_2\sqrt{T}}$$

$$\mu_1 = b_1 - \sigma_1^2/2 \qquad \mu_2 = b_2 - \sigma_2^2/2$$

Two-Asset "Out" Barriers

Down-and-out call (c_{do}) $\eta = 1, \quad \phi = -1$
Payoff: $\max(S_1 - X; 0)$ if $S_2 > H$ before T else 0 at hit

Up-and-out call (c_{uo}) $\eta = 1, \quad \phi = 1$
Payoff: $\max(S_1 - X; 0)$ if $S_2 < H$ before T else 0 at hit

Down-and-out put (p_{do}) $\eta = -1, \quad \phi = -1$
Payoff: $\max(X - S_1; 0)$ if $S_2 > H$ before T else 0 at hit

Up-and-out put (p_{uo}) $\eta = -1, \quad \phi = 1$
Payoff: $\max(X - S_1; 0)$ if $S_2 < H$ before T else 0

Two-Asset "In" Barriers

Down-and-in call $c_{di} = call - c_{do}$
Payoff: $\max(S_1 - X; 0)$ if $S_2 > H$ before T else 0 at expiration.

Up-and-in call $c_{ui} = call - c_{uo}$
Payoff: $\max(S_1 - X; 0)$ if $S_2 < H$ before T else 0 at expiration.

Down-and-in put $p_{di} = put - p_{do}$
Payoff: $\max(X - S_2; 0)$ if $S_2 > H$ before T else 0 at expiration.

Up-and-in put $p_{ui} = put - p_{uo}$
Payoff: $\max(X - S_2; 0)$ if $S_2 < H$ before T else 0 at expiration.

Table 5-7 shows values for two-asset barrier call and put options for different choices of strike price X, barrier H, and the correlation between the two assets ρ.

Application

To illustrate the use of two-asset barrier options, consider a Norwegian oil producer. As oil is typically sold for USD per barrel, the producer's income in Norwegian currency (NOK) depends on not only the oil price but also the currency price NOK per USD. The oil producer may wish to hedge the currency risk by using currency options. Should the oil price (dollar per barrel) increase, however, the producer can

TABLE 5-7

Two-Asset Barrier Option Values
($S_1 = S_2 = 100, T = 0.5, \sigma_1 = \sigma_2 = 0.2, r = 0.08, b_1 = b_2 = 0.08$)

Type	X	H	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$
c_{do}	90	95	3.2941	4.9485	6.6592
c_{do}	100	95	1.4173	2.6150	3.8906
c_{do}	110	95	0.4737	1.1482	1.8949
c_{uo}	90	105	4.6670	3.1827	1.8356
c_{uo}	100	105	2.8198	1.6819	0.7367
c_{uo}	110	105	1.4285	0.7385	0.2263
p_{do}	90	95	0.6184	0.3498	0.1141
p_{do}	100	95	2.0075	1.2821	0.6114
p_{do}	110	95	4.3298	3.0813	1.8816
p_{uo}	90	105	0.0509	0.2250	0.4795
p_{uo}	100	105	0.3042	0.8246	1.4811
p_{uo}	110	105	1.0134	1.9818	3.0712

afford a lower exchange rate. An ideal option would therefore be a currency option that is knocked out if the oil price increases to a particular level. This two-asset barrier option will naturally be cheaper than a similar standard currency option.

5.10 PARTIAL-TIME TWO-ASSET BARRIER OPTIONS

A partial-time two-asset barrier option is similar to a standard two-asset barrier option, except that the barrier hits are monitored only for a fraction of the option's lifetime. The option is knocked "in" or "out" if asset S_2 hits the barrier H during the monitoring period, while the payoff depends on asset S_1 and the strike price X. The formula of Bermin (1996c) can be used to price European partial-time two-asset barrier options, where the barrier monitoring is set to cover the first part t_1 of the full time to expiration T_2:

$$w = \eta S_1 e^{(b_1 - r)T_2} \left\{ M(\eta d_1, \phi e_1; -\eta\phi\rho\sqrt{t_1/T_2}) \right.$$

$$\left. - \exp\left[\frac{2(\mu_2 + \rho\sigma_1\sigma_2)\ln(H/S_2)}{\sigma_2^2}\right] M(\eta d_3, \phi e_3; -\eta\phi\rho\sqrt{t_1/T_2}) \right\}$$

$$- \eta X e^{-rT_2} \Bigg\{ M(\eta d_2, \phi e_2; -\eta\phi\rho\sqrt{t_1/T_2})$$

$$- \exp\left[\frac{2\mu_2 \ln(H/S_2)}{\sigma_2^2}\right] M(\eta d_4, \phi e_4; -\eta\phi\rho\sqrt{t_1/T_2}) \Bigg\}, \tag{5.20}$$

where

$$d_1 = \frac{\ln(S_1/X) + (\mu_1 + \sigma_1^2)T_2}{\sigma_1\sqrt{T_2}} \qquad d_2 = d_1 - \sigma_1\sqrt{T_2}$$

$$d_3 = d_1 + \frac{2\rho \ln(H/S_2)}{\sigma_2\sqrt{T_2}} \qquad d_4 = d_2 + \frac{2\rho \ln(H/S_2)}{\sigma_2\sqrt{T_2}}$$

$$e_1 = \frac{\ln(H/S_2) - (\mu_2 + \rho\sigma_1\sigma_2)t_1}{\sigma_2\sqrt{t_1}} \qquad e_2 = e_1 + \rho\sigma_1\sqrt{t_1}$$

$$e_3 = e_1 - \frac{2\ln(H/S_2)}{\sigma_2\sqrt{t_1}} \qquad e_4 = e_2 - \frac{2\ln(H/S_2)}{\sigma_2\sqrt{t_1}}$$

$$\mu_1 = b_1 - \sigma_1^2/2 \qquad \mu_2 = b_2 - \sigma_2^2/2$$

η is set equal to 1 for a call option and -1 for a put option. ϕ is set equal to 1 if the option is an up-and-out and equal to -1 if the option is a down-and-out.

Table 5-8 gives examples of partial-time two-asset barrier option values. The values are given for a range of input parameters.

TABLE 5-8

Partial-time Two-Asset Barrier Option Values

($S_1 = S_2 = X = 100$, $H = 85$, $T_2 = 1$, $r = b_1 = b_2 = 0.1$, $\sigma_1 = 0.25$, $\sigma_2 = 0.3$)

		Barrier Monitoring Time t_1				
Type	ρ	0	0.25	0.5	0.75	1
c_{do}	0.5	14.9758	12.3793	10.9409	10.2337	9.8185
p_{do}	0.5	5.4595	3.5109	2.3609	1.7502	1.3607
c_{do}	0.0	14.9758	11.2067	8.9828	7.8016	7.0480
p_{do}	0.0	5.4595	4.0855	3.2747	2.8441	2.5694
c_{do}	−0.5	14.9758	9.8818	6.8660	5.2576	4.2271
p_{do}	−0.5	5.4595	4.5801	4.1043	3.8778	3.7497

c_{do} and p_{do} indicate down-and-out call and put, respectively.

5.11 MARGRABE BARRIER OPTIONS

A down-and-in exchange-one-asset-for-another option (down-and-in Margrabe) is knocked in if the ratio of the two asset prices $S = S_1/S_2$ hits the barrier H before maturity. Haug and Haug (2002) derives a closed-form formula for these options:

$$c^e_{di} = S_2 e^{(b_2-r)T}\left(\frac{S}{H}\right)^{1-\frac{2b}{\sigma^2}} c_{BSM}\left(\frac{H^2}{S}, 1, T, 0, b, \sigma\right), \tag{5.21}$$

where c_{BSM} is the BSM call option with current price of underlying S, strike 1, time to maturity T, risk-free rate r, cost-of-carry b, and volatility σ. In the down-and-in Margrabe, $b = b_1 - b_2$. Similarly, the value of an up-and-in Margrabe put option is given by

$$p^e_{ui} = S_2 e^{(b_2-r)T}\left(\frac{S}{H}\right)^{1-\frac{2b}{\sigma^2}} p_{BSM}\left(\frac{H^2}{S}, 1, T, 0, b, \sigma\right), \tag{5.22}$$

where p_{BSM} is the BSM put option.

Table 5-9 shows down-and-in Margrabe values. The first row shows plain vanilla Margrabe prices. It is evident that the barrier option values converge to the Margrabe option values when the barrier converges to the current price ratio $\frac{125}{100} = 1.25$, because the probability of a barrier hit converges to unity.

The value of down-and-out Margrabe options follows from the in-out parity. A down-and-out Margrabe option is simply equal to a long plain Margrabe option plus a short down-and-in Margrabe option. In the special case when $H = 1$ and $b_1 = b_2$ the formula for a down-and-out Margrabe can be simplified to $c^e_{do} = e^{(b_1-r)T}(S_1 - S_2)$. For such parameters this option is "surprisingly" unaffected by volatility and

TABLE 5-9

Down-and-In Margrabe Values

For comparison, the first line shows plain Margrabe values without barriers ($S_1 = 125$, $S_2 = 100$, $T = 1$, $r = 0.07$, $b_1 = 0.03$, $b_2 = 0.05$, $\sigma_1 = 0.45$, $\sigma_2 = 0.47$)

H	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$
	45.8421	40.1508	32.7530
1.2499	45.8318	40.1401	32.7417
1.2000	40.7303	34.9214	27.2639
1.1500	35.7986	29.9309	22.1338
1.1000	31.0790	25.2257	17.4500
1.0500	26.6049	20.8523	13.2922
1.0000	22.4108	16.8553	9.7238

correlation. If $b_1 = b_2 = r$ we naturally get an even simpler formula; $c^e_{do} = S_1 - S_2$, this is the world's simplest two-asset option formula, see Lindset and Persson (2006).

Lastly, consider an up-and-out Margrabe option. Its price is given by

$$c^e_{uo} = S_1 e^{(b_1-r)T}\left[\hat{N}(k_1) - \hat{N}(k_3) - \left(\frac{H}{S}\right)^{1+\frac{2b}{\sigma^2}}\left\{\hat{N}(k_2) - \hat{N}(k_4)\right\}\right]$$

$$- S_2 e^{(b_2-r)T}\left[N(k_1) - N(k_3) - \left(\frac{S}{H}\right)^{1-\frac{2b}{\sigma^2}}\{N(k_2) - N(k_4)\}\right], \qquad (5.23)$$

where $\hat{N}(x) = N(x - \sigma\sqrt{T})$

$$k_1 = \frac{\ln(H/S) - \left(b - \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}} \qquad k_2 = \frac{\ln(S/H) - \left(b - \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}$$

$$k_3 = \frac{-\ln(S) - \left(b - \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}} \qquad k_4 = \frac{\ln(S/H^2) - \left(b - \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}},$$

and as before, $b = b_1 - b_2$. An up-and-in Margrabe can be valued as a plain Margrabe option plus a short up-and-out Margrabe option. Table 5-10 confirms that its price converges to zero as the barrier converges to the current price ratio. Moreover, the price converges

TABLE 5-10

Up-and-Out Margrabe Values

For comparison, the first line shows plain Margrabe values without barrier ($S_1 = 125$, $S_2 = 100$, $T = 1$, $r = 0.07$, $b_1 = 0.03$, $b_2 = 0.05$, $\sigma_1 = 0.45$, $\sigma_2 = 0.47$)

H	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$
	45.8421	40.1508	32.7530
1.2599	0.0025	0.0048	0.0136
1.5000	0.3011	0.5503	1.4583
2.0000	3.1487	5.2288	10.7486
2.5000	8.3667	12.5473	20.3393
3.0000	14.2419	19.5300	26.3218
3.5000	19.7277	25.1056	29.5072
5.0000	31.5188	34.3892	32.3216
25.0000	45.7671	40.1490	32.7530
50.0000	45.8399	40.1508	32.7530

to the price of a plain vanilla Margrabe as the barrier increases. This is also as expected, since the probability of knocking out the option contract diminishes as the barrier increases.

Applications

Margrabe barrier options are relevant to investors and traders that are concerned with the relative performance of stocks, or, for instance, a stock versus a stock index. Moreover, Margrabe (1978) indicated that exchange-one-asset-for-another options are often embedded in financial contracts, and thereby also relevant to corporate finance. An important case is when a firm bids for another firm by offering its own shares in exchange for shares of the target. The owners of the target in effect receive an option to exchange their shares for shares in the acquiring firm. This is only part of the story, however, as many bidding firms use explicit or implicit walk-away covenants that render the offer void if the share price of the target company drops significantly in relation to that of the acquiring firm. In this case, Margrabe barrier options as presented here should be of interest.

5.12 DISCRETE-BARRIER OPTIONS

For discrete-barrier options the barrier can in general be adjusted in the same way as for single-asset barrier options—see Chapter 4, "Exotic Options Single Asset," for more information on this.

5.13 TWO-ASSET CASH-OR-NOTHING OPTIONS

Four types of two-asset cash-or-nothing options exist:

1. A two-asset cash-or-nothing call pays out a fixed cash amount K if asset one, S_1, is above the strike X_1 and asset two, S_2, is above strike X_2 at expiration.
2. A two-asset cash-or-nothing put pays out a fixed cash amount if asset one, S_1, is below the strike X_1 and asset two, S_2, is below strike X_2 at expiration.
3. A two-asset cash-or-nothing up-down pays out a fixed cash amount if asset one, S_1, is above the strike X_1 and asset two, S_2, is below strike X_2 at expiration.
4. A two-asset cash-or-nothing down-up pays out a fixed cash amount if asset one, S_1, is below the strike X_1 and asset two, S_2, is above strike X_2 at expiration.

The formulas published by Heynen and Kat (1996a) can be used to price these binary options:

$$[1] = Ke^{-rT} M(d_{1,1}, d_{2,2}; \rho) \tag{5.24}$$

$$[2] = Ke^{-rT} M(-d_{1,1}, -d_{2,2}; \rho) \tag{5.25}$$

$$[3] = Ke^{-rT} M(d_{1,1}, -d_{2,2}; -\rho) \tag{5.26}$$

$$[4] = Ke^{-rT} M(-d_{1,1}, d_{2,2}; -\rho), \tag{5.27}$$

where

$$d_{i,j} = \frac{\ln(S_i/X_j) + (b_i - \sigma_i^2/2)T}{\sigma_i\sqrt{T}}$$

Table 5-11 reports values for two-asset cash-or-nothing options of types 1, 2, 3, and 4, for different choices of time to maturity T and correlation ρ.

Application

Two-asset cash-or-nothing options can be useful building blocks for constructing more complex exotic option products. One example is a C-Brick option, which pays out a prespecified cash amount K if asset S_1 is between X_1 and X_2 and asset S_2 is between X_3 and X_4. This option can be engineered by using four type [1] two-asset cash-or-nothing call options with different strikes:

$$\text{C-Brick} = Ke^{-rT}[M(d_{1,1}, d_{2,3}; \rho) - M(d_{1,2}, d_{2,3}; \rho) - M(d_{1,1}, d_{2,4}; \rho) + M(d_{1,2}, d_{2,4}; \rho)]$$

In a similar way, Heynen and Kat (1996a) show how to value four types of bivariate asset-or-nothing options.

TABLE 5-11

Two-Asset Cash-or-Nothing Options

($S_1 = S_2 = 100$, $X_1 = 110$, $X_2 = 90$, $K = 10$, $r = 0.1$, $b_1 = 0.05$, $b_2 = 0.06$, $\sigma_1 = 0.2$, $\sigma_2 = 0.25$)

	$T = 0.5$			$T = 1$		
Type	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$
1	1.45845	2.03611	2.49875	1.73130	2.37027	2.94710
2	1.11639	1.69406	2.15669	1.04202	1.68099	2.25782
3	1.25311	0.67545	0.21281	1.63473	0.99576	0.41893
4	5.68434	5.10667	4.64404	4.64032	4.00136	3.42452

5.14 BEST OR WORST CASH-OR-NOTHING OPTIONS

Brockhaus, Farkas, Ferraris, Long, and Overhaus (2000) shows how to value best or worst cash-or-nothing options on two assets. At maturity a best of two asset call option pays out a predefined cash amount K if asset S_1 or S_2 is above or equal to the strike price X. A best of two asset put similarly pays out K at maturity if S_1 or S_2 is below or equal to X. A worst of two asset call option pays out a predefined cash amount if asset S_1 and S_2 is above or equal to the strike. Similarly, a put pays out if both the assets are below or equal to the strike. The value of a best of two asset cash-or-nothing call is

$$c_{best} = Ke^{-rT}[M(y, z_1; -\rho_1) + M(-y, z_2; -\rho_2)] \tag{5.28}$$

$$y = \frac{\ln(S_1/S_2) + (b_1 - b_2 + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$\sigma = \sqrt{\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2}$$

$$z_1 = \frac{\ln(S_1/X) + (b_1 + \sigma_1^2/2)T}{\sigma_1\sqrt{T}}, \qquad z_2 = \frac{\ln(S_2/X) + (b_2 + \sigma_2^2/2)T}{\sigma_2\sqrt{T}}$$

$$\rho_1 = \frac{\sigma_1 - \rho\sigma_2}{\sigma} \qquad \rho_2 = \frac{\sigma_2 - \rho\sigma_1}{\sigma}$$

and for the put

$$p_{best} = Ke^{-rT}[1 - M(y, z_1; -\rho_1) - M(-y, z_2; -\rho_2)] \tag{5.29}$$

The formula for a worst of two assets cash-or-nothing call is

$$c_{worst} = Ke^{-rT}[M(-y, z_1; \rho_1) + M(y, z_2; \rho_2)], \tag{5.30}$$

and the worst of two assets cash-or-nothing put value is given by

$$p_{worst} = Ke^{-rT}[1 - M(-y, z_1; \rho_1) + M(y, z_2; \rho_2)] \tag{5.31}$$

Table 5-12 shows values of best or worst two asset cash-or-nothing options for different choices of strike price X and correlation ρ.

TABLE 5-12

Examples of Best or Worst Two Asset Cash-or-Nothing Options Values

($S_1 = 105$, $S_2 = 100$, $K = 5$, $T = 0.5$, $r = 0.08$, $b_1 = 0.06$, $b_2 = 0.02$, $\sigma_1 = 0.3$, $\sigma_2 = 0.2$)

X	$\rho = -0.75$	$\rho = -0.5$	$\rho = -0.25$	$\rho = 0$	$\rho = 0.25$	$\rho = 0.5$	$\rho = 0.75$
Best of two asset cash-or-nothing call							
90	3.1495	3.2280	3.3131	3.4058	3.5114	3.6394	3.8121
100	1.7329	1.8770	2.0175	2.1652	2.3291	2.5244	2.7939
110	0.7782	0.8659	0.9809	1.1136	1.2661	1.4506	1.7089
Best of two asset cash-or-nothing put							
90	1.6544	1.5759	1.4908	1.3981	1.2925	1.1645	0.9919
100	3.0710	2.9269	2.7865	2.6387	2.4749	2.2796	2.0100
110	4.0258	3.9380	3.8231	3.6904	3.5379	3.3534	3.0950
Worst of two asset cash-or-nothing call							
90	4.7641	4.6856	4.6005	4.5078	4.4022	4.2742	4.1015
100	4.2226	4.0785	3.9381	3.7903	3.6265	3.4312	3.1616
110	3.1055	3.0177	2.9027	2.7701	2.6176	2.4331	2.1747
Worst of two asset cash-or-nothing put							
90	0.0398	0.1183	0.2034	0.2961	0.4018	0.5298	0.7024
100	0.5813	0.7254	0.8659	1.0136	1.1775	1.3728	1.6423
110	1.6985	1.7862	1.9012	2.0339	2.1864	2.3709	2.6292

5.15 OPTIONS ON THE MINIMUM OR MAXIMUM OF TWO AVERAGES

Wu and Zhang (1999) have developed closed-form formulas for European options on the minimum or maximum of two geometric average prices. Assume T is the reminding time to maturity and T_o is the original time to maturity when the option was issued. A call on the minimum of two averages gives the holder at expiration the right to receive $\max[\min(G_1, G_2) - X, 0]$, where G_1 and G_2 are the realized geometric average of, respectively, asset one S_1 and asset two S_2. Its option value is given by

$$c_{MinAsian} = \hat{S}_1 M(a_1, a_5; \rho_1) + \hat{S}_2 M(a_3, a_6; \rho_2) - Xe^{-rT} M(a_2, a_4; \rho), \tag{5.32}$$

where ρ is the correlation between the return on the two assets and

$$a_1 = \frac{\ln(\hat{S}_1/X) + (b_1 + \hat{\sigma}_1{}^2/2)T}{\hat{\sigma}_1\sqrt{T}} \qquad a_2 = a_1 - \hat{\sigma}_1\sqrt{T}$$

$$a_3 = \frac{\ln(\hat{S}_2/X) + (b_2 + \hat{\sigma}_2{}^2/2)T}{\hat{\sigma}_2\sqrt{T}} \qquad a_4 = a_3 - \hat{\sigma}_2\sqrt{T}$$

$$a_5 = \frac{\ln(\hat{S}_2/\hat{S}_1) - T\hat{\sigma}^2/2}{\hat{\sigma}\sqrt{T}} \qquad a_6 = \frac{\ln(\hat{S}_1/\hat{S}_2) - T\hat{\sigma}^2/2}{\hat{\sigma}\sqrt{T}}$$

$$\hat{\sigma_1} = \frac{\sigma_1}{\sqrt{3}}\frac{T}{T_o} \qquad \hat{\sigma_2} = \frac{\sigma_2}{\sqrt{3}}\frac{T}{T_o}$$

$$\hat{\sigma} = \frac{\sigma}{\sqrt{3}}\frac{T}{T_o} \qquad \sigma = \sqrt{\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2}$$

$$\hat{S}_1 = S_1^{T/T_o} e^{(\mu_1 - r)T} G_1$$

$$\hat{S}_2 = S_2^{T/T_o} e^{(\mu_2 - r)T} G_2$$

$$\mu_1 = (b_1 - \sigma_1^2/2)\frac{T}{2T_o} + \frac{1}{6}\sigma_1^2\frac{T^2}{T_o^2}$$

$$\mu_2 = (b_2 - \sigma_2^2/2)\frac{T}{2T_0} + \frac{1}{6}\sigma_2^2\frac{T^2}{T_o^2}$$

$$\rho_1 = \frac{\rho\hat{\sigma_2} - \hat{\sigma_1}}{\hat{\sigma}} \qquad \rho_2 = \frac{\rho\hat{\sigma_1} - \hat{\sigma_2}}{\hat{\sigma}}$$

Even if geometric averages are not used much in practice, the closed-form solution for this case is useful in implementing a variance reduction technique for valuation of arithmetic average options, using Monte Carlo simulation.

The formula above holds for the case where the average covers the option's entire lifetime. Wu and Zhang (1999) discusses how to extend it to a forward starting average.

Table 5-13 gives values for a call on the minimum of two geometric averages for different choices of strike X, time to maturity T, and correlation ρ.

Call on the Maximum of Two Averages

This is equivalent to a long position in two ordinary geometric average call options and a short call on the minimum of two geometric averages.

Put on the Minimum of Two Averages

This is equivalent to a long discount bond with face value X and a short call on the minimum of two averages with zero strike, and a long call on the minimum of two assets with strike X:

$$p_{MinAsian} = Xe^{-rT} - c_{MinAsian}(G_1, G_2, 0) + c_{MinAsian}(G_1, G_2, X) \tag{5.33}$$

TABLE 5-13

Examples of Call on the Minimum of Two Geometric Averages Values

$(S_1 = 100, S_2 = 105, t_1 = 0, r = 0.05, b_1 = b_2 = 0.05, \sigma_1 = 0.3, \sigma_2 = 0.4)$

	$T = 0.25$			$T = 0.5$		
X	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$
80	15.2652	16.4730	17.9827	12.9408	14.6644	16.8381
85	10.5019	11.7079	13.2051	8.7259	10.4274	12.5329
90	6.1880	7.3733	8.8061	5.2044	6.8204	8.7668
95	2.8580	3.9291	5.1737	2.6449	4.0492	5.7199
100	0.9279	1.7007	2.6131	1.1062	2.1627	3.4656
105	0.1919	0.5822	1.1187	0.3714	1.0351	1.9476
110	0.0238	0.1560	0.4044	0.0989	0.4441	1.0165
115	0.0017	0.0327	0.1240	0.0209	0.1714	0.4942
120	0.0001	0.0054	0.0325	0.0035	0.0598	0.2249

Put on the Maximum of Two Averages

A put on the maximum of two averages can be constructed from a long discount bond with face value X, short a call on the maximum of two averages with strike 0, and long a call on the maximum of two averages with strike X:

$$p_{MinAsian} = Xe^{-rT} - c_{MaxAsian}(G_1, G_2, 0) + c_{MaxAsian}(G_1, G_2, X) \tag{5.34}$$

5.16 CURRENCY-TRANSLATED OPTIONS

5.16.1 Foreign Equity Options Struck in Domestic Currency

As the name indicates, these are options on foreign equity where the strike is denominated in domestic currency. At expiration, the foreign equity is translated into the domestic currency.

The payoff to a U.S. investor for an option linked to the Nikkei index is

$$c_{(\$/share)} = \max(E_{(\$/yen)} S^*_{(yen/share)} - X_{(\$/share)}, 0)$$

$$p_{(\$/share)} = \max(X_{(\$/share)} - E_{(\$/yen)} S^*_{(yen/share)}, 0)$$

Valuation of these options is achieved using the formula attributed to Reiner (1992).[3]

[3] Valuation of this option is based on the same technique as developed by Margrabe (1978).

Option Value in Domestic Currency (i.e., USD)

$$c = ES^* e^{-qT} N(d_1) - Xe^{-rT} N(d_2) \tag{5.35}$$

$$p = Xe^{-rT} N(-d_2) - ES^* e^{-qT} N(-d_1), \tag{5.36}$$

where

$$d_1 = \frac{\ln(ES^*/X) + (r - q + \sigma_{ES^*}^2/2)T}{\sigma_{ES^*}\sqrt{T}}$$

$$d_2 = d_1 - \sigma_{ES^*}\sqrt{T}$$

$$\sigma_{ES^*} = \sqrt{\sigma_E^2 + \sigma_{S^*}^2 + 2\rho_{ES^*}\sigma_E\sigma_{S^*}}$$

Option Value in Foreign Currency (i.e., JPY)

$$c = S^* e^{-qT} N(d_1) - E^* Xe^{-rT} N(d_2) \tag{5.37}$$

$$p = E^* Xe^{-rT} N(-d_2) - S^* e^{-qT} N(-d_1), \tag{5.38}$$

where

$$d_1 = \frac{\ln(S^*/(E^*X)) + (r - q + \sigma_{E^*S^*}^2/2)T}{\sigma_{E^*S^*}\sqrt{T}}$$

$$d_2 = d_1 - \sigma_{E^*S^*}\sqrt{T}$$

$$\sigma_{E^*S^*} = \sqrt{\sigma_{E^*}^2 + \sigma_{S^*}^2 - 2\rho_{E^*S^*}\sigma_{E^*}\sigma_{S^*}}$$

S^* = Underlying asset price in foreign currency.
X = Delivery price in domestic currency.
r = Domestic interest rate.
q = Instantaneous proportional dividend payout rate of the underlying asset.
E = Spot exchange rate specified in units of the domestic currency per unit of the foreign currency.
E^* = Spot exchange rate specified in units of the foreign currency per unit of the domestic currency.
σ_{S^*} = Volatility of the underlying asset.
σ_E = Volatility of the domestic exchange rate.
σ_E^* = Volatility of the foreign exchange rate.
ρ_{ES^*} = Correlation between asset and domestic exchange rate.
$\rho_{E^*S^*}$ = Correlation between asset and foreign exchange rate.

Example

Consider a foreign equity call option struck in domestic currency with six months to expiration. The stock index is 100, the strike is 160, the spot exchange rate is 1.5, the domestic interest rate is 8% per year, the dividend yield is 5% per year, the volatility of the stock index is 20%, the volatility of the currency is 12% per year, and the correlation between the stock index and the currency rate is 0.45. $S^* = 100$, $X = 160$, $T = 0.5$, $E = 1.5$, $r = 0.08$, $q = 0.05$, $\sigma_{S^*} = 0.2$, $\sigma_E = 0.12$, $\rho_{E,S^*} = 0.45$. Thus:

$$\sigma_{ES^*} = \sqrt{0.12^2 + 0.2^2 + 2 \times 0.45 \times 0.12 \times 0.2} = 0.2757$$

$$d_1 = \frac{\ln(1.5 \times 100/160) + (0.08 - 0.05 + 0.2757^2/2)0.5}{0.2757\sqrt{0.5}} = -0.1567$$

$$d_2 = d_1 - 0.2757\sqrt{0.5} = -0.3516$$

$$N(d_1) = N(-0.1567) = 0.4378 \quad N(d_2) = N(-0.3516) = 0.3626$$

$$c = 1.5 \times 100e^{-0.05\times 0.5}N(d_1) - 160e^{-0.08\times 0.5}N(d_2) = 8.3056$$

5.16.2 Fixed Exchange Rate Foreign Equity Options

A fixed exchange rate foreign equity option (quanto) is denominated in another currency than that of the underlying equity exposure. The face value of the currency protection expands or contracts to cover changes in the foreign currency value of the underlying asset. Quanto options are traded on stock indexes on several exchanges as well as in the OTC equity market.

The payoff to a U.S. investor for an option linked to the Nikkei index is

$$c_{(\$/share)} = E_{P(\$/yen)} \max(S^*_{(yen/share)} - X^*_{(yen/share)}, 0)$$

$$p_{(\$/share)} = E_{P(\$/yen)} \max(X^*_{(yen/share)} - S^*_{(yen/share)}, 0)$$

Valuation of quanto options was originally introduced by Derman, Karasinski, and Wecker (1990) and was later extended and discussed by Reiner (1992), Dravid, Richardson, and Sun (1993), and others.

Option Value in Domestic Currency (i.e., USD)

$$c = E_p[S^* e^{(r_f - r - q - \rho\sigma_{S^*}\sigma_E)T} N(d_1) - X^* e^{-rT} N(d_2)] \tag{5.39}$$

$$p = E_p[X^* e^{-rT} N(-d_2) - S^* e^{(r_f - r - q - \rho\sigma_{S^*}\sigma_E)T} N(-d_1)] \tag{5.40}$$

Option Value in Foreign Currency (i.e., JPY)

$$c = E^* E_p[S^* e^{(r_f - r - q - \rho\sigma_{S^*}\sigma_E)T} N(d_1) - X^* e^{-rT} N(d_2)] \tag{5.41}$$

$$p = E^* E_p[X^* e^{-rT} N(-d_2) - S^* e^{(r_f - r - q - \rho\sigma_{S^*}\sigma_E)T} N(-d_1)] \tag{5.42}$$

where

$$d_1 = \frac{\ln(S^*/X^*) + (r_f - q - \rho\sigma_{S^*}\sigma_E + \sigma_{S^*}^2/2)T}{\sigma_{S^*}\sqrt{T}}$$

$$d_2 = d_1 - \sigma_{S^*}\sqrt{T}$$

S^* = Underlying asset price in foreign currency.
X^* = Delivery price in foreign currency.
r = Domestic interest rate.
r_f = Foreign interest rate.
q = Instantaneous proportional dividend payout rate of the underlying asset.
E_p = Predetermined exchange rate specified in units of domestic currency per unit of foreign currency.
E^* = Spot exchange rate specified in units of foreign currency per unit of domestic currency.
σ_{S^*} = Volatility of the underlying asset.
σ_E = Volatility of the domestic exchange rate.
ρ = Correlation between asset and domestic exchange rate.

Note that if the exchange rate had been specified in, for example, yen per dollar, the sign of ρ would be reversed: $\rho_{(Nikkei,yen/\$)} = -\rho_{(Nikkei,\$/yen)}$.

Example

Consider a fixed exchange rate foreign equity call option with six months to expiration. The stock index is 100, the strike is 105, the predetermined exchange rate is 1.5, the domestic interest rate is 8% per year, the foreign interest rate is 5% per year, the average dividend yield is 4% per year, the volatility of the stock index is 20%, the volatility of the currency is 10% per year, and the correlation between the stock index and the currency rate is 0.3, $S^* = 100$, $X^* = 105$, $T = 0.5$, $E_p = 1.5$, $r = 0.08$, $r_f = 0.05$, $q = 0.04$, $\sigma_S^* = 0.2$, $\sigma_E = 0.1$, and $\rho = 0.3$. What is the value in domestic currency?

$$d_1 = \frac{\ln(100/105) + (0.05 - 0.04 - 0.3 \times 0.2 \times 0.1 + 0.2^2/2)0.5}{0.2\sqrt{0.5}} = -0.2601$$

$$d_2 = d_1 - 0.2\sqrt{0.5} = -0.4016$$

$$N(d_1) = N(-0.2601) = 0.3974, \quad N(d_2) = N(-0.4016) = 0.3440$$

$$c = 1.5[100e^{(0.05-0.08-0.04-0.3\times0.2\times0.1)0.5}N(d_1)$$
$$- 105e^{-0.08\times0.5}N(d_2)] = 5.3280$$

Computer algorithm

The code returns the value of a call or put quanto option quoted in domestic currency.

```
Function Quanto(CallPutFlag As String, Ep As Double, S As Double, _
  X As Double, T As Double, r As Double, rf As Double, q As Double, _
  vS As Double, vE As Double, rho As Double) As Double

    Dim d1 As Double, d2 As Double

    d1 = (Log(S/X) + (rf - q - rho * vS * vE + vS^2 / 2) * T) / (vS * Sqr(T))
    d2 = d1 - vS * Sqr(T)

    If CallPutFlag = "c" Then
        Quanto = Ep * (S * Exp((rf - r - q - rho * vS * vE) * T) * CND(d1) _
        - X * Exp(-r * T) * CND(d2))
    ElseIf CallPutFlag = "p" Then
        Quanto = Ep * (X * Exp(-r * T) * CND(-d2) _
        - S * Exp((rf - r - q - rho * vS * vE) * T) * CND(-d1))
    End If

End Function
```

5.16.3 Equity Linked Foreign Exchange Options

In an equity linked foreign exchange option, the quantity of the face value is linked to the level of the forward price of a stock or equity index. This is an ideal option for an investor who wants to speculate directly in a foreign equity market but wishes to place a floor on the currency exposure.

The payoff to a U.S. investor for an option linked to the Nikkei index is

$$c_{(\$/share)} = S^*_{(yen/share)}\max(E_{(\$/yen)} - X_{(\$/yen)}, 0)$$

$$p_{(\$/share)} = S^*_{(yen/share)}\max(X_{(\$/yen)} - E_{(\$/yen)}, 0)$$

Valuation of these options has been described by Reiner (1992).

Option Value in Domestic Currency (i.e., USD)

$$c = ES^*e^{-qT}N(d_1) - XS^*e^{(r_f-r-q-\rho\sigma_{S^*}\sigma_E)T}N(d_2) \tag{5.43}$$

$$p = XS^*e^{(r_f-r-q-\rho\sigma_{S^*}\sigma_E)T}N(-d_2) - ES^*e^{-qT}N(-d_1) \tag{5.44}$$

Option Value in Foreign Currency (i.e., JPY)

$$c = S^* e^{-qT} N(d_1) - E^* X S^* e^{(r_f - r - q - \rho\sigma_{S^*}\sigma_E)T} N(d_2) \tag{5.45}$$

$$p = E^* X S^* e^{(r_f - r - q - \rho\sigma_{S^*}\sigma_E)T} N(-d_2) - S^* e^{-qT} N(-d_1) \tag{5.46}$$

where

$$d_1 = \frac{\ln(E/X) + (r - r_f + \rho\sigma_{S^*}\sigma_E + \sigma_E^2/2)T}{\sigma_E\sqrt{T}}$$

$$d_2 = d_1 - \sigma_E\sqrt{T}$$

S^* = Underlying asset price in foreign currency.
X = Currency strike price in domestic currency.
r = Domestic interest rate.
r_f = Foreign interest rate.
q = Instantaneous proportional dividend payout rate of the underlying asset.
E = Spot exchange rate specified in units of the domestic currency per unit of the foreign currency.
E^* = Spot exchange rate specified in units of the foreign currency per unit of the domestic currency.
σ_{S^*} = Volatility of the underlying asset.
σ_E = Volatility of the domestic exchange rate.
ρ = Correlation between asset and the domestic exchange rate.

Example

Consider an equity linked foreign exchange put option with three months to expiration. The stock index is 100, the exchange rate is 1.5, the strike is 1.52, the domestic interest rate is 8% per year, the foreign interest rate is 5% per year, the dividend yield is 4% per year, the volatility of the stock index is 20%, the volatility of the currency is 12% per year, and the correlation between the stock index and the currency rate is −0.4. $S^* = 100$, $E = 1.5$, $X = 1.52$, $T = 0.25$, $r = 0.08$, $r_f = 0.05$, $q = 0.04$, $\sigma_S^* = 0.2$, $\sigma_E = 0.12$, and $\rho = -0.4$. What is the value in domestic currency?

$$d_1 = \frac{\ln(1.5/1.52) + (0.08 - 0.05 - 0.4 \times 0.2 \times 0.12 + 0.12^2/2)0.25}{0.12\sqrt{0.25}} = -0.1057$$

$$d_2 = d_1 - 0.12\sqrt{0.25} = -0.1657$$

$$N(-d_1) = N(0.1057) = 0.5421 \quad N(-d_2) = N(0.1657) = 0.5658$$

$$p = 1.52 \times 100e^{(0.05-0.08-0.04+0.4\times0.2\times0.12)0.25} N(-d_2)$$
$$- 1.50 \times 100e^{-0.04\times0.25} N(-d_1) = 4.2089$$

5.16.4 Takeover Foreign Exchange Options

A takeover foreign exchange call gives the buyer the right to buy N units of a foreign currency at the strike price X if and only if the corporate takeover is successful. A successful takeover is defined as having occurred when the value of the foreign firm V in the foreign currency is less than or equal to the number of currency units B at the option expiration. The value of this option can be found by using the Schnabel and Wei (1994) formula:

$$c = N[Ee^{-r_f T} M(a_2 + \sigma_E\sqrt{T}, -a_1 - \rho\sigma_E\sqrt{T}; -\rho)$$
$$- Xe^{-rT} M(-a_1, a_2; -\rho)], \tag{5.47}$$

where

$$a_1 = \frac{\ln(V/N) + (r_f - \rho\sigma_E\sigma_V - \sigma_V^2/2)T}{\sigma_V\sqrt{T}} \qquad a_2 = \frac{\ln(E/X) + (r - r_f - \sigma_E^2/2)T}{\sigma_E\sqrt{T}}$$

Both the strike price X and the currency price E are quoted in units of the domestic currency per unit of the foreign currency.

5.17 GREEKS FOR TWO-ASSET OPTIONS

To calculate option sensitivities for options on two assets, some analytical partial derivatives are given in some of the papers referred to in this chapter. Alternatively one can use numerical approximations. Any numeric Greek of interest for two-asset options can be calculated using the approach described in Chapter 2. In addition to the Greeks mentioned in that chapter, traders involved in options on two or more assets often consider cross-Greeks, like cross-gamma $\frac{\partial^2 c}{\partial S_1 \partial S_2}$, cross-vanna $\frac{\partial^2 c}{\partial S_1 \partial \sigma_2}$ or $\frac{\partial^2 c}{\partial S_2 \partial \sigma_1}$, cross-vomma $\frac{\partial^2 c}{\partial \sigma_1 \partial \sigma_2}$, cross-zomma $\frac{\partial^3 c}{\partial S_1^2 \partial \sigma_2}$ or $\frac{\partial^3 c}{\partial S_2^2 \partial \sigma_1}$, and cross-speed $\frac{\partial^3 c}{\partial S_1^2 \partial S_2}$ or $\frac{\partial^3 c}{\partial S_2^2 \partial S_1}$. All such cross-Greeks can be calculated by the mixed numerical Greeks given in Chapter 2. The accompanying CD implements all of these and more, using numerical approximations. The included code also implements the ability to plot the Greeks in three dimensions.

CHAPTER

6

BLACK-SCHOLES-MERTON ADJUSTMENTS AND ALTERNATIVES

Life is just one damn trade after another.

Trader

A model is only a simplified representation of an aspect of the real world. Models are typically developed by starting out with very simple representations that are later generalized to accommodate more detail. In this chapter we loosen up many of the assumptions originally made by Black and Scholes, and Merton back in 1973, and consider how it affects the option value. We first consider small adjustments of the Black-Scholes-Merton (BSM) formula, that take into account the following aspects:

- Delayed settlement
- Trading-day volatility versus calendar day volatility
- Discrete time hedging
- Transaction cost
- Trending markets

We will subsequently move on to alternative models that take into account non-normal distributed returns such as:

- Generalized constant elasticity of variance
- Skewness and Kurtosis
- Merton's jump-diffusion model
- Bates' generalized jump-diffusion model
- The Hull-White 1987 stochastic volatility model
- The Hull-White 1988 stochastic volatility model
- SABR 2001 stochastic volatility model

6.1 THE BLACK-SCHOLES-MERTON MODEL WITH DELAYED SETTLEMENT

The original BSM formula assumes one pays for the option immediately or at least the same day as one buys the option—immediate settlement. In practice, this is often not the case. In many markets like foreign exchange or for OTC bond options, one typically pays (gets paid) two to three days after entering into the agreement to buy (sell). This leads to a simple adjustment that can be economically significant in some market situations:

$$c = e^{r_1 t_1} e^{-r_2 T_2}[Se^{bT} N(d_1) - XN(d_2)] \tag{6.1}$$

$$p = e^{r_1 t_1} e^{-r_2 T_2}[XN(-d_2) - Se^{bT} N(-d_1)] \tag{6.2}$$

$$d_1 = \frac{\ln(S/X) + (b + \sigma^2/2)T}{\sigma\sqrt{T}} \qquad d_2 = d_1 - \sigma\sqrt{T},$$

where t_1 is the time in years until payment for the option ($t_1 <= T$). T_2 is the time in years until settlement of the option's payoff if in-the-money at maturity ($T_2 \geq T$). T is, as usual, the option's number of years until maturity. Taking into account delays of settlement can be important in circumstances where the short-term rate is very high compared to long-term rates. Setting $t_1 = 0$, $T_2 = T$, and $r_1 = r_2$ yields the standard BSM formula, with instant settlement.

Example

Consider a call option with a strike price of 120 and six months to expiration. Settlement is two days, and the underlying stock trades at 130 and has a volatility of 12%. The short continuous zero coupon rate with two days to maturity is 50%, while the rate until expiration settlement is 10%. Also, the rate to option expiration is 10%. With $S = 130$, $X = 120$, $t_1 = 2/365 = 0.0055$, $T_2 = 0.5 + 2/365 = 0.5055$, $T = 0.5$, $r_1 = 0.5$, $r_2 = 0.1$, $b = 0.1$, and $\sigma = 0.12$ we get

$$d_1 = \frac{\ln(130/120) + (0.1 + 0.12^2/2)0.5}{0.12\sqrt{0.5}} = 1.5750$$

$$d_2 = 1.5209 - 0.12\sqrt{0.5} = 1.4901$$

$$N(d_1) = N(1.5750) = 0.9424 \qquad N(d_2) = N(1.4901) = 0.9319$$

$$c = e^{0.5\times 0.0055} e^{-0.1\times 0.5055}[130e^{0.1\times 0.5} N(d_1) - 120N(d_2)] = 16.1688$$

The value of 16.1688 compares to 16.1334 using the standard BSM formula. For a large position, this difference can represent a nontrivial amount of money.

6.2 THE BLACK-SCHOLES-MERTON MODEL ADJUSTED FOR TRADING DAY VOLATILITY

The formula of French (1984) takes into consideration that volatility is usually higher on trading days than on nontrading days.[1]

$$c = SN(d_1) - Xe^{-rT}N(d_2) \qquad (6.3)$$

$$p = Xe^{-rT}N(-d_2) - SN(-d_1) \qquad (6.4)$$

where

$$d_1 = \frac{\ln(S/X) + rT + \sigma^2 t/2}{\sigma\sqrt{t}}$$

$$d_2 = \frac{\ln(S/X) + rT - \sigma^2 t/2}{\sigma\sqrt{t}} = d_1 - \sigma\sqrt{t}$$

$$t = \frac{\text{Trading days until maturity}}{\text{Trading days per year}}$$

and

$$T = \frac{\text{Calendar days until maturity}}{\text{Calendar days per year}}$$

Example

Consider a European put option with 146 calendar days and 104 trading days to expiration, and assume there are 365 calendar days and 253 trading days per year. The stock price is 70, the strike price is 75, the risk-free interest rate is 8% per year, and the volatility is 30% per year. $S = 70$, $X = 75$, $T = 0.4(146/365)$, $t = 0.4111(104/253)$, $r = 0.08$, and $\sigma = 0.3$. Thus:

$$d_1 = \frac{\ln(70/75) + 0.08 \times 0.4 + 0.3^2 \times 0.4111/2}{0.3\sqrt{0.4111}} = -0.0962$$

$$d_2 = d_1 - 0.3\sqrt{0.4111} = -0.2885$$

$$N(-d_1) = N(0.0962) = 0.5383 \qquad N(-d_2) = N(0.2885) = 0.6135$$

$$p = 75^{-0.08\times 0.4}N(-d_2) - 70N(-d_1) = 6.8836$$

---

[1]This has been supported by several empirical studies, for example, French (1980), Gibbons and Hess (1981), and French and Roll (1986).

6.3 DISCRETE HEDGING

The main idea of Black, Scholes, and Merton was that by holding an option and continuously delta hedging it, one removes all of the portfolio's market risk. The expected return on the portfolio should therefore be equal to the risk-free rate. It is not possible, however, to delta hedge in continuous time.[2] In the case of discrete-time delta hedging, one has to be careful, as the BSM risk-neutral valuation argument no longer holds. We will here look at some insights into how discrete-time hedging causes hedging errors, and how to value and optimally hedge options with discrete-time hedging.

6.3.1 Hedging Error

A discrete-time hedge is imperfect and, consequently, the dynamic portfolio is no longer risk-free. Derman and Kamal (1999) have published a couple of useful formulas that give an idea of some of the effects from discrete-time hedging. They assume geometric Brownian motion and delta hedging using the BSM delta updated at discrete, evenly spaced time intervals. The standard deviation of the final profit and loss (P&L) given N number of rebalances during the option's lifetime is given by the following approximation:

$$\sigma_{\text{P\&L}} \approx \sqrt{\frac{\pi}{4}} S e^{(b-r)T} n(d_1)\sqrt{T}\frac{\sigma}{\sqrt{N}} = \sqrt{\frac{\pi}{4}} \text{Vega}\frac{\sigma}{\sqrt{N}} \tag{6.5}$$

where Vega is the sensitivity with respect to changes in volatility. If the option is approximately at-the-money forward ($Se^{bT} = X$), the expression can be simplified. In that case, the option value is approximately linear with respect to changes in volatility. We can then express the standard deviation in P&L in percent of the initial option premium as

$$\frac{\sigma_{\text{P\&L}}}{c} \approx \sqrt{\frac{\pi}{4}}\frac{1}{\sqrt{N}} = \sqrt{\frac{\pi}{4N}} \tag{6.6}$$

where c is the initial option price. The right-hand side of this equation now depends only on the number of rebalances. The formulas above assume constant volatility, while in practice one observes stochastic volatility. With this in mind, the formulas above can be seen as a lower boundary for the standard deviation in P&L for an option being hedged in the market. Notice, moreover, that one typically does not

[2] Even if one did manage to update the portfolio continuously in time, the trading activity would incur infinite transactions costs.

rebalance the hedge at even intervals. The decision to rebalance is typically based on the values of the option's delta and gamma. A higher gamma generally necessitates more frequent rebalancing.

Example

Consider a futures option with three months to expiration. The stock price is 60, the strike price is 60, the risk-free interest rate is 6% per year, and the volatility is 30% per year. Assuming we rebalance the hedge 20 times, what is the standard deviation in P&L? $S = 60$, $X = 60$, $T = 0.25$, $r = 0.06$, $b = 0$, $\sigma = 0.3$, and $N = 20$. Given these parameters, the vega of the option is

$$d_1 = \frac{\ln(60/60) + (0.3^2/2)0.25}{0.3\sqrt{0.25}} = 0.0750$$

$$n(d_1) = n(0.0750) = 0.3978$$

$$\text{vega} = 60e^{(0-0.06)0.25} \times 0.3978\sqrt{0.25} = 11.7570$$

The standard deviation in terms of dollars is then given by

$$\sigma_{\text{P\&L}} \approx \sqrt{\frac{\pi}{4}} \times 11.7570 \times \frac{0.3}{\sqrt{20}} = 0.6990$$

The one standard deviation in our expected profit and loss is thus ± 0.6990 dollars. This is independent of a call or a put option.

Without going through the calculations, we can say the call or put option[3] value with these parameters is 3.5337. This gives us a standard deviation in percent of option value of $0.6990/3.5337 = 19.78\%$. We could alternatively have calculated this using formula (6.6):

$$\frac{\sigma_{\text{P\&L}}}{c} \approx \sqrt{\frac{\pi}{4 \times 20}} = 19.82\%$$

As pointed out earlier, stochastic volatility, jumps, and other real market effects will in general increase the standard deviation of the profit and loss from the discrete hedge.

6.3.2 Discrete-Time Option Valuation and Delta Hedging

We have seen how discrete-time hedging causes hedging errors. Although there is no way we can eliminate all risk for discrete-time hedging in the BSM economy, we can find a better hedge than the

[3]From Chapter 1 we know an option in the Black-Scholes-Merton world where $Se^{bT} = X$ must have value symmetry—that is, the same value for call and put options.

BSM delta. One intuitive criteria is that the hedge should minimize the variance of profit/loss over the next time step Δt. By identifying the delta that satisfies this criterion, we will also be able to find the fair price of the option contract, taking into account discrete hedging. Assuming hedging takes place at regular times Δt apart, Wilmott (2000) shows that the optimal delta is

$$\Delta_{\text{option}} = \frac{\partial c}{\partial S} + \Delta t(\mu - r + \sigma^2/2)S\frac{\partial^2 c}{\partial S^2} \tag{6.7}$$

This expression shows that the optimal delta consists of the standard delta $\frac{\partial c}{\partial S}$ plus a correction term multiplied by the option's standard gamma $\frac{\partial^2 c}{\partial S^2}$. An important point to notice is that the delta now depends also on the expected growth rate of the asset μ. It is therefore clear that the idea of risk-neutral valuation does not hold for discrete-time hedging. If μ is large, the discrete-time delta can be significantly different from the standard risk-neutral BSM delta.

Based on the optimal delta, Wilmott (2000) shows how we can still value the option with the BSM formula, but now with an adjusted volatility equal to

$$\hat{\sigma} = \sigma\left(1 + \frac{\Delta t}{2\sigma^2}(\mu - r)(r - \mu - \sigma^2)\right) \tag{6.8}$$

Again, the expected growth rate of the asset turns up.

Example

Consider an option that we delta hedge once a week. Assume the expected volatility of the asset is 50%, the risk-free rate is 5%, and the expected growth rate of the stock is 20%. What is the volatility we should use in the BSM formula to value the option? $\Delta t = \frac{1}{52}$, $\sigma = 0.5$, $r = 0.05$, and $\mu = 0.2$ yields

$$\hat{\sigma} = \sigma\left(1 + \frac{\frac{1}{52}}{2 \times 0.5^2}(0.2 - 0.05)(0.05 - 0.2 - 0.5^2)\right) = 49.88\%$$

An adjustment of 0.12 percentage points to the volatility may seem trivial. In a competitive options market, this may still be of economic consequence.

6.3.3 Discrete-Time Hedging with Transaction Cost

Leland (1985) looks at how hedging with transactions costs affects the BSM formula, assuming the hedge is rebalanced at evenly spaced time intervals Δt apart. He moreover assumes that transactions cost are proportional to the value of the transaction in the underlying shares. If n shares are bought ($n > 0$) or sold ($n < 0$), then the transaction cost is $\kappa|n|S$, where κ is some positive constant. Leland finds that a long option can be valued with the BSM formula but now with an adjusted volatility of

$$\sigma_{\text{long}} = \sigma \left(1 - \frac{\kappa}{\sigma}\sqrt{\frac{8}{\pi \Delta t}}\right)^{\frac{1}{2}}, \tag{6.9}$$

and in the case of short options the volatility adjustment is

$$\sigma_{\text{short}} = \sigma \left(1 + \frac{\kappa}{\sigma}\sqrt{\frac{8}{\pi \Delta t}}\right)^{\frac{1}{2}} \tag{6.10}$$

It is worth mentioning that the approach of Leland (1985) holds only for plain vanilla calls and puts. It does in general not hold for option portfolios or exotic options. Further, Kabanov and Safarian (1997) shows that the Leland approach has several problems in the case of a constant level of transaction costs. See also Grandits and Schachinger (2001).

Hoggard, Whalley, and Wilmott (1994) look at discrete hedging for more general cases. See also Wilmott (2000) for more details on transaction costs and derivatives valuation.

Example

Assume the transaction costs are 0.1% of the stock value, that we rebalance the hedge daily, and that the expected volatility of the stock is 30%. What volatility should be used to price plain vanilla options? With $\sigma = 0.3$, $\kappa = 0.001$, and $\Delta t = \frac{1}{365}$, the volatility we should use for long options is

$$\sigma_{\text{long}} = 0.3 \left(1 - \frac{0.001}{0.3}\sqrt{\frac{8}{\pi \frac{1}{365}}}\right)^{\frac{1}{2}} = 28.43\%,$$

while for short options we should use

$$\sigma_{\text{short}} = 0.3 \left(1 + \frac{0.001}{0.3}\sqrt{\frac{8}{\pi \frac{1}{365}}}\right)^{\frac{1}{2}} = 31.49\%$$

These volatilities are then to be used as input in the BSM formula.

6.4 OPTION PRICING IN TRENDING MARKETS

Many different and complex price processes fit into the category of "trending" or "predictable" markets. Lo and Wang (1995) look at the special case of a trending Ornstein-Uhlenbeck process that implies serially correlated asset returns.

Lo and Wang shows how to adjust the BSM formula to take into account the trending market:

$$c = Se^{(b-r)T}N(d_1) - Xe^{-rT}N(d_2) \tag{6.11}$$

$$p = Xe^{-rT}N(-d_2) - Se^{(b-r)T}N(-d_1), \tag{6.12}$$

where

$$d_1 = \frac{\ln(S/X) + (b + \hat{\sigma}^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \hat{\sigma}\sqrt{T}$$

and

$$\hat{\sigma} = \sqrt{\frac{\sigma^2}{\Delta t}\frac{\ln(1+2\rho)}{(1+2\rho)^{1/\Delta t} - 1}}$$

The adjustment for autocorrelation in the returns is done through the input volatility, $\hat{\sigma}$, where σ is the unconditional volatility estimate and $\rho \in (-\frac{1}{2}, 0]$ is the first-order autocorrelation over the holding period Δt which in turn is measured in units of the holding period. Since ρ is restricted to the interval -0.5 to zero, the adjustment works for negative autocorrelation. The adjustment for drift in the volatility is related to the fact that volatility has to be estimated based on discretely sampled data, while the BSM model is based on continuous delta hedging. Even if the true continuous volatility was constant, as first assumed by Black and Scholes, the sampling error from any estimate will be related to two sources: the discreteness of the sampling interval and misspecifications of the drift. As we increase the sampling interval, both of these sources of sampling error will diminish. As the holding period Δt goes to zero, the autocorrelation effect on the volatility approaches zero.

An increase in the absolute value of the autocorrelation will always increase the adjusted, volatility. For this reason, it should not come as a surprise that the options sensitivity to autocorrelation is also related to the vega from the BSM formula. An increase in the absolute value of autocorrelation always increases the option value, as the vega is always positive and since we consider only negative autocorrelation. This holds for the trending Ornstein-Uhlenbeck process as well as for other types of trending processes.

TABLE 6-1

Call Option Values under a Trending Ornstein-Uhlenbeck Process

($S = 100, \sigma = 0.3, r = 0.08, b = 0.08, \Delta t = 1$)

X	BSM	$\rho = -0.05$	$\rho = -0.1$	$\rho = --0.2$	$\rho = -0.3$	$\rho - 0.4$	$\rho - 0.45$
			Time to Maturity 7 days				
80	20.1226	20.1226	20.1226	20.1226	20.1227	20.1227	20.1232
90	10.1443	10.1459	10.1481	10.1554	10.1710	10.2149	10.2814
100	1.7338	1.7776	1.8269	1.9491	2.1240	2.4262	2.7259
110	0.0181	0.0219	0.0268	0.0421	0.0718	0.1464	0.2490
120	0.0000	0.0000	0.0000	0.0001	0.0003	0.0019	0.0072
			Time to Maturity 182 days				
80	24.0808	24.1728	24.2807	24.5655	25.0132	25.8767	26.8203
90	16.3942	16.5585	16.7461	17.2193	17.9159	19.1585	20.4257
100	10.3714	10.5856	10.8271	11.4255	12.2834	13.7661	15.2361
110	6.1206	6.3416	6.5912	7.2101	8.0993	9.6399	11.1704
120	3.3945	3.5860	3.8044	4.3549	5.1645	6.6068	8.0758
			Time to Maturity 365 days				
80	28.2411	28.4046	28.5938	29.0822	29.8248	31.2010	32.6510
90	21.3720	21.6088	21.8783	22.5553	23.5454	25.2968	27.0673
100	15.7113	16.0018	16.3296	17.1423	18.3087	20.3258	22.3244
110	11.2596	11.5746	11.9295	12.8067	14.0604	16.2177	18.3457
120	7.8966	8.2078	8.5595	9.4331	10.6903	12.8716	15.0392

Table 6-1 shows call option values that are adjusted according to the above recipe. The column marked BSM is option values calculated with the Black-Scholes-Merton formula. As we can see from the table, the values adjusted for autocorrelation are always larger than (or equal to) the BSM values.

Example

Consider a one-year call option on a stock currently trading at 100. The volatility is 30%, the risk-free rate is 8%, the strike is 120, and the one-day autocorrelation is −0.2. What is the option value? With $S = 100$, $X = 120$, $T = 1$, $r = 0.08$, $b = 0.08$, and $\sigma = 0.3$ we get

$$\hat{\sigma} = \sqrt{\frac{0.3^2}{1}\frac{\ln(1+2(-0.2))}{(1+2(-0.2))^{1/1}-1}} = 0.3390$$

We can now calculate the option value:

$$d_1 = \frac{\ln(100/120) + (0.08 + 0.3390^2/2)1}{0.3390\sqrt{1}} = -0.1323$$

$$d_2 = -0.1323 - 0.3390\sqrt{1} = -0.4713$$

$$N(d_1) = N(-0.1323) = 0.4474, \quad N(d_2) = N(-0.4713) = 0.3187$$

which gives a call option value of

$$c = 100e^{(0.08-0.08)1}N(d_1) - 120e^{-0.08\times 1}N(d_2) = 9.4331$$

That is, a value of 9.4331 versus only 7.8966 (as seen from Table 6-1) when not adjusting for autocorrelation. In practice, far more complex stochastic processes can be causing trends.

6.5 ALTERNATIVE STOCHASTIC PROCESSES

Most option pricing formulas in the literature (including most formulas in this book) assume the asset price follows a geometric Brownian motion. This implies that the asset price is lognormally distributed and consistent with limited liability for stocks, and that the returns are normally distributed. This is naturally a simplification of reality. Observed stock return distributions typically have fatter tails and higher peaks, first pointed out by Mitchell (1915), than what the lognormal BSM stock price model implies. There are two main approaches to get around this problem. The first approach is to simply try to "fudge" the BSM model or its numerical equivalent binomial, trinomial, or finite difference implementation. This can be achieved by using different volatilities for every strike and time to maturity. The second approach is to use more complex and hopefully also more realistic models for the stock price process.

We will look at four different adjustments:

1. Constant elasticity of variance (CEV)
2. BSM adjusted for excess skewness and kurtosis
3. Jump-diffusions (JD)
4. Stochastic volatility (SV)

In the following sections, I present closed-form option pricing models for these adjustments. It is, of course, possible to implement the adjustments using numerical methods for option pricing.

6.6 CONSTANT ELASTICITY OF VARIANCE

The BSM formula assumes volatility is constant for the duration of the option contract. Empirical evidence from stock markets shows, however, that the volatility is often negatively correlated with the stock price level. In other words, volatility tends to increase as the stock price decreases. Cox (1975) and Cox and Ross (1976) built such an effect into what is known as the constant elasticity of variance model,

$$dS = \mu S dt + \sigma S^{\beta} dz,$$

where σ is the instantaneous volatility of the asset price returns, β is the elasticity parameter, and dz is a Wiener process. Some well-known special cases for this model obtain for different values of β. $\beta = 0$ gives a normally distributed asset price. $\beta = 0.5$ results in a square root constant elasticity of variance model (SRCEV), equivalent to the Cox, Ingersoll, and Ross (1985) model without the mean reverting part. Finally, $\beta = 1$ yields the BSM stock price model.

CEV for Futures and Forwards

Let's assume the future/forward follows a general constant elasticity of variance model. See Cox (1975) and Cox and Ross (1976).

$$dF = \sigma F^{\beta} dz$$

Hagan and Woodward (1999) shows how to approximate the value of options with this price process as underlying assets. Their approach is to use a Black-equivalent volatility that corresponds to a CEV model:

$$\hat{\sigma} = \frac{\sigma}{f^{1-\beta}} \left[1 + \frac{(1-\beta)(2+\beta)}{24} \left(\frac{F-X}{f} \right)^2 + \frac{(1-\beta)^2}{24} \frac{\sigma^2 T}{f^{2-2\beta}} + \cdots \right], \tag{6.13}$$

where "..." represents additional "negligible" terms, and $f = \frac{1}{2}(F + X)$. This volatility can then simply be used as input in the Black-76 model.

In the special case when the option is at-the-money forward, this Black-equivalent volatility can be simplified further to (see Haug, 2001c):

$$\hat{\sigma} \approx \frac{\sigma}{F^{1-\beta}}$$

Feeding this equivalent volatility into the Brenner and Subrahmanyam (1988) at-the-money forward approximation yields a generalized CEV approximation for options that are at-the-money forward:

$$\text{call} = \text{put} \approx e^{-rT} F^{\beta} \sigma 0.4 \sqrt{T}$$

With this simple expression, we can value at-the-money forward options with remarkable accuracy. With access to a computer power, it is, of course, better to use formula (6.13), as it is more accurate. The at-the-money forward approximation still gives simple access to some intuition about the behavior of the option value and is useful for back-of-the-envelope calculations.

TABLE 6-2

Constant Elasticity of Variance Call Option Values

($S = 100, T = 0.25, r = 0.1, \beta = 0.5$)

σ	$X = 90$	$X = 95$	$X = 100$	$X = 105$	$X = 110$
0.50	9.7531	4.8956	0.9727	0.0224	0.0000
1.00	9.7881	5.2679	1.9453	0.4209	0.0028
2.00	10.5052	6.7612	3.8897	1.9705	0.3330
3.00	11.8414	8.5222	5.8323	3.7838	1.3468
4.00	13.4270	10.3571	7.7721	5.6693	2.7660
5.00	15.1240	12.2200	9.7083	7.5823	4.3931

Table 6-2 shows call option values from the constant elasticity of variance model with $\beta = 0.5$, for different choices of strikes X and volatility σ.

6.7 SKEWNESS-KURTOSIS MODELS

We now consider option pricing models that directly adjust the BSM model to take into account the third and fourth moment of the distribution—skewness and kurtosis.

6.7.1 Definition of Skewness and Kurtosis

The skewness of a series of price data can be measured in terms of the third moment about the mean. If the distribution is symmetric, the skewness will be zero.

$$\text{Skewness} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^3/n}{\sigma^3}, \tag{6.14}$$

where $\bar{x}$ is the mean of the observations, σ is the standard deviation, and n is the number of observations. A normal distribution always has zero skewness, being a symmetric distribution.

The kurtosis describes the relative peakedness of a distribution. The kurtosis is measured by the fourth moment about the mean. To make things more confusing, there is more than one definition of kurtosis. Pearson kurtosis is defined as

$$\frac{\sum_{i=1}^{n}(x_i - \bar{x})^4/n}{\sigma^4} \tag{6.15}$$

Fisher kurtosis, on the other hand, is defined as

$$\frac{\sum_{i=1}^{n}(x_i - \bar{x})^4/n}{\sigma^4} - 3 \tag{6.16}$$

Fisher kurtosis is thus simply Pearson kurtosis minus 3. The normal distribution has a Pearson kurtosis of 3 (Fischer kurtosis of 0) and is called *mesokurtic*. Distributions with Pearson kurtosis larger than 3 (Fisher higher than 0) are called *leptokurtic*, indicating higher peaks and fatter tails than the normal distribution. Pearson kurtosis smaller than 3 (Fischer lower than 0) is termed *playakurtic*. Pearson kurtosis higher than 3 is also called *excess kurtosis*, or simply "fat tails." Before calculating skewness and kurtosis from asset prices, make sure you know if the software you use returns Pearson or Fischer kurtosis.

6.7.2 The Skewness and Kurtosis for a Lognormal Distribution

The skewness and kurtosis of a lognormal distribution will vary across different lognormal distributions depending on the volatility and time horizon. The skewness and kurtosis for different lognormal distribution can be calculated by the following expressions:

$$\text{Skewness} = 3y + y^3$$

$$\text{Fischer kurtosis} = 16y^2 + 15y^4 + 6y^6 + y^8,$$

where

$$y = \sqrt{e^{\sigma^2 T} - 1},$$

σ is the annualized volatility, and T is the time horizon for our analysis (typically the expiration of a derivative contract). Notice that all lognormal distributions have a positive skewness. In other words, the lognormal distribution is always skewed to the right.

Example
What is the skewness and kurtosis for the stock price in a BSM economy, where the stock price follows a geometric Brownian motion? Consider a volatility of 30% and a time horizon of three months. $\sigma = 0.3$, and $T = 0.25$ yields

$$y = \sqrt{e^{0.3^2 \times 0.25} - 1} = 0.1508$$

$$\text{Skewness} = 3 \times 0.1508 + 0.1508^3 = 0.4560$$

$$\text{Fischer kurtosis} = 16 \times 0.1508^2 + 15 \times 0.1508^4 + 6 \times 0.1508^6 + 0.1508^8 = 0.3719$$

The skewness and Fischer kurtosis of the asset returns are still zero, since the returns are normally distributed when the asset price is lognormally distributed.

6.7.3 Jarrow and Rudd Skewness and Kurtosis Model

The Jarrow and Rudd (1982) model is basically the BSM model adjusted for skewness and kurtosis that are different from the lognormal distribution. The model thus adjusts for skewness and kurtosis in the asset price directly (and not in the return distribution):

$$c \approx c_{BSM} + \lambda_1 Q_3 + \lambda_2 Q_4, \tag{6.17}$$

where c_{BSM} are the standard BSM formula. Moreover,

$$d_1 = \frac{\ln(S/X) + (r + \sigma^2/2)T}{\sigma\sqrt{T}} \qquad d_2 = d_1 - \sigma\sqrt{T}$$

$$Q_3 = -(Se^{-rT})^3(e^{\sigma^2 T} - 1)^{3/2}\frac{e^{-rT}}{6}\frac{da(X)}{dS}$$

$$Q_4 = (Se^{-rT})^4(e^{\sigma^2 T} - 1)^2\frac{e^{-rT}}{24}\frac{d^2a(X)}{dS^2}$$

$$\lambda_1 = \gamma_1(F) - \gamma_1(A) \qquad \lambda_2 = \gamma_2(F) - \gamma_2(A)$$

$\gamma_1(A)$ and $\gamma_2(A)$ represent the skewness and kurtosis from the lognormal distribution, while $\gamma_1(F)$ and $\gamma_2(F)$ represent the skewness and kurtosis from the distribution we want to approximate. λ_1 and λ_2 are therefore the excess skewness and kurtosis.

$$a(X) = \frac{1}{X\sigma\sqrt{T2\pi}}e^{-d_2^2/2}$$

$$\frac{da(X)}{dS} = \frac{a(X)(d_2 - \sigma\sqrt{T})}{X\sigma\sqrt{T}}$$

$$\frac{d^2a(X)}{dS^2} = \frac{a(X)}{X^2\sigma^2 T} \times [(d_2 - \sigma\sqrt{T})^2 - \sigma\sqrt{T}(d_2 - \sigma\sqrt{T}) - 1]$$

The put can be found by put-call supersymmetry, or by put-call parity.

Computer algorithm

The computer code for the Jarrow-Rudd skewness-kurtosis model follows.

```
Function JarrowRuddSkewKurt(CallPutFlag As String, S As Double, X _
            As Double, T As Double, r As Double, b As Double, _
    v As Double, Skew As Double, Kurt As Double) As Double


    Dim Q3 As Double, Q4 As Double
    Dim d1 As Double, d2 As Double
    Dim CallValue As Double
```

```
        Dim aX As Double, daX As Double, daXX As Double
        Dim q As Double, GA As Double, gAA As Double
        Dim Lambda1 As Double, Lambda2 As Double

        d1 = (Log(S / X) + (b + v^2 / 2) * T) / (v * Sqr(T))
        d2 = d1 - v * Sqr(T)

        aX = (X * v * Sqr(T * 2 * Pi))^(-1) * Exp(-d2^2 / 2)
        daX = aX * (d2 - v * Sqr(T)) / (X * v * Sqr(T))
        daXX = aX / (X^2 * v * Sqr(T)) _
    * ((d2 - v * Sqr(T))^2 - v * Sqr(T) * (d2 - v * Sqr(T)) - 1)

         q = Sqr(Exp(v^2 * T) - 1)

         GA = 3 * q + q^3
         gAA = 16 * q^2 + 15 * q^4 + 6 * q^6 + q^8 + 3

         Lambda1 = Skew - GA
         Lambda2 = Kurt - gAA

        Q3 = -(S * Exp(r * T))^3 _
        * (Exp(v^2 * T) - 1) ^ (3 / 2) * Exp(-r * T) / 6 * daX
        Q4 = (S * Exp(r * T))^4 * (Exp(v^2 * T) - 1) ^ 2 _
        * Exp(-r * T) / 24 * daXX

        CallValue = (GBlackScholes("c", S, X, T, r, b, v) _
        + Lambda1 * Q3 + Lambda2 * Q4)
         If CallPutFlag = "c" Then
            JarrowRuddSkewKurt = CallValue
        Else '// Use put-call parity to find put value
            JarrowRuddSkewKurt = CallValue - S * Exp((b - r) * T) _
            + X * Exp(-r * T)
        End If
End Function
```

6.7.4 The Corrado and Su Skewness and Kurtosis Model

Corrado and Su (1996) (including a correction by Brown and Robinson, 2002) has published a model somewhat similar to the Jarrow and Rudd (1982) model. Corrado and Su extend the BSM model for non-normal skewness and kurtosis in the stock return distribution rather than through the stock price. This has the advantage that the parameters typically are time-invariant for returns, but not for price levels.

Mathematically, Corrado and Su use a Gram-Charlier series expansion for the probability density to come up with a closed-form adjustment to the BSM formula. The expansion results in

$$c \approx c_{BSM} + \mu_3 Q_3 + (\mu_4 - 3) Q_4, \tag{6.18}$$

where μ_3 is the skewness and μ_4 is the Pearson kurtosis of the asset returns, c_{BSM} is the plain vanilla BSM value, and

$$Q_3 = \frac{1}{6} S\sigma\sqrt{T}[(2\sigma\sqrt{T} - d_1)n(d_1) + \sigma^2 T N(d_1)]$$

TABLE 6-3

Call and Put Option Values from the Corrado-Su Skewness-Kurtosis Model
($S = 100, T = 0.25, r = 0.07, b = 0.07, \sigma = 0.35$)

μ_3	$\mu_4 = 3$	$\mu_4 = 3.5$	$\mu_4 = 4$	$\mu_4 = 4.5$	$\mu_4 = 5$	$\mu_4 = 5.5$
			Put option with strike 75			
−0.2	0.2932	0.3354	0.3776	0.4199	0.4621	0.5044
−0.1	0.2696	0.3118	0.3541	0.3963	0.4386	0.4808
0	0.2460	0.2882	0.3305	0.3727	0.4150	0.4572
0.1	0.2224	0.2646	0.3069	0.3491	0.3914	0.4336
0.2	0.1988	0.2410	0.2833	0.3255	0.3678	0.4100
			Put option with strike 100			
−0.2	6.0317	5.8940	5.7563	5.6186	5.4809	5.3432
−0.1	6.0555	5.9177	5.7800	5.6423	5.5046	5.3669
0	6.0792	5.9415	5.8038	5.6660	5.5283	5.3906
0.1	6.1029	5.9652	5.8275	5.6898	5.5521	5.4143
0.2	6.1266	5.9889	5.8512	5.7135	5.5758	5.4381
			Call option with strike 100			
−0.2	7.7665	7.6288	7.4911	7.3534	7.2157	7.0780
−0.1	7.7902	7.6525	7.5148	7.3771	7.2394	7.1017
0	7.8139	7.6762	7.5385	7.4008	7.2631	7.1254
0.1	7.8376	7.6999	7.5622	7.4245	7.2868	7.1491
0.2	7.8614	7.7237	7.5859	7.4482	7.3105	7.1728
			Call option with strike 125			
−0.2	0.9543	1.0226	1.0910	1.1593	1.2277	1.2961
−0.1	1.0481	1.1165	1.1848	1.2532	1.3215	1.3899
0	1.1419	1.2103	1.2786	1.3470	1.4154	1.4837
0.1	1.2357	1.3041	1.3725	1.4408	1.5092	1.5776
0.2	1.3296	1.3979	1.4663	1.5347	1.6030	1.6714

$$Q_4 = \frac{1}{24} S\sigma\sqrt{T}[(d_1^2 - 1 - 3\sigma\sqrt{T}d_2)n(d_1) + \sigma^3 T^{3/2} N(d_1)]$$

$$d_1 = \frac{\ln(S/X) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

Table 6-3 gives call and put values from the Corrado-Su model for different choices of skewness and kurtosis.

Example

Consider a European call option on a stock with three months to expiration. The current stock price is 100, the strike price is 125, the risk-free rate is 7%, the volatility is 35%, the skewness is 0.1, and the Pearson kurtosis is 5.5 of the expected returns. Hence, $S = 100$,

$X = 125, \; T = 0.25, \; r = 0.07, \; b = 0.07, \; \sigma = 0.35, \; \mu_3 = 0.1, \; \mu_4 = 5.5,$ which yields

$$d_1 = \frac{\ln(100/125) + (0.07 + 0.35^2/2)0.25}{0.35\sqrt{0.25}} = -1.0876$$

$$d_2 = d_1 - 0.35\sqrt{0.25} = -1.2626$$

$$n(d_1) = n(-1.0876) = 0.2208$$

$$N(d_1) = N(-1.0876) = 0.1384$$

$$Q_3 = \frac{1}{6} \times 100 \times 0.35\sqrt{0.25}[(2 \times 0.35\sqrt{0.25} - d_1)n(d_1) + 0.35^2 \times 0.25N(d_1)] = 0.9383$$

$$Q_4 = \frac{1}{24}100 \times 0.35\sqrt{0.25}[(d_1^2 - 1 - 3 \times 0.35\sqrt{0.25}d_2)n(d_1) + 0.35^3 \times 0.25^{3/2}N(d_1)] = 0.1367$$

Moreover, the Black-Scholes value is $c_{BSM}("c",100,125,0.25,0.07,0.07,0.35) = 1.1419$, which gives us the skewness and kurtosis adjusted call value of

$$c = 1.1419 + 0.1 \times Q_3 + (5.5 - 3)Q_4 = 1.5776$$

Skewness and kurtosis can evidently have significant impact on the option value, in this case, 1.5776 versus only 1.1419 for the BSM formula.

Computer algorithm

The computer code for the Corrado-Su skewness-kurtosis model follows.

```
Function SkewKurtCorradoSu(CallPutFlag As String, S As Double, X _
                As Double, T As Double, r As Double, b As Double, _
        v As Double, Skew As Double, Kurt As Double) As Double

        Dim Q3 As Double, Q4 As Double
        Dim d1 As Double, d2 As Double
        Dim CallValue As Double

        d1 = (Log(S / X) + (b + v^2 / 2) * T) / (v * Sqr(T))
        d2 = d1 - v * Sqr(T)
        Q4 = 1 / 24 * S * v * Sqr(T) * ((d1^2 - 1 - 3 * v * Sqr(T) * d2) _
        * ND(d1) + v^3 * T^1.5 * CND(d1))
        Q3 = 1 / 6 * S * v * Sqr(T) * ((2 * v * Sqr(T) - d1) _
        * ND(d1) + v^2 * T * CND(d1))

        CallValue = GBlackScholes("c", S, X, T, r, b, v) _
        + Skew * Q3 + (Kurt - 3) * Q4

         If CallPutFlag = "c" Then
            SkewKurtCorradoSu = CallValue
        Else '// Use put-call parity to find put value
            SkewKurtCorradoSu = CallValue - S * Exp((b - r) * T) _
```

```
            + X * Exp(-r * T)
    End If

End Function
```

Example: *SkewKurtCorradoSu*("c",100, 125, 0.25, 0.07, 0.07, 0.35, 0.1, 5.5) will return a call option value of 1.5776.

6.7.5 Modified Corrado-Su Skewness-Kurtosis Model

For an option model to be arbitrage-free, it must satisfy a martingale restriction (see, for instance, Longstaff, 1995). Jurczenko, Maillet, and Negrea (2004) show that the Corrado-Su formula does not satisfy the martingale restriction and modify it accordingly. This gives the call value

$$c = c_{BSM} + \mu_3 Q_3 + (\mu_4 - 3) Q_4, \tag{6.19}$$

where μ_3 is the skewness, μ_4 is the Pearson kurtosis of the asset returns, c_{BSM} is the plain vanilla BSM value, and

$$Q_3 = \frac{1}{6(1+w)} S\sigma\sqrt{T}(2\sigma\sqrt{T} - d)n(d)$$

$$Q_4 = \frac{1}{24(1+w)} S\sigma\sqrt{T}(d^2 - 3d\sigma\sqrt{T} + 3\sigma^2 T - 1)n(d)$$

$$d = \frac{\ln(S/X) + (b + \sigma^2/2)T - \ln(1+w)}{\sigma\sqrt{T}}$$

$$w = \frac{\mu_3}{6}\sigma^3 T^{3/2} + \frac{\mu_4}{24}\sigma^4 T^2,$$

where, as before, $n(\cdot)$ is the normal density function, μ_3 is the skewness, and μ_4 is the Pearson kurtosis of the return distribution. The put value can be found by put-call supersymmetry or put-call parity.

Table 6-4 gives call and put values from the modified Corrado-Su model for different choices of skewness and kurtosis.

Computer algorithm

The computer code for the modified Corrado-Su skewness-kurtosis model follows.

```
Function SkewKurtCorradoSuModified(CallPutFlag As String, S As Double, _
    X As Double, T As Double, r As Double, b As Double, _
    v As Double, Skew As Double, Kurt As Double) As Double

    Dim Q3 As Double, Q4 As Double
```

TABLE 6-4

Call and Put Option Values from the Modified Corrado-Su Skewness-Kurtosis Model

($S = 100, T = 0.25, r = 0.07, b = 0.07, \sigma = 0.35$)

μ_3	$\mu_4 = 3$	$\mu_4 = 3.5$	$\mu_4 = 4$	$\mu_4 = 4.5$	$\mu_4 = 5$	$\mu_4 = 5.5$
			Put option with strike 75			
−0.2	0.3104	0.3508	0.3911	0.4315	0.4719	0.5122
−0.1	0.2782	0.3186	0.3589	0.3993	0.4396	0.4800
0	0.2460	0.2863	0.3267	0.3670	0.4074	0.4477
0.1	0.2137	0.2541	0.2944	0.3347	0.3751	0.4154
0.2	0.1814	0.2218	0.2621	0.3024	0.3427	0.3831
			Put option with strike 100			
−0.2	6.0421	5.9032	5.7644	5.6255	5.4867	5.3478
−0.1	6.0606	5.9217	5.7829	5.6440	5.5052	5.3664
0	6.0792	5.9403	5.8015	5.6627	5.5238	5.3850
0.1	6.0979	5.9591	5.8202	5.6814	5.5426	5.4038
0.2	6.1167	5.9779	5.8391	5.7003	5.5615	5.4227
			Call option with strike 100			
−0.2	7.7769	7.6380	7.4992	7.3603	7.2215	7.0826
−0.1	7.7953	7.6565	7.5177	7.3788	7.2400	7.1011
0	7.8139	7.6751	7.5363	7.3974	7.2586	7.1198
0.1	7.8327	7.6938	7.5550	7.4162	7.2774	7.1386
0.2	7.8515	7.7127	7.5739	7.4351	7.2963	7.1574
			Call option with strike 125			
−0.2	0.9567	1.0248	1.0929	1.1610	1.2292	1.2974
−0.1	1.0493	1.1175	1.1856	1.2538	1.3221	1.3903
0	1.1419	1.2101	1.2784	1.3466	1.4149	1.4833
0.1	1.2344	1.3027	1.3710	1.4394	1.5077	1.5761
0.2	1.3269	1.3953	1.4636	1.5320	1.6005	1.6689

```
    Dim d As Double, w As Double
    Dim CallValue As Double

    w = Skew / 6 * v^3 * T^1.5 + Kurt / 24 * v^4 * T^2
    d = (Log(S / X) + (b + v^2 / 2) * T - Log(1 + w)) / (v * Sqr(T))
    Q3 = 1 / (6 * (1 + w)) * S * v * Sqr(T) * (2 * v * Sqr(T) - d) * ND(d)
    Q4 = 1 / (24 * 1 + w) * S * v * Sqr(T) _
      * (d^2 - 3 * d * v * Sqr(T) + 3 * v^2 * T - 1) * ND(d)

    CallValue = GBlackScholes("c", S, X, T, r, b, v) + Skew * Q3 + (Kurt - 3
    If CallPutFlag = "c" Then
        SkewKurtCorradoSuModified = CallValue
    Else '// Use put-call parity to find put value
        SkewKurtCorradoSuModified = CallValue - S * Exp((b - r) * T) _
        + X * Exp(-r * T)
    End If

End Function
```

Example: *SkewKurtCorradoSuModified*("c", 100, 125, 0.25, 0.07, 0.07, 0.35, 0.1, 5.5) will return a call option value of 1.5761.

6.7.6 Skewness-Kurtosis Put-Call Supersymmetry

Haug (2002) extends the standard supersymmetry introduced in Chapter 1 to also hold between puts and calls with skewness and kurtosis taken into account:

$$c(S, X, T, r, b, \sigma, \mu_3, \mu_4) = -p(S, X, T, r, b, -\sigma, -\mu_3, \mu_4) \tag{6.20}$$

and naturally also

$$p(S, X, T, r, b, \sigma, \mu_3, \mu_4) = -c(S, X, T, r, b, -\sigma, -\mu_3, \mu_4), \tag{6.21}$$

where μ_3 is the skewness and μ_4 is the kurtosis. This result holds for Edgeworth and Gram-Charlie expansions of the BSM formula, as well as for many discrete implementations.

6.7.7 Skewness-Kurtosis Equivalent Black-Scholes-Merton Volatility

Backus, Foresi, and Wu (1997) show how one can adjust the volatility in a BSM model to take skewness and kurtosis into account:

$$\hat{\sigma} \approx \sigma \left[1 - \frac{\mu_3}{6} d_1 - \frac{\mu_4 - 3}{24}(1 - d_1^2) \right], \tag{6.22}$$

where

$$d_1 = \frac{\ln(S/X) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$$

To value the option simply input the skewness and kurtosis adjusted volatility $\hat{\sigma}$ into the BSM formula. The adjustment seems to be quite consistent with the Corrado and Su (1996) model as long as the option is at-the-money forward (or at least close to at-the-money forward). For out-of-the-money options, this approach does not seem very accurate.

6.7.8 Gram Charlier Density

Knight and Satchell (2001) show how to turn the normal density function $\frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ into a density function that takes into account non-normal skewness and kurtosis:

$$n'(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \left(1 + \frac{\mu_3}{6}(x^3 - 3x) + \frac{\mu_4 - 3}{24}(x^4 - 6x^2 + 3) \right), \tag{6.23}$$

where μ_3 is the skewness and μ_4 is the Pearson kurtosis.

6.7.9 Skewness-Kurtosis Trees

I present a binomial model in Chapter 7 that takes skewness and kurtosis into account.

6.8 PASCAL DISTRIBUTION AND OPTION PRICING

To incorporate fat tails in the distribution, Ray (1993) suggests switching from normal distributed returns to the Pascal distribution. The probability density of the Pascal distribution is given by

$$f(x) = \frac{e^{-|x|/\beta}}{2\beta}$$

$$\beta = \int_{-\infty}^{\infty} |x| f(x) dx,$$

where x is the change in price and β is the mean absolute deviation of change in price (MAD). The value of a call and put option is then given by

$$c = e^{-rT+(F-X)/\beta} \frac{\beta}{2} \tag{6.24}$$

$$p = e^{-rT+(X-F)/\beta} \frac{\beta}{2}, \tag{6.25}$$

where F is the forward price of the underlying asset. Ray (1993) offers little explanation of what type of stochastic process results in this distribution. She considers multiple examples where the Pascal distribution fits historical bond yields better than the normal distribution.

6.9 JUMP-DIFFUSION MODELS

Observed asset prices do not move continuously in time. A reasonable model is therefore to let prices make discrete jumps from time to time. Since jumps can have important effects on the option value, we will look at two of the better known jump-diffusion models described in the literature.

6.9.1 The Merton Jump-Diffusion Model

An example of such a model is the Merton (1976) jump-diffusion model. The model assumes that the underlying asset price follows the jump-diffusion process:

$$dS = (b - \lambda k) S dt + \sigma S dz + k dq,$$

where dz is a Brownian motion as before, while dq is the jump component. The two stochastic processes dz and dq are assumed to be uncorrelated. The model now requires two additional parameters to be estimated, over and above the BSM model: the expected number of jumps per year λ and the percentage of the total volatility explained by jumps γ. Merton then shows that

$$c = \sum_{i=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^i}{i!} c_i(S, X, T, r, \sigma_i) \tag{6.26}$$

$$p = \sum_{i=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^i}{i!} p_i(S, X, T, r, \sigma_i), \tag{6.27}$$

where

$$\sigma_i = \sqrt{z^2 + \delta^2(i/T)}$$

$$\delta = \sqrt{\frac{\gamma v^2}{\lambda}} \qquad z = \sqrt{v^2 - \lambda\delta^2}$$

Where v is the total volatility including jumps. Merton (1976) assumes that the jump risk can be fully diversified by holding a stock portfolio, that some stocks will always jump in opposite direction, and that the jump risk in this way is nonsystematic.

Any of the option Greeks can be found by simply replacing the $c_i(S, X, T, r, \sigma_i)$ or $p_i(S, X, T, r, \sigma_i)$ option formula in (6.26) or (6.27) with the any of the BSM Greek formulas given in Chapter 2.

Computer algorithm

```
Function JumpDiffusionMerton(CallPutFlag As String, S As Double, _
    X As Double, T As Double, r As Double, v As Double, _
                lambda As Double, gamma As Double) As Double

    Dim delta As Double, Sum As Double
    Dim Z As Double, vi As Double
    Dim i As Integer

    delta = Sqr(gamma * v^2 / lambda)
    Z = Sqr(v^2 - lambda * delta^2)
    Sum = 0
    For i = 0 To 50
        vi = Sqr(Z^2 + delta^2 * (i / T))
        Sum = Sum + Exp(-lambda*T)*(lambda * T)^i / Application.Fact(i) _
        * GBlackScholes(CallPutFlag, S, X, T, r, r, vi)
    Next
        JumpDiffusionMerton = Sum

End Function
```

TABLE 6-5

Merton-76 Jump-Diffusion Call Values

($S = 100, \sigma = 0.25, r = 0.08$)

γ	Strike	$\lambda = 1$ Time to Maturity			$\lambda = 5$ Time to Maturity			$\lambda = 10$ Time to Maturity		
		0.1	0.25	0.5	0.1	0.25	0.5	0.1	0.25	0.5
	80	20.67	21.74	23.63	20.65	21.70	23.61	20.64	21.70	23.61
	90	11.00	12.74	15.40	10.98	12.75	15.42	10.98	12.75	15.42
0.25	100	3.42	5.88	8.95	3.51	5.96	9.02	3.53	5.97	9.03
	110	0.55	2.11	4.67	0.56	2.16	4.73	0.56	2.17	4.74
	120	0.10	0.64	2.23	0.06	0.63	2.25	0.05	0.62	2.25
	80	20.72	21.83	23.71	20.66	21.73	23.63	20.65	21.71	23.62
	90	11.04	12.72	15.34	11.02	12.76	15.41	11.00	12.75	15.41
0.50	100	3.14	5.58	8.71	3.39	5.87	8.96	3.46	5.93	9.00
	110	0.53	1.93	4.42	0.58	2.11	4.67	0.57	2.15	4.71
	120	0.19	0.71	2.15	0.10	0.66	2.23	0.07	0.64	2.24
	80	20.79	21.96	23.86	20.68	21.78	23.67	20.66	21.74	23.64
	90	11.11	12.75	15.30	11.09	12.78	15.39	11.04	12.76	15.41
0.75	100	2.70	5.08	8.24	3.16	5.71	8.85	3.33	5.85	8.95
	110	0.54	1.69	3.99	0.62	2.05	4.57	0.60	2.11	4.66
	120	0.29	0.84	2.09	0.15	0.71	2.22	0.11	0.67	2.23

where $CND(\cdot)$ is the cumulative normal distribution function and the function $GBlackScholes(\cdot)$ is the generalized BSM function.

Example: *JumpDiffusion*("c", 45, 55, 0.25, 0.1, 0.25, 3, 0.4) returns a call value of 0.2417.

Table 6-5 shows call option values from the Merton jump-diffusion model. Values are tabulated with different values for the option's gamma γ, strike, number of jumps per year λ, and time to maturity.

6.9.2 Bates Generalized Jump-Diffusion Model

In the footsteps of the Merton-76 model, Bates (1991) published a more general jump-diffusion model.[4] His work differs from the Merton (1976), Ball and Torous (1983), and Ball and Torous (1985) jump-diffusion models in several important ways:

- Jumps are allowed to be asymmetric—in other words, with nonzero mean.

[4] For more on option pricing under the assumptions of jump diffusion, see also Ball and Torous (1983), Ball and Torous (1985), Aase (1988), and Amin (1993).

- Since we often have to deal with options on stock index futures (e.g., S&P index options), it is hardly plausible to maintain Merton's simplifying assumption that jump risk is idiosyncratic and thus fully diversifiable.

The Bates (1991) jump-diffusion model is consistent with an asymmetric volatility smile (generated from a BSM-type model). This is often what we observe in practice.

Bates assumes that consumers have time-separable power utility and that optimally invested wealth follows a jump-diffusion. He moreover assumes jump risk is systematic: all asset prices and wealth jump simultaneously, possibly by different amounts. This seems to be much closer to reality than the Merton (1976) model, especially for stock index options during a market crash. Bates (1991) prescribes the following jump-diffusion process under a risk-neutral probability measure:

$$dS = (b - \lambda\bar{k})Sdt + \sigma Sdz + kdq$$

The process resembles geometric Brownian motion most of the time, but on average λ times per year, the price jumps discretely by a random amount.

S = Asset.
b = Cost-of-carry.
σ = Volatility of the relative price change based on no jumps.
k = Random percentage jump conditional upon a Poisson-distributed event occurring, where $1 + k$ is lognormal distributed.
$\bar{k}$ = Expected jump size.
λ = Frequency of Poisson events.
q = Poisson counter with intensity λ.

From this process, Bates develops the following formulas for European call and put options:

$$c = \sum_{i=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^i}{i!} c_i(S, X, T, r, b_i, \sigma_i) \tag{6.28}$$

$$p = \sum_{i=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^i}{i!} p_i(S, X, T, r, b_i, \sigma_i), \tag{6.29}$$

where

$$b_i = b - \lambda\bar{k} + \frac{i\bar{\gamma}}{T}$$

$$\sigma_i = \sqrt{\sigma^2 + \delta^2(i/T)},$$

where $\bar{\gamma} = \ln(1 + \bar{k})$ and δ is the standard deviation of log asset price jumps. Input of the expected jump size $\bar{k}$ greater (smaller) than zero implies that the distribution is positively (negatively) skewed relative to geometric Brownian motion. The option formula is in the form of an infinite sum. One can show that it converges rather fast.

Any of the Bates jump-diffusion option Greeks can be found by replacing the BSM formula $c_i(S, X, T, r, b_i, \sigma_i)$ in (6.28) or (6.29) with any of the BSM Greeks, but with the same input parameters. VBA code for the Bates jump-diffusion and all its Greeks are included on the accompanying CD.

From the model, we can also find several interesting implicit parameters like implied total annual volatility.

$$v = \sqrt{\sigma^2 + \lambda(\hat{\gamma}^2 + \delta^2)},$$

where $\hat{\gamma} = \bar{\gamma} - \delta^2/2$. From this we can easily compare jump-diffusion values with BSM values. Moreover, the skewness (third moment) and Pearson kurtosis (fourth moment) of this jump-diffusion model is given by

$$Skewness = \frac{\lambda\hat{\gamma}(\hat{\gamma}^2 + 3\delta^2)T^{-\frac{1}{2}}}{v^3}$$

$$Kurtosis = 3 + \frac{\lambda(\hat{\gamma}^4 + 6\hat{\gamma}^2\delta^2 + 3\delta^4)T^{-1}}{v^4}$$

An alternative way to value options under a jump-diffusion process is to approximate the jump-diffusion value using the skewness and kurtosis adjusted BSM model described earlier in this chapter. It is evident from the above expressions that the kurtosis and skewness we get from a jump-diffusion model can be quite large though, especially for options with short time to maturity. The Edgeworth and Gram-Charlier expansion methods presented above have limited ability to match distributions with high kurtosis and large skewness. This can easily be tested by using the skewness and kurtosis from the jump-diffusion process as input to the skewness-kurtosis models. We will then often see that the skewness-kurtosis models have problems matching the jump-diffusion model.

Table 6-6 shows call option values from the Bates jump-diffusion model. Values are tabulated with different values for the option's delta δ, strike, number of jumps per year λ, and time to maturity.

Computer algorithm

VBA code for the Bates generalized jump-diffusion model follows.

```
Function JumpDiffusionBates(CallPutFlag As String, S As Double, X As Double, _
```

TABLE 6-6

Bates-91 Jump-Diffusion Call Values

($S = 100, \sigma = 0.25, r = 0.08, \bar{k} = -0.04$)

δ	Strike	$\lambda = 1$ Time to Maturity			$\lambda = 5$ Time to Maturity			$\lambda = 10$ Time to Maturity		
		0.1	0.25	0.5	0.1	0.25	0.5	0.1	0.25	0.5
	80	20.67	21.81	23.86	20.83	22.32	24.85	21.05	22.97	25.88
	90	11.12	13.06	15.88	11.68	14.13	17.50	12.30	15.28	19.08
0.10	100	3.77	6.38	9.61	4.59	7.77	11.60	5.49	9.23	13.55
	110	0.67	2.48	5.29	1.14	3.67	7.25	1.75	5.04	9.28
	120	0.07	0.78	2.67	0.22	1.52	4.30	0.46	2.51	6.16
	80	20.95	22.45	24.97	22.11	25.12	29.48	23.46	27.95	33.58
	90	11.57	14.00	17.44	13.69	18.03	23.42	15.88	21.78	28.41
0.25	100	4.27	7.46	11.40	6.88	12.25	18.34	9.61	16.63	23.97
	110	1.10	3.49	7.07	3.17	8.05	14.21	5.57	12.53	20.20
	120	0.37	1.58	4.25	1.74	5.31	10.97	3.49	9.43	17.01
	80	21.66	24.08	27.84	25.31	31.66	39.68	29.16	38.58	48.67
	90	12.41	15.93	20.83	17.45	25.62	34.99	22.52	33.83	45.11
0.50	100	5.18	9.55	15.11	10.93	20.48	30.89	16.78	29.65	41.92
	110	1.99	5.59	10.88	7.25	16.52	27.35	12.87	26.10	39.06
	120	1.21	3.60	8.00	5.64	13.69	24.35	10.60	23.15	36.49

```
T As Double, r As Double, b As Double, v As Double, _
lambda As Double, avgk As Double, delta As Double) As Double

    Dim Sum As Double, gam0 As Double, gambar As Double
    Dim bi As Double, vi As Double
    Dim i As Integer

    gam0 = Log(1 + avgk)
    Sum = 0
    For i = 0 To 50
        bi = b - lambda * avgk + gam0 * (i / T)
        vi = Sqr(v^2 + delta^2 * (i / T))
        Sum = Sum + Exp(-lambda * T) * (lambda * T)^i / Application.Fact(i) _
        * GBlackScholes(CallPutFlag, S, X, T, r, bi, vi)
    Next
        JumpDiffusionBates = Sum

End Function
```

6.10 STOCHASTIC VOLATILITY MODELS

Observed volatility is not constant but rather contains some deterministic part and some stochastic parts. We now look at a couple of stochastic volatility models, meaning that the volatility itself is modeled as a stochastic process.

6.10.1 Hull-White Uncorrelated Stochastic Volatility Model

Hull and White (1987) discuss option valuation under stochastic volatility for the special case where the volatility is uncorrelated with the asset price. Assume the asset price and the instantaneous variance $V = \sigma^2$ follow the following stochastic processes in a risk neutral world:

$$dS = bS_t dt + \sigma S_t dz$$

$$dV = \alpha V + \xi V dw,$$

where the drift of the variance α and the volatility of the variance ξ do not depend on S. Moreover, dz and dw are independent Wiener processes, meaning that the asset price and volatility are uncorrelated. b is the cost-of-carry of the asset as before. Hull and White (1987) show that the option value is then simply given as an integral over the BSM formula, over the distribution of the mean volatility:

$$c_{HW87} = \int c(\bar{V}) h(\bar{V}|\sigma_t^2) d\bar{V}, \tag{6.30}$$

where $\bar{V}$ is the mean variance over the lifetime of the option,

$$\bar{V} = \frac{1}{T}\int_0^T \sigma^2(t)dt,$$

and $c(\bar{V})$ is the BSM formula, here written in the notation of variance instead of standard deviation:

$$c(\bar{V}) = SN(d_1) - Xe^{-rT}N(d_2)$$

$$d_1 = \frac{\ln(S/X) + (b + \bar{V}/2)T}{\sqrt{\bar{V}T}}$$

$$d_2 = d_1 - \sqrt{\bar{V}T}$$

Although it does not seem possible to find a simple closed-form solution for the option price, it is still possible to find relatively simple formulas for the moments of $\bar{V}$ (see Hull and White, 1987). When $\mu \neq 0$:

$$E[\bar{V}] = \frac{e^{\mu T} - 1}{\mu T} V_0$$

$$E[\bar{V}^2] = \left[\frac{2e^{(2\mu+\xi^2)T}}{(\mu+\xi^2)(2\mu+\xi^2)T^2} + \frac{2}{\mu T^2}\left(\frac{1}{2\mu+\xi^2} - \frac{e^{\mu T}}{\mu+\xi^2}\right)\right] V_0^2,$$

and if $\mu = 0$ we have

$$E[\bar{V}] = V_0$$

The second moment (the variance of the variance) is as follows.

$$E[\bar{V}^2] = \frac{2(e^{\xi^2 T} - \xi^2 T - 1)}{\xi^4 T^2} V_0^2,$$

and the third moment (skewness) is

$$E[\bar{V}^3] = \frac{e^{3\xi^2 T} - 9e^{\xi^2 T} + 6\xi^2 T + 8}{3\xi^6 T^3} V_0^3$$

In the case of sufficiently small values of ξ, Hull and White (1987) indicate that a Taylor series converges quickly and in this case can be used to approximate the option value. In the case of $\mu = 0$, we get a call value of

$$c_{HWuSV} \approx c_{BSM}(\sigma) + \frac{1}{2}\frac{\partial^2 c(\bar{V})}{\partial \bar{V}^2} \times \left(\frac{2\sigma^4(e^k - k - 1)}{k^2} - \sigma^4\right) + \frac{1}{6}\frac{\partial^3 c(\bar{V})}{\partial \bar{V}^3}$$
$$\times \sigma^6 \left(\frac{e^{3k} - (9 + 18k)e^k + 8 + 24k + 18k^2 + 6k^3}{3k^3}\right) + \ldots, \qquad (6.31)$$

where $k = \xi^2 T$ and

$$\frac{\partial^2 c(\bar{V})}{\partial \bar{V}^2} = \frac{Se^{(b-r)T}\sqrt{T}}{4\sigma^3} n(d_1)(d_1 d_2 - 1)$$

$$\frac{\partial^3 c(\bar{V})}{\partial \bar{V}^3} = \frac{Se^{(b-r)T}\sqrt{T}}{8\sigma^5} n(d_1)[(d_1 d_2 - 1)(d_1 d_2 - 3) - (d_1^2 + d_2^2)]$$

The value of a put option we can simply find by using put-call parity:

$$p_{HW87} = c_{HW87} - Se^{(b-r)T} + Xe^{-rT}$$

Computer algorithm

Here is a computer algorithm of the Hull and White (1987) stochastic volatility model. The algorithm is fast and efficient even in VBA, as the model is a closed-form approximation.

```
Function HullWhite87SV(CallPutFlag As String, S As Double, X As Double, _
  T As Double, r As Double, b As Double, _
  v As Double, Vvol As Double) As Double
```

```
    '// v: intitial volatility/standard deviation
    '// VVol: volatility of volatility
    '// rho: correlation between asset price and volatility

    Dim d1 As Double, d2 As Double, k As Double
    Dim CallValue As Double, cgbs As Double, ek As Double
    Dim cVV As Double, cVVV As Double

    k = Vvol^2 * T
    ek = Exp(k)

    d1 = (Log(S / X) + (b + v^2 / 2) * T) / (v * Sqr(T))
    d2 = d1 - v * Sqr(T)
    cgbs = GBlackScholes("c", S, X, T, r, b, v)

    '//Partial derivatives
    cVV = S * Exp((b - r) * T) * Sqr(T) * ND(d1) _
        * (d1 * d2 - 1) / (4 * v^3)
    cVVV = S * Exp((b - r) * T) * Sqr(T) * ND(d1) _
        * ((d1 * d2 - 1)  * (d1 * d2 - 3) - (d1^2 + d2^2)) / (8 * v^5)

    CallValue = cgbs + 1 / 2 * cVV * (2 * v^4 * (ek - k - 1) / k^2 - v^4) _
            + 1 / 6 * cVVV * v^6 * (ek ^ 3 - (9 + 18 * k) * ek + 8 _
            + 24 * k + 18 * k^2 + 6 * k^3) / (3 * k^3)

    If CallPutFlag = "c" Then
        HullWhite87SV = CallValue
    ElseIf CallPutFlag = "p" Then '// Use put call parity to find put
        HullWhite87SV = CallValue - S * Exp((b - r) * T) + X * Exp(-r * T)
    End If

End Function
```

6.10.2 Hull-White Correlated Stochastic Volatility Model

Hull and White (1988) develop a closed-form approximation for European option values under stochastic volatility. Unlike the Hull and White (1987) model, their 1988 model opens up for volatility to be instantaneously correlated to the asset price. More precisely, they assume a square-root stochastic volatility process

$$dS = \mu S dt + S\sqrt{V} dz$$

$$dV = (a + \beta V)dt + \xi\sqrt{V} dw,$$

where V is the stochastic variance of the underlying asset, ξ is the volatility of the volatility (standard deviation of $\sqrt{V}$), and a and β are constants that determine the drift of the variance. To ensure that V remains positive, $a \geq 0$. When $a > 0$ and $\beta < 0$, this is a mean reverting stochastic volatility process $-\frac{a}{\beta}$ is the mean reversion level of V, with speed of mean reversion $-\beta$. dz and dw are two correlated Wiener processes, thus making it possible for the volatility to be correlated with the asset price.

The time required for the expected deviation to be halved, the half-life, is given by $H_L = \frac{-\ln(2)}{\beta}$, or alternatively we can find the β given the half-life $\beta = \frac{-\ln(2)}{H_L}$. The mean reversion level of the variance is given by $\hat{V} = -a/\beta$. Assuming we know the mean reversion level of the variance $\hat{V}$, we can naturally solve for the constant a: $a = -\beta\hat{V}$.

Hull and White (1988) are able to come up with a closed-form approximation (series solution) for plain vanilla European calls under stochastic volatility c_{HW88}:

$$c_{HW88} \approx f_0 + f_1\xi + f_2\xi^2, \tag{6.32}$$

where, in the case $\beta \neq 0$

$$f_0 = c(\bar{V})$$

$$f_1 = \frac{\rho}{\beta^3 T}[(a + \beta V)(1 - e^\delta + \delta e^\delta) + a(1 + \delta - e^\delta)]\frac{\partial^2 C(\bar{V})}{\partial S \partial \bar{V}}$$

$$f_2 = \frac{\phi_1}{T}\frac{\partial^2 C(\bar{V})}{\partial S \partial \bar{V}} + \frac{\phi_2}{T^2}\frac{\partial^2 C(\bar{V})}{\partial \bar{V}^2} + \frac{\phi_3}{T^2}\frac{\partial^3 C(\bar{V})}{\partial S \partial \bar{V}^2} + \frac{\phi_4}{T^3}\frac{\partial^3 C(\bar{V})}{\partial \bar{V}^3},$$

where[5]

$$\bar{V} = \left(V + \frac{a}{\beta}\right)\frac{e^\delta - 1}{\delta} - \frac{a}{\beta}$$

$$\frac{\partial^2 c(\bar{V})}{\partial S \partial \bar{V}} = -Se^{(b-r)T}n(d_1)\frac{d_2}{2\bar{V}}$$

$$\frac{\partial^2 c(\bar{V})}{\partial \bar{V}^2} = \frac{Se^{(b-r)T}\sqrt{T}}{4\bar{V}^{3/2}}n(d_1)(d_1 d_2 - 1)$$

$$\frac{\partial^3 c(\bar{V})}{\partial S \partial \bar{V}^2} = \frac{Se^{(b-r)T}}{4\bar{V}^2}n(d_1)(-d_1 d_2^2 + d_1 + 2d_2)$$

$$\frac{\partial^3 c(\bar{V})}{\partial \bar{V}^3} = \frac{Se^{(b-r)T}\sqrt{T}}{8\bar{V}^{5/2}}n(d_1)[(d_1 d_2 - 1)(d_1 d_2 - 3) - (d_1^2 + d_2^2)]$$

$$\phi_1 = \frac{\rho^2}{\beta^4}\left\{(a + \beta V)\left[e^\delta\left(\frac{\delta^2}{2} - \delta + 1\right) - 1\right] + a[e^\delta(2 - \delta) - (2 + \delta)]\right\}$$

$$\phi_2 = 2\phi_1 + \frac{1}{2\beta^4}\left((a + \beta V)(e^{2\delta} - 2\delta e^\delta - 1) - \frac{a}{2}(e^{2\delta} - 4e^\delta + 2\delta + 3)\right)$$

[5]In their 1987 model no mixed derivatives are used. The reason is that Hull and White here assume zero correlation between spot and variance. In their 1988 extension they add correlation, and now mixed derivatives are necessary.

$$\phi_3 = \frac{\rho^2}{2\beta^6}[(a + \beta V)(e^\delta - \delta e^\delta - 1) - a(1 + \delta - e^\delta)]^2$$

$$\phi_4 = 2\phi_3$$

$$d_1 = \frac{\ln(S/X) + (b + \bar{V}/2)T}{\sqrt{\bar{V}T}}$$

$$d_2 = d_1 - \sqrt{\bar{V}T},$$

and $\delta = \beta T$, $V = \sigma^2$. In the special case when $\beta = 0$ we get

$$\bar{V} = V + \frac{aT}{2}$$

$$f_1 = \rho\left(V + \frac{aT}{3}\right)\frac{T}{2}\frac{\partial^2 C(\bar{V})}{\partial S \partial \bar{V}}$$

and

$$\phi_1 = \rho^2\left(V + \frac{aT}{4}\right)\frac{T^3}{6}$$

$$\phi_2 = \left(2 + \frac{1}{\rho^2}\right)\phi_1$$

$$\phi_3 = \rho^2\left(V + \frac{aT}{3}\right)^2\frac{T^4}{8}$$

$$\phi_4 = 2\phi_3$$

When the volatility is constant $\bar{V} = V$, we get the BSM formula. The value of a put option can be found from the put-call parity

$$p_{HW88} = c_{HW88} - Se^{(b-r)T} + Xe^{-rT}$$

Comments

The Hull and White (1988) is a great model to play around with to get some intuition on how stochastic volatility can affect option values. The closed-form approximation is not used much in practice, however, as there are several limits to what values of parameters it accepts. Moreover it is not a very robust approximation, and for some input parameters, the model can even return negative option values.

Table 6-7 reports call and put values generated by the Hull and White (1988) stochastic volatility model. The values with strike below 100 are put options, and the values with strike above 100 are call options. In the case of strike 100, call options and put options will have the same value due to put-call parity. Table 6-8 shows how these values are reflected in implied BSM volatilities. We can easily see the impact on correlation between the volatility and the asset price showing up in the BSM volatility smile.

TABLE 6-7

Hull and White (1988) Stochastic Volatility Model Option Values

($S = 100, T = 0.25, r = 0.08, b = 0, \sigma_0^2 = V = 0.09, \hat{V} = 0.0625, H_L = 0.1, \xi = 0.5$)

	Put Values			Call = Put	Call Values		
ρ	$X = 70$	$X = 80$	$X = 90$	$X = 100$	$X = 110$	$X = 120$	$X = 130$
−0.75	0.0904	0.4942	1.8559	5.2591	1.6192	0.2705	0.0110
−0.50	0.0646	0.4278	1.7809	5.2765	1.7568	0.4039	0.0678
−0.25	0.0436	0.3596	1.6982	5.2921	1.8874	0.5323	0.1298
0.00	0.0273	0.2895	1.6078	5.3061	2.0110	0.6559	0.1968
0.25	0.0158	0.2175	1.5098	5.3183	2.1274	0.7745	0.2690
0.50	0.0090	0.1436	1.4040	5.3289	2.2368	0.8881	0.3463
0.75	0.0069	0.0678	1.2906	5.3378	2.3391	0.9968	0.4287

Computer algorithm

Here is a computer algorithm of the Hull and White (1988) stochastic volatility model. As the model is a closed-form approximation, it is fast and efficient even in VBA.

```
Function HullWhite88SV(CallPutFlag As String, S As Double, X As Double, _
  T As Double, r As Double, b As Double, sig0 As Double, sigLR As Double, _
  HL As Double, Vvol As Double, rho As Double) As Double

    '// sig0: initial volatility
    '// sigLR: the long run mean reversion level of volatility
    '// HL: half-life of volatility deviation
    '// VVol: volatility of volatility
    '// rho: correlation between asset price and volatility

   Dim phi1 As Double, phi2 As Double, phi3 As Double, phi4 As Double
   Dim f0 As Double, f1 As Double, f2 As Double
   Dim d1 As Double, d2 As Double
   Dim cSV As Double, cVV As Double, cSVV As Double, cVVV As Double
   Dim ed As Double, delta As Double, Beta As Double, a As Double
   Dim v As Double, Vbar As Double, CallValue As Double

   Beta = -Log(2) / HL '// Find  constant, beta, from Half Life
   a = -Beta * sigLR^2 '// Find constant, a, from long run volatility
   delta = Beta * T
   ed = Exp(delta)
   v = sig0^2

   If Abs(Beta) < 0.0001 Then
       Vbar = v + 0.5 * a * T '// Average expected variance
    Else
    '// Average expected variance:
       Vbar = (v + a / Beta) * (ed - 1) / delta - a / Beta
   End If

   d1 = (Log(S / X) + (b + Vbar / 2) * T) / Sqr(Vbar * T)
```

```
    d2 = d1 - Sqr(Vbar * T)

    '// Partial derivatives
    cSV = -S * Exp((b - r) * T) * ND(d1) * d2 / (2 * Vbar)
    cVV = S * Exp((b - r) * T) * ND(d1) * Sqr(T) _
        / (4 * Vbar^1.5) * (d1 * d2 - 1)
    cSVV = S * Exp((b - r) * T) / (4 * Vbar^2) * ND(d1) * _
            (-d1 * d2^2 + d1 + 2 * d2)
    cVVV = S * Exp((b - r) * T) * ND(d1) * Sqr(T) / (8 * Vbar^2.5) _
            * ((d1 * d2 - 1) * (d1 * d2 - 3) - (d1^2 + d2^2))

    If Abs(Beta) < 0.0001 Then
        f1 = rho * (a * T / 3 + v) * T / 2 * cSV
        phi1 = rho^2 * (a * T / 4 + v) * T^3 / 6
        phi2 = (2 + 1 / rho^2) * phi1
        phi3 = rho^2 * (a * T / 3 + v)^2 * T^4 / 8
        phi4 = 2 * phi3
    Else '// Beta different from zero
        phi1 = rho^2 / Beta^4 * ((a + Beta * v) _
            * (ed * (delta^2 / 2 - delta + 1) - 1) _
            + a * (ed * (2 - delta) - (2 + delta)))
        phi2 = 2 * phi1 + 1 / (2 * Beta^4) * ((a + Beta * v) _
            * (ed^2 - 2 * delta * ed - 1) _
            - a / 2 * (ed^2 - 4 * ed + 2 * delta + 3))
        phi3 = rho^2 / (2 * Beta^6) * ((a + Beta * v) _
        * (ed - delta * ed - 1) - a * (1 + delta - ed))^2
        phi4 = 2 * phi3
        f1 = rho / (Beta^3 * T) * ((a + Beta * v) * (1 - ed + delta * ed) _
        + a * (1 + delta - ed)) * cSV
    End If

    f0 = S * Exp((b - r) * T) * CND(d1) - X * Exp(-r * T) * CND(d2)
    f2 = phi1 / T * cSV + phi2 / T^2 * cVV _
      + phi3 / T^2 * cSVV + phi4 / T^3 * cVVV

    CallValue = f0 + f1 * Vvol + f2 * Vvol^2

    If CallPutFlag = "c" Then
        HullWhite88SV = CallValue
    ElseIf CallPutFlag = "p" Then '// Use put call parity to find put
        HullWhite88SV = CallValue - S * Exp((b - r) * T) + X * Exp(-r * T)
    End If

End Function
```

6.10.3 The SABR Model

The SABR model (Stochastic, α, β, ρ model) published by Hagan, Kumar, Lesniewski, and Woodward (2002) is an interesting model from an option trader's perspective. The Black-76 model has for years been the benchmark model for most European options on currency, interest rates, and stock indices, as well as on commodity and energy futures. The main drawback of the Black-76 (Black-Scholes-Merton) model has been the assumption of constant volatility, or at best a deterministic time varying volatility. Traders have naturally been aware

TABLE 6-8

Hull and White (1988) Stochastic Volatility Model in Form of Black-Scholes-Merton Implied Volatilities

($S = 100$, $T = 0.25$, $r = 0.08$, $b = 0$, $\sigma_0^2 = V = 0.09$, $\hat{V} = 0.0625$, $H_L = 0.1$, $\xi = 0.5$)

ρ	$X = 70$	$X = 80$	$X = 90$	$X = 100$	$X = 110$	$X = 120$	$X = 130$
−0.75	34.00%	31.61%	29.12%	26.92%	24.81%	22.56%	19.34%
−0.50	32.41%	30.55%	28.59%	27.01%	25.71%	24.62%	23.99%
−0.25	30.76%	29.36%	28.00%	27.09%	26.54%	26.30%	26.42%
0.00	29.05%	28.01%	27.35%	27.16%	27.32%	27.74%	28.31%
0.25	27.32%	26.43%	26.63%	27.22%	28.04%	29.00%	29.95%
0.50	25.78%	24.46%	25.84%	27.28%	28.71%	30.14%	31.45%
0.75	25.14%	21.63%	24.98%	27.32%	29.33%	31.17%	32.86%

of this weakness and have been adjusting the BSM formula by using different volatilities for every strike and maturity, which is reflected in the market as a volatility smile. The SABR model is an extension of the Black-76 model to include an easily implementable stochastic volatility model. The SABR model is a two-factor model:

$$dF = \alpha F^{\beta} dz$$

$$d\alpha = \xi \alpha dw,$$

where F is the future/forward price, β is a constant deciding the distribution of the asset price, α is the volatility of the forward price, and ξ is the volatility of the volatility. dz and dw are two correlated Wiener processes. Using a singular perturbation technique, Hagan, Kumar, Lesniewski, and Woodward (2002) obtain an analytical solution for an input volatility as a function of the current forward price. The analytical input volatility can then be plugged directly into the Black-76 formula. The Black-76 equivalent volatility is given by

$$\sigma_B = \frac{\alpha}{(FX)^{(1-\beta)/2}\left(1 + \frac{(1-\beta)^2}{24}\ln(F/X)^2 + \frac{(1-\beta)^4}{1920}\ln(F/X)^4\right)} \frac{z}{\chi(z)}$$
$$\times \left[1 + \left(\frac{(1-\beta)^2}{24}\frac{\alpha^2}{(FX)^{1-\beta}} + \frac{1}{4}\frac{\rho\beta\xi\alpha}{(FX)^{(1-\beta)/2}} + \frac{2-3\rho^2}{24}\xi^2\right)\right] T, \tag{6.33}$$

where

$$z = \frac{\xi}{\alpha}(FX)^{(1-\beta)/2}\ln(F/X)$$

$$\chi(z) = \ln\left(\frac{\sqrt{1-2\rho z + z^2} + z - \rho}{1-\rho}\right)$$

In the case where the option is at-the-money forward ($F = X$), the formula above reduces to

$$\sigma_{B_{ATM}} = \frac{\alpha}{F^{1-\beta}}\left(\frac{(1-\beta)^2}{24}\frac{\alpha^2}{F^{2-2\beta}} + \frac{1}{4}\frac{\rho\beta\alpha\xi}{F^{1-\beta}} + \frac{2-3\rho^2}{24}\xi^2\right)T + \ldots$$

Special Cases

In the special case of $\beta = 1$, formula (6.33) can be simplified to

$$\sigma_B = \alpha\left(\frac{z}{\chi(z)}\right)\left[1 + \left(\frac{1}{4}\rho\alpha\xi + \frac{1}{24}(2-3\rho^2)\xi^2\right)T + \ldots\right],$$

where $\chi(z)$ is as before and

$$z = \frac{\xi}{\alpha}\ln(F/X)$$

In the special case where $\beta = 0$, formula (6.33) can be simplified to

$$\sigma_B = \alpha\frac{\ln(F/X)}{F-X}\left(\frac{z}{\chi(z)}\right)\left[1 + \left(\frac{\alpha^2}{24FX} + \frac{2-3\rho^2}{24}\xi^2\right)T + \ldots\right].$$

where $\chi(z)$ is as before and

$$z = \frac{\xi}{\alpha}\sqrt{FX}\ln(F/X)$$

Calibration

The beta parameter will typically be chosen a priori according to how the traders prefer to observe their market prices. For example, in the FX markets it is standard to assume lognormal terms, $\beta = 1$, while in some fixed-income markets traders prefer to assume normal terms, $\beta = 0$.

To calibrate the model, one will typically want the SABR equivalent Black-76 volatility to match the at-the-money volatility in the market. This involves calibrating the α based on the at-the-money volatility. As discussed in detail by West (2005a), it is important to recognize that the α is not the same as the at-the-money implied Black-76 volatility. Assuming the at-the-money Black-76 volatility is σ_{atm}, West (2005a) shows that α is the root of the cubic of

$$\frac{(1-\beta)^2 T}{24F^{2-2\beta}}\alpha^3 + \frac{\rho\beta v T}{4F^{1-\beta}}\alpha^2 + \left(1 + \frac{2-3\rho^2}{24}v^2 T - \sigma_{atm}F^{1-\beta}\right) = 0,$$

where we assume we already have solved for ρ and v. For typical inputs, this cubic has only one root. It is, however, possible for it to have as many as three roots, in which case we can use the smallest positive root. We want a fast and efficient algorithm to find the root. West (2005a) suggests the Viéte method (from 1916) for this job, using the implementation in Press, Teukolsky, Vetterling, and Flannery (1992). Any decent root finding algorithm will do though.

The correlation between asset prices and volatility can be calibrated to be consistent with the current volatility smile, or chosen by the trader based on his or her view of the market.

Application

Even if this model is relatively new, it is already in use by traders to manage and hedge option portfolios. For options with very short time to maturity, the volatility of volatility often has to be set to extreme levels to fit commonly observed volatility smiles. In that case, a model combining jumps with stochastic volatility would probably fit better.

Computer algorithm

The computer code below consists of several functions. *SABRVolatility*(·) returns the Black-76 equivalent SABR volatility. *alphaSABR*(·) and *CRoot*(·) are used to calibrate the α to the at-the-money Black-76 market volatility.

```
 Function SABRVolatility(F As Double, X As Double, T As Double, _
   ATMvol As Double, Beta As Double, _
   VolVol As Double, rho As Double) As Double

    SABRVolatility = alphaSABR(F, X, T, FindAlpha(F, X, T, ATMvol, _
        Beta, VolVol, rho), Beta, VolVol, rho)

End Function



Function alphaSABR(F As Double, X As Double, T As Double, Alpha As Double, _
    Beta As Double, VolVol As Double, rho As Double) As Double

  'the SABR skew vol function

  Dim dSABR(1 To 3) As Double
  Dim sabrz As Double, y As Double

  dSABR(1) = Alpha / ((F * X)^((1 - Beta) / 2) * (1 + (((1 - Beta)^2) / 24) _
    * (Log(F / X)^2) + ((1 - Beta)^4 / 1920) * (Log(F / X)^4)))

  If Abs(F - X) > 10^-8 Then
    sabrz = (VolVol / Alpha) * (F * X)^((1 - Beta) / 2) * Log(F / X)
    y = (Sqr(1 - 2 * rho * sabrz + sabrz^2) + sabrz - rho) / (1 - rho)
    If Abs(y - 1) < 10^-8 Then
      dSABR(2) = 1
    ElseIf y > 0 Then
      dSABR(2) = sabrz / Log(y)
    Else
      dSABR(2) = 1
    End If
  Else
```

```
    dSABR(2) = 1
  End If

  dSABR(3) = 1 + ((((1 - Beta)^2 / 24) * Alpha^2 / ((F * X)^(1 - Beta))) _
    + 0.25 * rho * Beta * VolVol * Alpha / ((F * X)^((1 - Beta) / 2)) _
    + (2 - 3 * rho^2) * VolVol^2 / 24) * T
  alphaSABR = dSABR(1) * dSABR(2) * dSABR(3)

End Function



Function FindAlpha(F As Double, X As Double, T As Double, ATMvol As Double, _
    Beta As Double, VolVol As Double, rho As Double) As Double

  'alpha is a function of atmvol etc.

  FindAlpha = CRoot((1 - Beta)^2 * T / (24 * F^(2 - 2 * Beta)), _
    0.25 * rho * VolVol * Beta * T / F^(1 - Beta), 1_
    + (2 - 3 * rho^2) / 24 * VolVol^2 * T, -ATMvol * F^(1 - Beta))

End Function

Function CRoot(cubic As Double, quadratic As Double, _
    linear As Double, constant As Double) As Double

  'finds the smallest positive root of the input cubic polynomial
  'algorithm from Numerical Recipes

  Dim roots(1 To 3) As Double
  Dim a As Double, b As Double, C As Double
  Dim r As Double, Q As Double
  Dim capA As Double, capB As Double, theta As Double

  a = quadratic / cubic
  b = linear / cubic
  C = constant / cubic
  Q = (a^2 - 3 * b) / 9
  r = (2 * a^3 - 9 * a * b + 27 * C) / 54

  If r^2 - Q^3 >= 0 Then
    capA = -Sgn(r) * (Abs(r) + Sqr(r^2 - Q^3))^(1 / 3)
    If capA = 0 Then capB = 0 Else capB = Q / capA
    CRoot = capA + capB - a / 3
  Else

    theta = ArcCos(r / Q^1.5)

    ' The three roots;
    roots(1) = -2 * Sqr(Q) * Cos(theta / 3) - a / 3
    roots(2) = -2 * Sqr(Q) * Cos(theta / 3 + 2.0943951023932) - a / 3
    roots(3) = -2 * Sqr(Q) * Cos(theta / 3 - 2.0943951023932) - a / 3

    'locate that one which is the smallest positive root
    'assumes there is such a root (true for SABR model)
    'there is always a small positive root

    If roots(1) > 0 Then
      CRoot = roots(1)
```

```
    ElseIf roots(2) > 0 Then
      CRoot = roots(2)
    ElseIf roots(3) > 0 Then
      CRoot = roots(3)
    End If

    If roots(2) > 0 And roots(2) < CRoot Then
      CRoot = roots(2)
    End If
    If roots(3) > 0 And roots(3) < CRoot Then
      CRoot = roots(3)
    End If
  End If
End Function

Function ArcCos(y As Double) As Double

  ArcCos = Atn(-y / Sqr(-y * y + 1)) + 2 * Atn(1)

End Function
```

Table 6-9 shows volatilities generated by the SABR model for varying correlations between the asset price and volatility. As expected, a negative correlation yields a negatively sloped volatility skew, while a positive correlation yields a positive skew. The α in the model is calibrated to the at-the-money volatility.

Example

Consider a European put option on a forward with forward price at 100. The strike is 70, time to maturity is six months, and the risk-free rate is 5%. Assume the α is found to be 30%, the volatility of the volatility is 50%, and the correlation between the volatility and the underlying asset is −0.4. What is the value of the option assuming a lognormal stochastic volatility model ($\beta = 1$)? $F = 100$, $X = 70$, $T = 0.5$, $r = 0.05$, $\alpha = 0.3$, $\xi = 0.5$, $\rho = -0.4$.

TABLE 6-9

SABR Black-76 Equivalent Volatilities

($S = 100$, $T = 0.5$, $\sigma_{atm} = 30\%$, $\beta = 0.999$, $\xi = 50\%$)

ρ	$X = 70$	$X = 80$	$X = 90$	$X = 100$	$X = 110$	$X = 120$	$X = 130$
−0.75	36.68%	34.18%	31.97%	30.00%	28.26%	26.74%	25.46%
−0.50	35.17%	33.11%	31.40%	30.00%	28.90%	28.07%	27.51%
−0.25	33.51%	31.95%	30.80%	30.00%	29.52%	29.31%	29.30%
0.00	31.65%	30.68%	30.16%	30.00%	30.13%	30.46%	30.92%
0.25	29.55%	29.29%	29.48%	30.00%	30.71%	31.54%	32.41%
0.50	27.09%	27.74%	28.78%	30.00%	31.28%	32.55%	33.78%
0.75	24.04%	25.98%	28.03%	30.00%	31.83%	33.51%	35.05%

First, we calculate the SABR Black-76 equivalent volatility:

$$z = \frac{0.5}{0.3}(100 \times 70)^{(1-1)/2}\ln(100/70) = 0.5945$$

$$x(z) = x(0.5945) = \ln\left(\frac{\sqrt{1 - 2 \times (-0.4) \times 0.5945 + 0.5945^2} + 0.5945 - (-0.4)}{1 - (-0.4)}\right) = 0.5166$$

$$\sigma_B = 0.3\left(\frac{0.5945}{x(z)}\right)\left[1 + \left(\frac{1}{4}(-0.4) \times 0.3 \times 0.5 + \frac{1}{24}(2 - 3(-0.4)^2)0.5^2\right)0.5\right] = 0.3454$$

Next, we simply plug this volatility into the Black-76 formula:

$$d_1 = \frac{\ln(100/75) + 0.5 \times 0.3454^2/2}{0.3454\sqrt{0.5}} = 1.5827$$

$$d_2 = 1.5827 - 0.3454\sqrt{0.5} = 1.3385$$

$$N(-d_1) = N(-1.5827) = 0.0567, \quad N(-d_2) = N(-1.3385) = 0.0904$$

$$p = e^{-0.05\times0.5}[70N(-d_2) - 100N(-d_1)] = 0.6352$$

6.11 VARIANCE AND VOLATILITY SWAPS

Variance and volatility swaps are excellent derivatives for hedging and speculation on the realized volatility levels of an asset, as well as for trading the spread between realized and implied volatility. In the equity market there is a relatively liquid market for OTC variance swaps. In the interbank currency market, most investment banks and larger commercial banks are quoting volatility swaps. We will take a quick look at how to value both variance and volatility swaps.

6.11.1 Variance Swaps

A variance swap offers a payoff at maturity equal to the difference between the realized variance over the swap period and the contract variance, multiplied by a notional. To hedge a variance swap with plain vanilla options, we want to keep exposure to variance invariant to the level of the spot—spot-independent variance vega. From the BSM formula we know that the variance vega of a plain European option is given by[6]

$$\text{Variance vega} = Se^{(b-r)T}n(d_1)\frac{\sqrt{T}}{2\sigma}$$

[6]See Chapter 2 for more details on variance vega.

From this formula it is apparent that the variance vega of a plain option can be highly sensitive to changes in spot, and that we cannot hedge a variance swap with an option with a single strike. What we need is an option portfolio that gives us a portfolio variance vega that is independent of the spot level. Based on this idea, Carr and Madan (1998) and Demeterifi, Derman, Kamal, and Zou (1999) have come up with a hedge and a fair variance, X_{Var}, for a variance swap:

$$X_{Var} = \frac{2}{T}\left(rT - \frac{S}{\hat{S}}e^{rT} + 1 - \ln(\hat{S}/S) + e^{rT}\int_0^{\hat{S}} \frac{1}{X^2}p(x)dX + e^{rT}\int_{\hat{S}}^{\infty} \frac{1}{X^2}c(x)dX\right), \tag{6.34}$$

where $\hat{S}$ is an asset price level typically selected to be equal to current spot, or alternatively at-the-money forward S^{bT}. This can be regarded as the market's expectation of future realized volatilities. The variance swap can be hedged by a short forward contract and a portfolio of options all expiring at the same time as the variance-swap T:

- A short position in $\frac{1}{\hat{S}}$ forward contracts struck at $\hat{S}$
- A long position in $\frac{1}{X^2}$ put options with strike X, for all strikes from 0 to $\hat{S}$
- A long position in $\frac{1}{X^2}$ call options with strike X, for all strikes from $\hat{S}$ to infinite

Even if equation (6.34) and its hedge involves the BSM formula, it actually holds also in the presence of a volatility smile or skew. Unfortunately, the perfect hedge requires a portfolio of options with an infinite number of strikes (from zero to ∞), which is not practically possible. In reality, we have only a few strikes to select from. Demeterifi, Derman, Kamal, and Zou (1999) describe a practical approximation for replicating a variance swap using a limited number of options. Rewriting equation (6.34), we have

$$X_{Var} = \frac{2}{T}\left(rT - \frac{S}{\hat{S}}e^{rT} + 1 - \ln(\hat{S}/S)\right) + e^{rT}\Pi_{\text{CP}}, \tag{6.35}$$

where Π_{CP} is the value of the portfolio of options with payoff at expiration equal to

$$f(S_T) = \frac{2}{T}\left(\frac{S_T - \hat{S}}{\hat{S}} - \ln(S_T/\hat{S})\right)$$

Suppose you can trade call options with strikes $X_{i,c}$ such that $X_{0,c} = \hat{S} < X_{1,c} < X_{2,c} < X_{3,c} \ldots < X_{n,c}$ and put options with strikes

$X_{i,p}$ such that $X_{0,p} = \hat{S} > X_{1,p} > X_{2,p} > X_{3,p} \ldots > X_{n,p}$. The weights given to each strike in the replicating portfolio is given by

$$w_c(X_{n,c}) = \frac{f(X_{n+1,c}) - f(X_{n,c})}{X_{n+1,c} - X_{n,c}} - \sum_{i=0}^{n-1} w_c(X_{i,c})$$

$$w_p(X_{n,p}) = \frac{f(X_{n+1,p}) - f(X_{n,p})}{X_{n,p} - X_{n+1,p}} - \sum_{i=0}^{n-1} w_p(X_{i,p})$$

We can now simply find the value of the portfolio as

$$\Pi_{CP} \approx \sum_{i=0}^{n_c} w_p(X_{i,p}) p_{BSM}(S, X_{i,p}) + \sum_{i=0}^{n_p-1} w_c(X_{i,c}) c_{BSM}(S, X_{i,c}), \tag{6.36}$$

where n_c is number of call options used and n_p is number of put options used. The best way to understand how to calculate the fair value of the variance swap is by taking a look at the example below together with the Excel spreadsheet on the accompanying CD.

Example

Assume we want to value a variance swap with three months to maturity on a stock currently trading at 100. The strikes and implied volatilities for plain vanilla options is given in Table 6-10. For simplicity, we have a linear volatility skew where the volatility increases by one percentage point for every strike we move downward. The contribution column is the weights multiplied by the option values. The sum of the contribution column is the value of the option portfolio, $\Pi_{CP} = 419.6756$. We now have all we need to find the fair variance using equation (6.35): $S = 100$, $T = 0.25$, $r = 0.05$, $b = 0.05$,

$$X_{Var} = \frac{2}{0.25}\left(0.05 \times 0.25 - \frac{100}{100}e^{0.05\times 0.25} + 1 - \ln(100/100)\right) + e^{0.05\times 0.25} \times 419.6756 = 424.9539$$

Converted into an "equivalent" fair volatility, we have $\sqrt{424.9539}/100 = 20.61\%$—that is, the fair variance of the variance-swap is $\sigma^2 = 0.2061^2$.

Practical Tip When trading variance or volatility swaps in the OTC market, make sure you ask for the exact terms before you compare prices from different banks. This naturally holds for any OTC derivatives contract but is particularly true for variance and volatility swaps, since at current writing (2006), banks do not seem to have settled on a set of standardized contracts. For instance, how many days they use for annualization, and the exact formula used for calculating the realized variance or volatility, are typically important.

TABLE 6-10

Variance Swap Calculation

$(S = 100, T = 0.25, r = 0.05, b = 0.05)$

Strike	Implied Volatility	Weights	Option Value	$f(S, X, T)$	Contribution	Variance Vega
Put options:						
45				1.9881		
50	30.00%	160.8054	0.000003	1.5452	0.0004	0.0000
55	29.00%	132.7808	0.00003	1.1827	0.0039	0.0000
60	28.00%	111.4987	0.0002	0.8866	0.0265	0.0003
65	27.00%	94.9558	0.0015	0.6463	0.1387	0.0013
70	26.00%	81.8416	0.0072	0.4534	0.5874	0.0046
75	25.00%	71.2696	0.0291	0.3015	2.0732	0.0137
80	24.00%	62.6224	0.0996	0.1851	6.2387	0.0336
85	23.00%	55.4593	0.2933	0.1002	16.2687	0.0691
90	22.00%	49.4591	0.7520	0.0429	37.1913	0.1189
95	21.00%	44.3828	1.6932	0.0103	75.1503	0.1695
100	20.00%	20.6927	3.3728	0.0000	69.7918	0.1016
Call options:						
100	20.00%	19.3574	4.6150	0.0000	89.3341	0.0951
105	19.00%	36.3224	2.2886	0.0097	83.1277	0.1803
110	18.00%	33.0920	0.9073	0.0375	30.0243	0.1250
115	17.00%	30.2744	0.2670	0.0819	8.0833	0.0617
120	16.00%	27.8019	0.0529	0.1414	1.4694	0.0198
125	15.00%	25.6205	0.0062	0.2149	0.1577	0.0037
130	14.00%	23.6862	0.0004	0.3011	0.0083	0.0003
135	13.00%	21.9629	0.000007	0.3992	0.0002	0.0000
140				0.5082		
Sum:					**419.6756**	**0.9984**

6.11.2 Volatility Swaps

At expiration a volatility swap pays out the difference between the realized volatility covered by the swap and the contract volatility. In contrast to a variance swap, a volatility swap is linear in payoff as a function of the realized volatility. As has been noted previously in the literature (see, for example, Demeterifi, Derman, Kamal, and Zou, 1999; Brockhaus and Long, 2000), a convexity adjustment is needed to value a volatility swap, compared to a variance swap. Here we will limit ourselves to a closed-form valuation method for variance and volatility swaps assuming the variance follows a GARCH(1,1) process, based on a paper by Haug, Javaheri, and Wilmott (2004).

GARCH Volatility Swaps

Even when there is no closed-form solution for the expected volatility, one can use the Brockhaus and Long (2000) approximation (which is a

Taylor expansion of order two, of the square-root function on variable v around the point $v_0 = E[v]$) to calculate

$$E[\sqrt{v}] \approx \sqrt{E[v]} - \frac{\mathrm{Var}[v]}{8E[v]^{3/2}}$$

The GARCH Process

Let us assume the variance in a continuous version is defined by the GARCH(1,1) process:

$$dv = \kappa(\theta - v)dt + \gamma v\, dX,$$

where v is the variance, κ is the speed of mean reversion, θ is the mean reversion level, and γ is the volatility of volatility, or more precisely, the volatility of the square of volatility. The discrete version of the GARCH(1,1) process is described in Engle and Mezrich (1995) as

$$v_{n+1} = (1 - \alpha - \beta)V + \alpha u_n^2 + \beta v_n,$$

where V is the long-term variance, u_n is the drift adjusted stock return at time n, α is the weight assigned to u_n^2, and β is the weight assigned to v_n.[7] We moreover have the following relationship

$$\theta = \frac{V}{dt}$$
$$\kappa = \frac{1 - \alpha - \beta}{dt}$$
$$\gamma = \alpha\sqrt{\frac{\xi - 1}{dt}},$$

where[8] ξ is the Pearson kurtosis (fourth moment) of $u[n]$.

The problem for F can be written as

$$\frac{\partial F}{\partial t} + \frac{1}{2}\gamma^2 v^2 \frac{\partial^2 F}{\partial v^2} + \kappa(\theta - v)\frac{\partial F}{\partial v} + v\frac{\partial F}{\partial I} = 0$$

[7]The GARCH(1,1) model implies that the stock price process and the volatility process contain two uncorrelated Brownian motions. In an NGARCH process, described by Engle and Ng (1993), we have

$$v_{n+1} = (1 - \alpha - \beta)V + \alpha(u_n - c)^2 + \beta v_n$$

where c is another parameter to be estimated and creates the correlation between the two processes. This will not, however, affect the results we present, and we concentrate on the GARCH(1,1). The NGARCH process is discussed, for instance, in Ritchken and Trevor (1997).

[8]The γ in the Engle and Mezrich (1995) paper is $\gamma = \alpha\sqrt{(\xi - 1)dt}$, which is different from the one presented here. This is likely due to a small typo in their paper. See also Nelson (1990).

with $F(v, I, t) = I$, where

$$I_t = \int_0^t v_u du$$

The solution for $F(v, I, t)$ is now

$$F(v, I, t) = \theta \left(T - t + \frac{e^{-\kappa(T-t)} - 1}{\kappa} \right) + \frac{1}{\kappa} \left(1 - e^{-\kappa(T-t)} \right) v + I$$

Similarly, $G(v, I, t)$ has solution

$$G(v, I, t) = f(t) + g(t)v + h(t)v^2 + l(t)I + n(t)vI + I^2$$

with

$$f(t) = \theta^2 (T-t)^2 - \frac{4\theta^2(\gamma^2 - \kappa)}{\kappa(\gamma^2 - 2\kappa)} \left(T - t + \frac{e^{-\kappa(T-t)} - 1}{\kappa} \right)$$

$$- \frac{4\theta^2 \kappa^2}{(\gamma^2 - \kappa)^2 (\gamma^2 - 2\kappa)} \left(\frac{1 - e^{(\gamma^2 - 2\kappa)(T-t)}}{(\gamma^2 - 2\kappa)} + \frac{1 - e^{-\kappa(T-t)}}{\kappa} \right)$$

$$- \frac{2\theta^2(\gamma^2 + \kappa)}{\gamma^2 - \kappa} \left(e^{-\kappa(T-t)} \frac{T-t}{\kappa} + \frac{1}{\kappa^2}(e^{-\kappa(T-t)} - 1) \right)$$

$$g(t) = \frac{2\theta}{\kappa}(T-t) - \frac{4\theta(\gamma^2 - \kappa)}{\kappa^2(\gamma^2 - 2\kappa)} \left(1 - e^{-\kappa(T-t)} \right)$$

$$+ \frac{4\theta\kappa}{(\gamma^2 - \kappa)^2(\gamma^2 - 2\kappa)} \left(e^{(\gamma^2 - 2\kappa)(T-t)} - e^{-\kappa(T-t)} \right)$$

$$+ \frac{2\theta(\gamma^2 + \kappa)}{\kappa(\gamma^2 - \kappa)} (T-t) e^{-\kappa(T-t)}$$

$$h(t) = \frac{2}{\kappa(\gamma^2 - 2\kappa)} \left(e^{(\gamma^2 - 2\kappa)(T-t)} - 1 \right) - \frac{2}{\kappa(\gamma^2 - \kappa)} \left(e^{(\gamma^2 - 2\kappa)(T-t)} - e^{-\kappa(T-t)} \right)$$

$$l(t) = 2\theta \left(T - t + \frac{e^{-\kappa(T-t)} - 1}{\kappa} \right)$$

$$n(t) = \frac{2}{\kappa} \left(1 - e^{-\kappa(T-t)} \right)$$

Numerical Examples

Let us assume we are calibrating the GARCH parameters from five years of daily historic S&P 500 (SPX) prices (from 01/10/1996 to 09/28/2001). This gives us a long-term variance of $V = 0.00015763$

(so the annualized long-term volatility is around 19.93%) and a Pearson kurtosis of 5.81175. The discrete GARCH(1,1) parameters are $\alpha = 0.127455$ and $\beta = 0.7896510$. We take for inputs $v = 0.0361$ (corresponding to a volatility of 19%) and $I = 0$ (on the issue date of volatility swap). We use business daily data and, therefore, $dt = 1/252$. Further, let us assume six months to swap maturity. We now have the input we need to calculate the diffusion limit parameters θ, κ, and γ:

$$\theta = \frac{V}{dt} = \frac{0.00015763}{1/252} = 0.0397228$$

$$\kappa = \frac{1-\alpha-\beta}{dt} = \frac{1 - 0.127455 - 0.7896510}{1/252} = 20.889288$$

$$\gamma = \alpha\sqrt{\frac{\xi - 1}{dt}} = 0.127455\sqrt{\frac{5.81175 - 1}{1/252}} = 4.4382185$$

and

$$\begin{aligned}
f(t) &= 0.03972^2 \times 0.5^2 - \frac{4 \times 0.03972^2(4.4382^2 - 20.8893)}{20.8893(4.4382^2 - 2 \times 20.8893)}\left(0.5 + \frac{e^{-20.8893\times0.5} - 1}{20.8893}\right) \\
&\quad - \frac{4 \times 0.0397^2 \times 20.8893^2}{(4.4382^2 - 20.8893)^2(4.4382^2 - 2 \times 20.8893)} \\
&\quad \times \left(\frac{1 - e^{(4.4382^2-2\times20.8893)0.5}}{(4.4382^2 - 2 \times 20.8893)} + \frac{1 - e^{-20.8893\times0.5}}{20.8893}\right) \\
&\quad - \frac{2 \times 0.0397^2(4.4382^2 + 20.8893)}{4.4382^2 - 20.8893} \\
&\quad \times \left(e^{-20.8893\times0.5}\frac{0.5}{20.8893} + \frac{1}{20.8893^2}(e^{-20.8893\times0.5} - 1)\right) = 0.000368
\end{aligned}$$

$$\begin{aligned}
g(t) &= \frac{2 \times 0.0397}{20.8893}0.5 - \frac{4 \times 0.0397(4.4382^2 - 20.8893)}{20.8893^2(4.4382^2 - 2 \times 20.8893)}\left(1 - e^{-20.8893\times0.5}\right) \\
&\quad + \frac{4 \times 0.0397 \times 20.8893}{(4.4382^2 - 20.8893)^2(4.4382^2 - 2 \times 20.8893)} \\
&\quad \times \left(e^{(4.4382^2-2\times20.8893)0.5} - e^{-20.8893\times0.5}\right) \\
&\quad + \frac{2 \times 0.0397(4.4382^2 + 20.8893)}{20.8893(4.4382^2 - 20.8893)}0.5e^{-20.8893\times0.5} = 0.001881
\end{aligned}$$

$$\begin{aligned}
h(t) &= \frac{2}{20.8893(4.4382^2 - 2 \times 20.8893)}\left(e^{(4.4382^2-2\times20.8893)0.5} - 1\right) \\
&\quad - \frac{2}{20.8893(4.4382^2 - 20.8893)}\left(e^{(4.4382^2-2\times20.8893)0.5} - e^{-20.8893\times0.5}\right) = 0.004335,
\end{aligned}$$

$$l(t) = 2 \times 0.0397 \left(0.5 + \frac{e^{-20.8893\times 0.5} - 1}{20.8893}\right) = 0.035920$$

$$n(t) = \frac{2}{20.8893}\left(1 - e^{-20.8893\times 0.5}\right) = 0.095740$$

The result for $F(v, I, t)$ is now

$$F(v, I, t) = 0.0397\left(0.5 + \frac{e^{-20.8893\times 0.5} - 1}{20.8893}\right)$$

$$+ \frac{1}{20.8893}\left(1 - e^{-20.8893\times 0.5}\right) 0.0361 + 0 = 0.039376$$

Similarly, $G(v, I, t)$ returns

$$G(v, I, t) = f(t) + g(t)v + h(t)v^2 + l(t) \times 0 + n(t)0.0361 \times 0 + 0^2 = 0.001765$$

We now have all we need to calculate the convexity adjustment, and we can find the fair GARCH(1,1) value of the volatility swap. The variance is given by

$$\mathrm{Var}[I] = G - F^2 = 0.001765 - 0.039376^2 = 0.000215$$

The unadjusted volatility swap is thus given by $\sqrt{\mathrm{Var}[I]} = 0.1984337$. Further, the volatility swap with convexity adjustment is given by

$$E[\sqrt{v}] \approx \sqrt{0.039376} - \frac{0.000215}{8 \times 0.039376^{3/2}} = 0.1949984$$

The fair value of the volatility swap is thus 19.50% versus the unadjusted volatility of 19.84%. An Excel spreadsheet with VBA code to perform such calculation is included on the accompanying CD. See Haug, Javaheri, and Wilmott (2004) for more information on how to value and hedge volatility swaps.

6.12 MORE INFORMATION

The number of papers on stochastic volatility and jump-diffusion models is huge. A good start for more information about and references to stochastic volatility models are the excellent books by Fouque, Papanicolaou, and Sircar (2000), Lewis (2000), Rebonato (2004), Javaheri (2005), and Gatheral (2006).

CHAPTER

7

TREES AND FINITE DIFFERENCE METHODS

If you are continuously confused it's a state you should get used to.

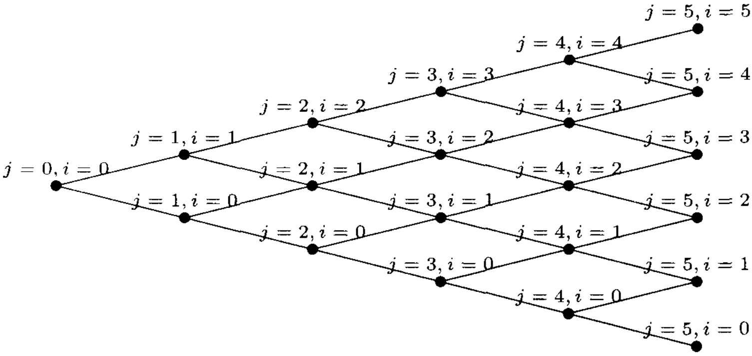
Finance Professor

This chapter describes numerical methods that are useful to price options and other derivative securities. The numerical methods are more flexible than analytical solutions and can be used to price a wide range of options contracts for which there are no known analytical solutions. The chapter starts out with binomial and trinomial trees for pricing options that depend on one underlying asset. The chapter next covers a method for pricing options on two correlated assets: the three-dimensional binomial model. The chapter then moves on to option pricing using so-called implied trees. The last part of the chapter describes finite difference methods, including explicit and implicit methods, and the the Crank-Nicolson method.

7.1 BINOMIAL OPTION PRICING

The binomial method is certainly the most widely used numerical method to price American options on stocks, futures, and currencies. The binomial method was first published by Cox, Ross, and Rubinstein (1979) and Rendleman and Bartter (1979). They explained how to construct a recombining binomial tree that discretizes and approximates the geometric Brownian motion. At the limit, a binomial tree (with a very large number of time steps) is equivalent to the continuous-time Black-Scholes-Merton formula used when pricing European options. Of more interest, the binomial model easily handles the pricing of American options, where no closed-form solution exists, as well as many exotic options.

The asset price in a binomial tree can over a time step Δt either increase by a fixed amount u with a probability p or decrease by a fixed amount d with a probability $1 - p$. The number of time steps is n. In all the tree models of this chapter, we start counting the first node as zero. The number of time steps to a node in the tree we define as j. The number of times the asset price has gone up to reach a node we will define as i (price step). The first node in the tree will be assigned $(j = 0, i = 0)$. If the asset price goes up at the second node, it will be assigned $(j = 1, i = 1)$. If the asset price went down at the first time step, we have $(j = 1, i = 0)$, this is best illustrated in a figure:



The number of paths leading to a node (j, i) is equal to

$$\frac{j!}{i!(j-i)!},$$

and the corresponding probability of reaching node (j, i) is

$$\frac{j!}{i!(j-i)!}p^i(1-p)^{j-i}$$

To price European plain vanilla call or put options, we are only concerned about the end nodes, n, and the binomial model can be expressed as

$$c = e^{-rT}\sum_{i=0}^{n}\left(\frac{n!}{i!(n-i)!}\right)p^i(1-p)^{n-i}\max[Su^i d^{n-i} - X, 0] \tag{7.1}$$

$$p = e^{-rT}\sum_{i=0}^{n}\left(\frac{n!}{i!(n-i)!}\right)p^i(1-p)^{n-i}\max[X - Su^i d^{n-i}, 0] \tag{7.2}$$

Many of the nodes will be out-of-the-money, and instead of starting to count from the lowest node $i = 0$, we can make the algorithm more efficient and count from a, (for a call option), the smallest nonnegative integer greater than $\frac{\ln(X/(Sd^n))}{\ln(u/d)}$. This gives us

$$c = e^{-rT} \sum_{i=a}^{n} \left(\frac{n!}{i!(n-i)!} \right) p^i (1-p)^{n-i} (Su^i d^{n-i} - X) \tag{7.3}$$

$$p = e^{-rT} \sum_{i=0}^{a-1} \left(\frac{n!}{i!(n-i)!} \right) p^i (1-p)^{n-i} (X - Su^i d^{n-i}) \tag{7.4}$$

The up and down jump factors and corresponding probabilities are chosen to match the first two moments of the stock price distribution (mean and variance). There are, however, more unknowns than there are equations in this set of restrictions, implying that there are many ways of choosing the parameters and still satisfy the moment restrictions. Cox, Ross, and Rubinstein (1979) (CRR) set the up and down parameters to

$$u = e^{\sigma\sqrt{\Delta t}} \qquad d = e^{-\sigma\sqrt{\Delta t}}, \tag{7.5}$$

where $\Delta t = T/n$ is the length of each time step (time between price movements) and n is the number of time steps. The probability of the stock price increasing at the next time step is

$$p = \frac{e^{b\Delta t} - d}{u - d} \tag{7.6}$$

Computer algorithm

This computer code implements a binomial tree for standard European options, using the CRR parameterization.

```
Function EuropeanBinomialPlainVanilla(CallPutFlag As String, S As Double, _
    X As Double, T As Double, r As Double, _
    b As Double, v As Double, n As Long) As Double

        Dim u As Double, d As Double, p As Double
        Dim sum As Double, dt As Double, a As Double
        Dim j As Integer

        dt = T / n
        u = Exp(v * Sqr(dt))
        d = 1 / u
        p = (Exp(b * dt) - d) / (u - d)
        a = Int(Log(X / (S * d^n)) / Log(u / d)) + 1

    sum = 0
    If CallPutFlag = "c" Then
        For j = a To n
            sum = sum + Application.Combin(n, j) * p^j * (1 - p)^(n - j) _
        * (S * u^j * d^(n - j) - X)
        Next
    ElseIf CallPutFlag = "p" Then
         For j = 0 To a - 1
            sum = sum + Application.Combin(n, j) * p^j * (1 - p)^(n - j) _
```

```
            * (X - S * u^j * d^(n - j))
        Next
    End If
    EuropeanBinomialPlainVanilla = Exp(-r * T) * sum

End Function
```

Generalized European Binomial

The more general case of the European binomial model is

$$c = e^{-rT} \sum_{i=0}^{n} \left(\frac{n!}{i!(n-i)!} \right) p^i (1-p)^{n-i} g[S(T), X] \tag{7.7}$$

$$p = e^{-rT} \sum_{i=0}^{n} \left(\frac{n!}{i!(n-i)!} \right) p^i (1-p)^{n-i} g[S(T), X], \tag{7.8}$$

where $S(T) = Su^i d^{n-i}$ and $g[S(T), X]$ is any given payoff function at maturity. This illustrates how powerful the simple binomial model is. It is capable of pricing any European option on a single asset, whose payoff is not path-dependent. If we, for instance, want to find the value of a power option where the payoff at maturity is $\max[S^2 - X, 0]$, then we simply replace $g[S(T), X]$ with $\max[(Su^i d^{n-i})^2 - X, 0]$. We will later also discuss how this model can be used to price many path-dependent options. The following payoff table gives some examples of options that can be valued with the above expressions (all included on the accompanying CD). The variable z equals 1 if the contract is a call, and -1 if it is a put. A * next to a payoff signifies that a closed-form solution for the option is included in this book. A ** signifies that a closed-form solution is included in the book but that the binomial method is more flexible. For instance, the closed-form solution for powered options restricts i to be an integer, while the binomial method allows i to take any real value. The Cap means that the payoff is capped, with maximum payoff equal to Cap.

Power contract:	: S^i
Capped power contract:	: $\min[S^i, \text{Cap}]$
Power contract*:	: $(S/X)^i$
Power contract:	: $(S - X)^i$
Standard power option*:	: $\max[z(S^i - X), 0]$
Capped standard power option*:	: $\min[\max[z(S^i - X), 0], \text{Cap}]$
Powered option**:	: $\max[z(S - X), 0]^i$
Capped powered option:	: $\min[\max[z(S - X), 0]^i, \text{Cap}]$
Sinus option:	: $\max[z(\sin(S) - X), 0]$

Cosinus option:	: $\max[z(\cos(S) - X), 0]$
Tangens option:	: $\max[z(\tan(S) - X), 0]$
Log contract*:	: $\ln(S)$
Log contract*:	: $\ln(S/X)$
Log option*:	: $\max[\ln(S/X), 0]$
Square root contract:	: $\sqrt{S}$
Square root contract:	: $\sqrt{S/X}$
Square root option:	: $\sqrt{\max[z(S - X), 0]}$

Computer algorithm

Here is the VBA code for a flexible European binomial valuation model. It can value 18 different derivatives contracts, and you can easily add a new derivatives instrument by simply adding a line in the payoff function. The *TypeFlag* takes integer values that decide what type of option should be valued. You set the *CallPutFlag* = "*c*" for a call option and to "*p*" for a put option. *pow* is the power the power options are raised to. *cap* is the cap on the payoff for any capped option. *n* is the number of time steps used for valuation. Noncapped power options are in general very slow to converge.

```
Function EuropeanBinomial(TypeFlag As Integer, CallPutFlag As String, _
  S As Double, X As Double, T As Double, r As Double, b As Double, _
  v As Double, pow As Double, cap As Double, n As Long) As Double

    Dim u As Double, d As Double, p As Double
    Dim sum As Double, dt As Double, Si As Double, z As Integer
    Dim j As Long

    z = 1
    If CallPutFlag = "p" Then z = -1

    dt = T / n
    u = Exp(v * Sqr(dt))
    d = 1 / u
    p = (Exp(b * dt) - d) / (u - d)

    sum = 0

    For j = 0 To n
            Si = S * u^j * d^(n - j)
            sum = sum + Application.Combin(n, j) * p^j _
    * (1 - p)^(n - j) * BinomialPayoff(TypeFlag, z, Si, X, pow, cap)
    Next

    EuropeanBinomial = Exp(-r * T) * sum

End Function


Function BinomialPayoff(TypeFlag As Integer, z As Integer, _
    S As Double, X As Double, pow As Double, cap As Double) As Double
```

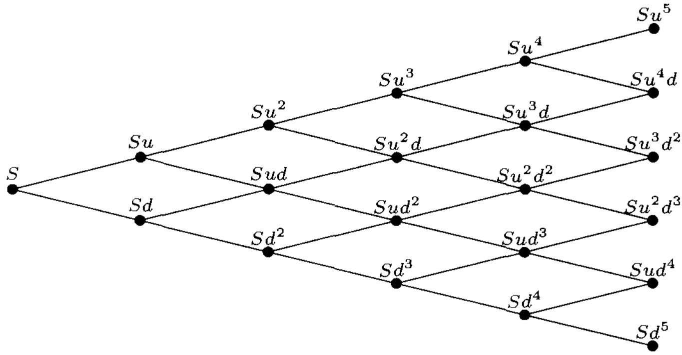
```
If TypeFlag = 1 Then 'Plain Vanilla
    BinomialPayoff = Max(z * (S - X), 0)
  ElseIf TypeFlag = 2 Then ' Power contract
    BinomialPayoff = S^pow
  ElseIf TypeFlag = 3 Then ' Capped Power contract
    BinomialPayoff = Min(S^pow, cap)
  ElseIf TypeFlag = 4 Then ' Power contract
    BinomialPayoff = (S / X)^pow
ElseIf TypeFlag = 5 Then ' Power contract
    BinomialPayoff = z * (S - X)^pow
ElseIf TypeFlag = 6 Then 'Standard power option
    BinomialPayoff = Max(z * (S^pow - X), 0)
  ElseIf TypeFlag = 7 Then 'Capped power option
    BinomialPayoff = Min(Max(z * (S^pow - X), 0), cap)
ElseIf TypeFlag = 8 Then ' Powered option
    BinomialPayoff = Max((z * (S - X)), 0)^pow
ElseIf TypeFlag = 9 Then ' Capped powered option
    BinomialPayoff = Min(Max((z * (S - X)), 0)^pow, cap)
  ElseIf TypeFlag = 10 Then ' Sinus option
    BinomialPayoff = Max(z * (Sin(S) - X), 0)
  ElseIf TypeFlag = 11 Then ' Cosinus option
    BinomialPayoff = Max(z * (Cos(S) - X), 0)
  ElseIf TypeFlag = 12 Then ' Tangens option
    BinomialPayoff = Max(z * (Tan(S) - X), 0)
  ElseIf TypeFlag = 13 Then ' Log contract
    BinomialPayoff = Log(S)
  ElseIf TypeFlag = 14 Then ' Log contract
    BinomialPayoff = Log(S / X)
  ElseIf TypeFlag = 15 Then ' Log option
    BinomialPayoff = Max(Log(S / X), 0)
  ElseIf TypeFlag = 16 Then 'Square root contract
    BinomialPayoff = Sqr(S)
ElseIf TypeFlag = 17 Then 'Square root contract
    BinomialPayoff = Sqr(S / X)
ElseIf TypeFlag = 18 Then 'Square root option
    BinomialPayoff = Sqr(Max(z * (S - X), 0))
End If

End Function
```

7.1.1 Cox-Ross-Rubinstein American Binomial Tree

Here we will look at how to use the Cox-Ross-Rubinstein binomial tree to value American-style options. The asset price at each node is set equal to

$$Su^i d^{j-i}, \qquad i = 0, 1, \ldots, j,$$

where the up and down jump factors for a time interval $\Delta t = T/n$ is given by (7.5), where n is the number of time steps, as before. The probability of the stock price increasing by the factor u is now given by equation (7.6). Since probabilities must sum to unity, the probability of the stock price decreasing by the factor d must be $1 - p$. Again, the up and down factors and probabilities are chosen to match the first two moments of the stock price distribution. This ensures that the probability distribution implied by the binomial tree converges to geometric Brownian motion when Δt goes to zero.



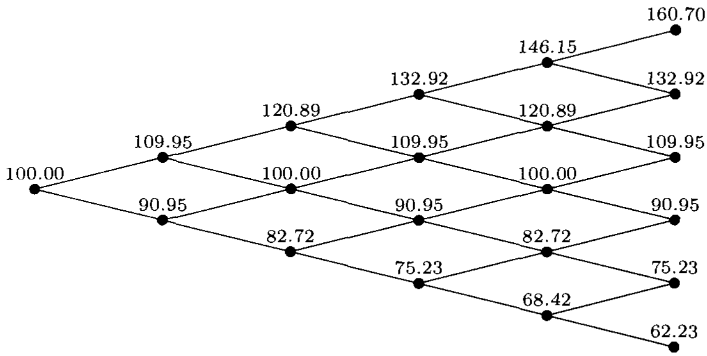
Example

Consider an American stock put option with six months to expiration. The stock price is 100, the strike price is 95, the risk-free interest rate is 8%, and the volatility is 30%. The option is priced in a binomial tree with five time steps. $S = 100$, $X = 95$, $T = 0.5$, $r = b = 0.08$, $\sigma = 0.3$, and $n = 5$.

$$\Delta t = \frac{0.5}{5} = 0.1$$

$$u = e^{0.3\sqrt{0.1}} = 1.0995 \qquad d = e^{-0.3\sqrt{0.1}} = 0.9095$$

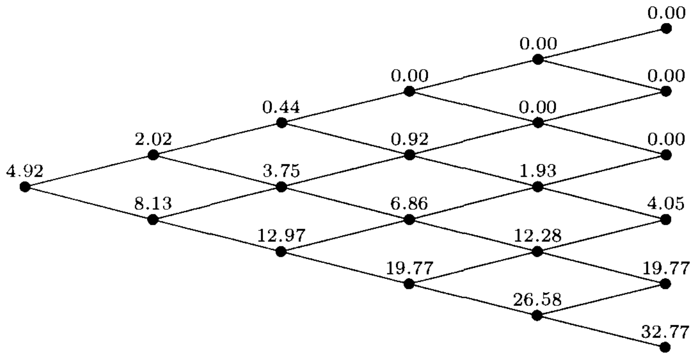
$$p = \frac{e^{0.08\times 0.1} - 0.9095}{1.0995 - 0.9095} = 0.5186$$



First, we start at the end of the tree to see if it is optimal to exercise the option $\max[X - S, 0]$. For example, in the end node with asset price 62.23, it is naturally optimal to exercise the put option: $\max[95 - 62.23, 0] = 32.77$, while at, for example, the end node with asset price

100.95, it is not optimal to exercise the option max[95 − 100.95, 0] = 0. After checking for optimal exercise at each end node, we can now easily find the value of the American put option by standard backward induction (rolling back through the tree), where we check at each node if it is optimal with early exercise:

$$P_{j,i} = \max\{X - Su^i d^{j-i}, e^{-r\Delta t}[pP_{j+1,i+1} + (1-p)P_{j+1,i}]\}$$



The value of the American put option is therefore approximately 4.92.

Number of Nodes in a Binomial Tree
The number of nodes in the binomial tree is $\frac{n(n+1)}{2}$ when we count the number of time steps from 1. When we count the number of time steps from 0, as we have done in this chapter, the number of nodes in the binomial tree is $\frac{(n+1)(n+2)}{2}$.

Local Volatility
The local volatility in a standard CRR binomial tree is naturally constant for each time step and is given by

$$\sigma_{j,i} = \frac{1}{\sqrt{\Delta t}}\sqrt{p(1-p)}\ln(u^2)$$

The local volatility is generally different from the input volatility. It converges to the global input volatility as the number of time steps becomes large.

Negative Probabilities in the CRR Tree
A low volatility and relatively high cost-of-carry can induce negative risk-neutral probabilities in the CRR tree; see Chriss (1997). More

precisely, we will get negative risk-neutral probabilities when[1]

$$\sigma < |b\sqrt{\Delta t}|$$

Negative probabilities are not necessarily a bad thing, despite being inconsistent with the basic axioms of probability. Allowing negative probabilities in a model will in general increase its flexibility (Haug, 2004). The problem with the CRR model is not negative probabilities per se, but rather that the state space is not large enough to cover all relevant events when $\sigma < |b\sqrt{\Delta t}|$.

7.1.2 Greeks in CRR Binomial Tree

We now look at how to calculate the most basic Greeks in a binomial tree.

The delta, the change in option price for a one unit change in asset price, is given by

$$\Delta = \frac{f_{1,1} - f_{1,0}}{Su - Sd}, \tag{7.9}$$

where $f_{j,i}$ is the price of the derivative security in the binomial tree at node (j, i). The gamma, the change in delta for a one unit change in asset price, is given by

$$\Gamma = \frac{\frac{f_{2,2}-f_{2,1}}{Su^2-S} - \frac{f_{2,1}-f_{2,0}}{S-Sd^2}}{\frac{1}{2}(Su^2 - Sd^2)} \tag{7.10}$$

The theta, the change in option value for a one-day closer to maturity, *ceteris paribus*, is given by

$$\Theta = \frac{f_{2,1} - f_{0,0}}{2\Delta t 365}, \tag{7.11}$$

where $f_{2,1}$ is the derivative's price in the tree at time step $j = 2$ and number of up-steps $i = 1$. Similarly, $f_{0,0}$ is the derivatives price at time step $j = 0$ and number of up-steps $i = 0$ (very first node). The current asset price is S, and u and d is the up and down jump sizes.

The delta, gamma, and theta calculations just presented are incorporated in the computer code below. An efficient incorporation of all of these sensitivities can be achieved by running through the binomial tree only once. Calculation of vega and rho, on the other

[1]See also Hull (2005).

hand, is less efficient:

$$\text{Vega} = \frac{f(\sigma + \Delta\sigma) - f(\sigma - \Delta\sigma)}{2\Delta\sigma}$$

We thus need to calculate the option value twice in the binomial tree to get an estimate of the sensitivity to changes in implied volatility (vega). The same is true for rho.

Computer algorithm

The computer code returns the value of European and American call or put options. Setting the *OutputFlag* = "*p*" returns the option price, *OutputFlag* = "*d*" returns the option's delta, *OutputFlag* = "*g*" yields the option's gamma, *OutputFlag* = "*t*" yields the theta, *OutputFlag* = "*a*" returns an array containing the option value, delta, gamma, and theta. Setting the *AmeEurFlag* = "*a*" gives American option values: *AmeEurFlag* = "*e*" gives European values. Similarly, setting the *CallPutFlag* = "*c*" returns a call value, and *CallPutFlag* = "*p*" returns a put value. In the computer code, $v = \sigma$ and $dt = \Delta t$.

```
Function CRRBinomial(OutputFlag As String, AmeEurFlag As String,_
 CallPutFlag As String, S As Double, X As Double, T As Double, _
 r As Double, b As Double, v As Double, n As Integer) As Variant

    Dim OptionValue() As Double
    Dim u As Double, d As Double, p As Double
    Dim ReturnValue(4) As Double
    Dim dt As Double, Df As Double
    Dim i As Integer, j As Integer, z As Integer

    ReDim OptionValue(0 To n + 1)

    If CallPutFlag = "c" Then
        z = 1
        ElseIf CallPutFlag = "p" Then
        z = -1
    End If

    dt = T / n
    u = Exp(v * Sqr(dt))
    d = 1 / u
    p = (Exp(b * dt) - d) / (u - d)
    Df = Exp(-r * dt)

    For i = 0 To n
         OptionValue(i) = Max(0, z * (S * u^i * d^(n - i) - X))
    Next

    For j = n - 1 To 0 Step -1
        For i = 0 To j
            If AmeEurFlag = "e" Then
                OptionValue(i) = (p * OptionValue(i + 1) _
        + (1 - p) * OptionValue(i)) * Df
            ElseIf AmeEurFlag = "a" Then
```

```
                OptionValue(i) = Max((z * (S * u^i * d^(j - i) - X)), _
                (p * OptionValue(i + 1) + (1 - p) * OptionValue(i)) * Df)
            End If

        Next
        If j = 2 Then
                ReturnValue(2) = ((OptionValue(2) - OptionValue(1)) _
                / (S * u^2 - S) - (OptionValue(1) - OptionValue(0)) _
                / (S - S * d^2)) / (0.5 * (S * u^2 - S * d^2))
                ReturnValue(3) = OptionValue(1)
            End If
        If j = 1 Then
                ReturnValue(1) = (OptionValue(1) - OptionValue(0)) _
                / (S * u - S * d)
            End If
    Next
    ReturnValue(3) = (ReturnValue(3) - OptionValue(0)) / (2 * dt) / 365
    ReturnValue(0) = OptionValue(0)
    If OutputFlag = "p" Then 'Option value
        CRRBinomial = ReturnValue(0)
    ElseIf OutputFlag = "d" Then 'Delta
        CRRBinomial = ReturnValue(1)
    ElseIf OutputFlag = "g" Then 'Gamma
        CRRBinomial = ReturnValue(2)
    ElseIf OutputFlag = "t" Then 'Theta
        CRRBinomial = ReturnValue(3)
   ElseIf OutputFlag = "a" Then
        CRRBinomial = Application.Transpose(ReturnValue())
    End If

End Function
```

7.1.3 Rendleman Bartter Binomial Tree

Rendleman and Bartter (1979) suggested setting the up probability in the tree equal to $p = 0.5$. This gives us

$$u = e^{(b-\sigma^2/2)\Delta t+\sigma\sqrt{\Delta t}} \qquad d = e^{(b-\sigma^2/2)\Delta t-\sigma\sqrt{\Delta t}}$$

This set of parameters is discussed in more detail in Jarrow and Rudd (1982).

The local volatility in this binomial tree is naturally constant for each time step and is given by

$$\sigma_{j,i} = \frac{1}{\Delta t}\sqrt{p(1-p)}\ln(u/d)$$

Greeks in the Rendleman Bartter Tree

The delta is given by

$$\Delta = \frac{f_{1,1} - f_{1,0}}{Su - Sd},$$

which is the same expression as in the CRR binomial tree. The calculation of gamma is slightly different though:

$$\Gamma = \frac{\frac{f_{2,2}-f_{2,1}}{Su^2-Sud} - \frac{f_{2,1}-f_{2,0}}{Sud-Sd^2}}{\frac{1}{2}(Su^2 - Sd^2)},$$

where $f_{j,i}$ is the derivatives price in the tree at time step j and number of up-steps i. The current asset price is S, and u and d are the up and down jump factors.

7.1.4 Leisen-Reimer Binomial Tree

Leisen and Reimer (1996) set the u and d factors in such a way that the tree centers around the strike price. This makes the tree converges "smoothly." This makes their tree more efficient for valuation of standard options relative to the CRR and Rendleman-Bartter trees. The probability of going up is set to

$$p = h(d_2),$$

where the probability of going down is naturally given by $1 - p$. Moreover, the up and down factors are set to

$$u = e^{b\Delta t}\frac{h(d_1)}{h(d_2)}$$

$$d = \frac{e^{b\Delta t} - pu}{1 - p},$$

where, as usual, $\Delta t = T/n$ and

$$d_1 = \frac{\ln(S/X) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

There are two alternatives to calculate $h(x)$. The first is the Preizer-Pratt inversion method 1:

$$h(x) = 0.5 + \eta\left\{0.25 - 0.25\exp\left[-\left(\frac{x}{n+\frac{1}{3}}\right)^2\left(n+\frac{1}{6}\right)\right]\right\}^{1/2}$$

The second is the Preizer-Pratt inversion method 2:

$$h(x) = 0.5 + \eta\left\{0.25 - 0.25\exp\left[-\left(\frac{x}{n+\frac{1}{3}+\frac{0.1}{n+1}}\right)^2\left(n+\frac{1}{6}\right)\right]\right\}^{1/2},$$

where $\eta = 1$ if $x >= 0$ and -1 if $x < 0$. Further, the number of time steps n should be odd to ensure the exercise price falls on a set of nodes.

Delta and gamma can be calculated in the same way as for the Rendleman-Bartter binomial tree.

Computer algorithm

The computer code returns the value of European and American call or put options using the Leisen-Reimer binomial algorithm with Preizer-Pratt inversion method 2. Setting the *OutputFlag* = "*p*" returns the option price. *OutputFlag* = "*d*" returns the option's delta. *OutputFlag* = "*g*" yields the option's gamma. *OutputFlag* = "*a*" returns an array containing the option value, delta, and gamma. Setting the *AmeEurFlag* = "*a*" gives American option values; *AmeEurFlag* = "*e*" gives European values. Similarly, setting the *CallPutFlag* = "*c*" returns a call value, and *CallPutFlag* = "*p*" returns a put value. In the computer code, $v = \sigma$ and $dt = \Delta t$.

```
Option Base 0

Function LeisenReimerBinomial(OutputFlag As String, _
  AmeEurFlag As String, CallPutFlag As String, S As Double, _
  X As Double, T As Double, r As Double, b As Double, _
  v As Double, n As Integer) As Variant


    Dim OptionValue() As Double
    Dim ReturnValue(3) As Double
    Dim d1 As Double, d2 As Double
    Dim hd1 As Double, hd2 As Double
    Dim u As Double, d As Double, p As Double
    Dim dt As Double, Df As Double
    Dim i As Integer, j As Integer, z As Integer

     n = Application.Odd(n)

    ReDim OptionValue(0 To n)

     If CallPutFlag = "c" Then
         z = 1
         ElseIf CallPutFlag = "p" Then
         z = -1
    End If

    d1 = (Log(S / X) + (b + v^2 / 2) * T) / (v * Sqr(T))
    d2 = d1 - v * Sqr(T)
    hd1 = 0.5 + Sgn(d1) * (0.25 - 0.25 _
    * Exp(-(d1 / (n + 1 / 3 + 0.1 / (n + 1)))^2 * (n + 1 / 6)))^0.5
    hd2 = 0.5 + Sgn(d2) * (0.25 - 0.25 _
    * Exp(-(d2 / (n + 1 / 3 + 0.1 / (n + 1)))^2 * (n + 1 / 6)))^0.5

    dt = T / n
    p = hd2
    u = Exp(b * dt) * hd1 / hd2
    d = (Exp(b * dt) - p * u) / (1 - p)
    Df = Exp(-r * dt)
    For i = 0 To n
         OptionValue(i) = Max(0, z * (S * u^i * d^(n - i) - X))
    Next

    For j = n - 1 To 0 Step -1
        For i = 0 To j
```

```
            If AmeEurFlag = "e" Then
                OptionValue(i) = (p * OptionValue(i + 1) + (1 - p) _
        * OptionValue(i)) * Df
            ElseIf AmeEurFlag = "a" Then
                OptionValue(i) = Max((z * (S * u^i * d^(j - i) - X)), _
                (p * OptionValue(i + 1) + (1 - p) * OptionValue(i)) * Df)
            End If

        Next
        If j = 2 Then
                ReturnValue(2) = ((OptionValue(2) - OptionValue(1)) _
        / (S * u^2 - S * u * d) _
                - (OptionValue(1) - OptionValue(0)) _
        / (S * u * d - S * d^2)) / (0.5 * (S * u^2 - S * d^2))
                ReturnValue(3) = OptionValue(1)
            End If
         If j = 1 Then
                ReturnValue(1) = (OptionValue(1) - OptionValue(0)) _
        / (S * u - S * d)
            End If
    Next
    ReturnValue(0) = OptionValue(0)
    If OutputFlag = "p" Then 'Option value
        LeisenReimerBinomial = ReturnValue(0)
    ElseIf OutputFlag = "d" Then 'Delta
        LeisenReimerBinomial = ReturnValue(1)
    ElseIf OutputFlag = "g" Then 'Gamma
        LeisenReimerBinomial = ReturnValue(2)
   ElseIf OutputFlag = "a" Then
        LeisenReimerBinomial = Application.Transpose(ReturnValue())
    End If

End Function
```

7.1.5 Convertible Bonds in Binomial Trees

A convertible bond can be seen as a combination of a plain bond and a stock option. If the stock price is far below the strike (conversion price), the convertible behaves like a straight bond. If the stock price is far above the strike, the convertible behaves like a stock. This should also affect the discounting of the cash flows. When the convertible is deep-out-of-the-money, the future cash flows should be discounted by a rate that takes into account the credit spread k above the treasury rate of the particular bond. If the convertible is deep-in-the-money, it is almost certain to be converted, and the cash flows should be discounted at the risk-free rate.

Bardhan, Bergier, Derman, Dosembet, and Kani (1994) have incorporated these effects by using a discounting rate that is a function of a variable conversion probability. The model starts out with a standard binomial stock price tree. The convertible bond price is then found by starting at the end of the stock price tree. At each end node, the convertible value must be equal to the maximum of the value of converting the bond into stocks or the face value plus the final coupon.

One next rolls backward through the tree, using backward induction. If it is optimal to convert the bond, the value is set equal to the conversion value at that node, or else the convertible bond value $P_{n,i}$ is set equal to

$$P_{n,i} = \max[mS,\ pP_{n+1,i+1}e^{-r_{n+1,i+1}\Delta t} + (1-p)P_{n+1,i}e^{-r_{n+1,i}\Delta t}], \tag{7.12}$$

where m is the conversion ratio. Some convertible bonds have an initial lockout period during which the investor is not allowed to convert the bond. The convertible bond value at these nodes can be simplified to

$$P_{n,i} = pP_{n+1,i+1}e^{-r_{n+1,i+1}\Delta t} + (1-p)P_{n+1,i}e^{-r_{n+1,i}\Delta t}$$

Instead of using a constant discount rate r, the discount rate $r_{n,i}$ is set to fluctuate with the conversion probability $q_{n,i}$ at each node.

The conversion probabilities $q_{n,i}$, where n is the time step and i the number of up moves (the state), are calculated by starting at the end of the stock price tree. If it is optimal to convert the bond, the conversion probability is set to 1; otherwise, the conversion probability is set to 0. For time steps before the end of the tree, the conversion probability is set equal to 1 if it is optimal to convert at that node; otherwise,

$$q_{n,i} = pq_{n+1,i+1} + (1-p)q_{n+1,i} \tag{7.13}$$

The credit-adjusted discount rate is set equal to a conversion probability weighted mixture of the risk-free rate and the credit-adjusted rate. This gives a discount rate for up moves equal to

$$r_{n,i} = q_{n,i}r + (1-q_{n,i})(r+k) \tag{7.14}$$

The discount rate is thus set equal to the constant risk-free rate r when the conversion probability is 1, and set equal to the risk-free rate plus the credit spread $r+k$ when the conversion probability is 0. The discount rate moves smoothly between the risk-free rate and the credit-adjusted rate for conversion probabilities between 0 and 1.

Example

Consider a convertible corporate bond with five years to maturity. The continuously compounding yield on a five-year treasury bond is 7%, the credit spread on the corporate bond is 3% above treasury, the face value is 100, the annual coupon is 6, the conversion ratio is 1, the current stock price is 75, and the volatility of the stock is 20%. What is the value of the convertible bond? $S = 75$, $T = 5$, $r = b = 0.07$, $k = 0.03$, $m = 1$, and $\sigma = 0.2$.

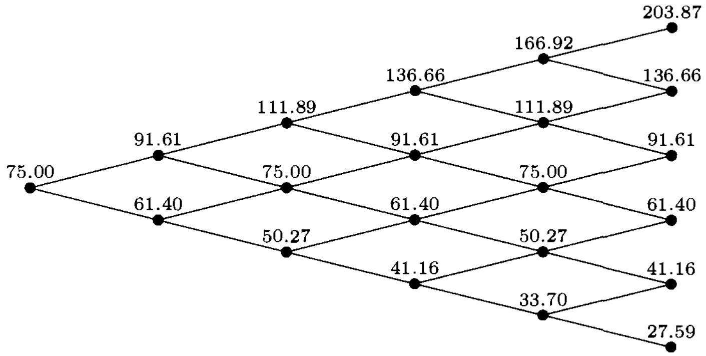
To price the convertible bond, we need to build a standard binomial stock price tree. With the number of time steps $n = 5$, we get $\Delta t = 1$ and up and down factors

$$u = e^{\sigma\sqrt{\Delta t}} = e^{0.2\sqrt{1}} = 1.2214, \qquad d = \frac{1}{u} = 0.8187$$

The probability of an increase in price is thus given by

$$p = \frac{e^{b\Delta t} - d}{u - d} = \frac{e^{0.07\times 1} - 0.8187}{1.2214 - 0.8187} = 0.6302,$$

and we obtain the following binomial stock price tree:



The next step is to find the convertible bond values and the conversion probabilities at each node in the tree. To see how this works, let's look at the calculation of several nodes.

At the end node with stock price 203.87, it is better to convert the bond into one stock and receive the stock price 203.87 than to get the notional plus the coupon $100 + 6$. The probability of conversion at this node, $q_{5,5}$, is 100%, which we write as 1.00 in the conversion probability tree.

At the end node, with a stock price of 91.61, it is better not to convert the bond and receive the face value plus the coupon of 106. The probability of conversion is $q_{5,3} = 0$.

For the node at year four ($n = 4$) with stock price 111.89, the convertible bond value of 121.77 is found by using equation (7.12):

$$P_{4,4} = \max\left[1 \times 111.89,\ 0.6302 \times 136.66e^{-r_{n+1,i+1}\times 1} + (1 - 0.6302)106.00e^{-r_{n+1,i}\times 1}\right]$$

The credit-adjusted discount rates are found by using equation (7.14):

$$r_{n+1,i+1} = 1 \times 0.07 + (1 - 1)(0.07 + 0.03) = 0.07$$

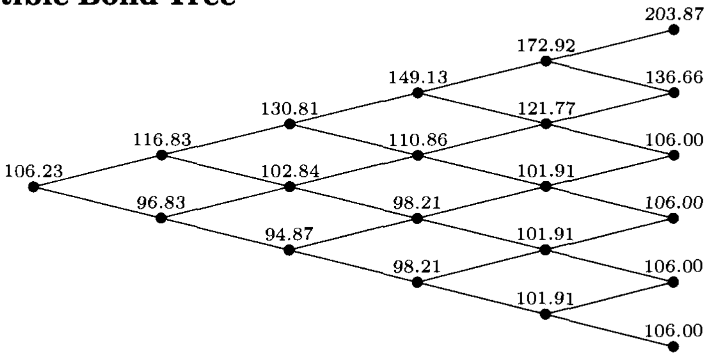
$$r_{n+1,i} = 0 \times 0.07 + (1 - 0)(0.07 + 0.03) = 0.1$$

The conversion probability of 0.63 at this node is given by equation (7.13):

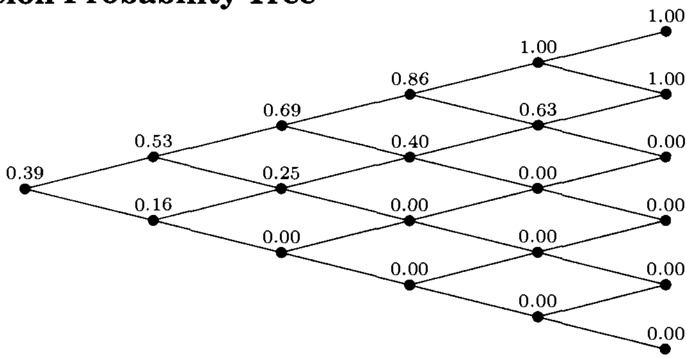
$$q_{4,4} = 0.6302 \times 1 + (1 - 0.6302) \times 0$$

The same procedure can be used to find any convertible bond value and conversion probability.

Convertible Bond Tree



Conversion Probability Tree



In the above section the main principles of how to incorporate a convertible bond model were outlined. In practice, there are many additional issues to take into account. Some convertible bonds allow the issuer to force investors to convert the bond if the stock price reaches a certain prespecified level (barrier). To include a barrier in the convertible binomial model, the number of time steps should be chosen to make the barrier fall exactly *on* the nodes. The conversion probability is then set to 1 if the stock price is larger than or equal to the barrier. The issuer of the convertible bond also often has the right to call the bond, and the investor has the right to put the bond. The paper of Bardhan, Bergier, Derman, Dosembet, and Kani (1994) is a good start to look into such practicalities.

Computer algorithm

```
Function ConvertibleBond(AmeEurFlag As String, S As Double, X As Double, _
        t2 As Double, t1 As Double, r As Double, k As Double, _
        q As Double, v As Double, F As Double, _
        Coupon As Double, n As Integer) As Double

    'T2:              Time to maturity
    't1:              Time to to first conversion date, only used when the option is Americ
    'k                Issuer's credit spread above risk-free return (similar Treasuries).
    'q                Stock dividend yield
    'v:               Stock volatility
    'n:               Number of time-steps in the tree
    'F                Bond face value

    Dim OptionValue() As Double            'Option Value at each node
    Dim ConversionProbability() As Double 'Conversion probability at each node
    Dim dt As Double
    Dim u As Double, d As Double, p As Double
    Dim ru As Double, rd As Double
    Dim CouponValue As Double
    Dim CouponTime As Double     'Time to last coupon payment from today
    Dim Time As Double
    Dim StockPrice As Double
    Dim CRatio As Double     'Conversion Ratio= FaceValue/Strike
    Dim i As Integer, j As Integer, CouponPayment As Integer

    CRatio = F / X

    ReDim OptionValue(n + 1)
    ReDim ConversionProbability(n + 1)

    dt = t2 / n
    u = Exp(v * Sqr(dt))
    d = 1 / u
    p = (Exp((r - q) * dt) - d) / (u - d)

    For i = 0 To n
        StockPrice = S * u^i * d^(n - i)
        OptionValue(i) = Max(F + Coupon, CRatio * StockPrice)
        If OptionValue(i) = F + Coupon Then
            ConversionProbability(i) = 0
        Else
            ConversionProbability(i) = 1
        End If
    Next

    CouponTime = t2
    For j = n - 1 To 0 Step -1:

     If CouponTime <> t2 - Int((n - j) * dt) Then
        CouponPayment = 1
        CouponTime = t2 - Int((n - j) * dt)
        Time = CouponTime - dt * j
    Else
        CouponPayment = 0
     End If

        For i = 0 To j
```

```
            ru = ConversionProbability(i + 1) * r _
              + (1 - ConversionProbability(i + 1)) * (r + k)
            rd = ConversionProbability(i) * r _
              + (1 - ConversionProbability(i)) * (r + k)
            StockPrice = S * u^i * d ^ Abs(i - j)
            If CouponPayment = 1 Then
                CouponValue = Coupon * (p * Exp(-ru * Time) _
                + (1 - p) * Exp(-rd * Time))
            Else
                CouponValue = 0
            End If

            If AmeEurFlag = "e" Or AmeEurFlag = "a" And dt * j <= t1 Then
                OptionValue(i) = CouponValue + p * OptionValue(i + 1)_
         * Exp(-ru * dt) + (1 - p) * OptionValue(i) * Exp(-rd * dt)
                ConversionProbability(i) = p * ConversionProbability(i + 1) _
        + (1 - p) * ConversionProbability(i)
            ElseIf AmeEurFlag = "a" Then
                OptionValue(i) = Max(CRatio * (StockPrice - X) + F, _
                CouponValue + p * OptionValue(i + 1) * Exp(-ru * dt) _
        + (1 - p) * OptionValue(i) * Exp(-rd * dt))
                If OptionValue(i) = CRatio * (StockPrice - X) + F Then
                    ConversionProbability(i) = 1
                Else
                    ConversionProbability(i) = p * ConversionProbability(i + 1) _
        + (1 - p) * ConversionProbability(i)
                End If
            End If
        Next
    Next
    ConvertibleBond = OptionValue(0)

End Function
```

7.2 BINOMIAL MODEL WITH SKEWNESS AND KURTOSIS

Rubinstein (1998) use an Edgeworth expansion to take into account skewness and kurtosis in a standard discrete binomial probability distribution. This generalized binomial model is naturally more flexible than a closed-form solution (Chapter 6) and can be used to value a variety of options.

The method takes the Rendelman-Bartter node probability as a primitive:

$$b(x) = \frac{n!}{i!(n-i)!}\left(\frac{1}{2}\right)^n$$

To get a skewness-kurtosis tree, multiply the path probability with either the Edgeworth expansion

$$1 + \frac{1}{6}\mu_3(x^3 - 3x) + \frac{1}{24}(\mu_4 - 3)(x^4 - 6x^2 + 3) + \frac{1}{72}\mu_3^2(x^6 - 15x^4 + 45x^2 - 15)$$

or the Gram-Charlier expansion

$$1 + \frac{1}{6}\mu_3(x^3 - 3x) + \frac{1}{24}(\mu_4 - 3)(x^4 - 6x^2 + 3),$$

where μ_3 is the skewness and μ_4 is the Pearson kurtosis. Let's call this new skewness and kurtosis adjusted path probability for $f(x_i)$. Because this approach is only an approximation, the sum of path probabilities will not necessarily add up to 1: $\sum_i^n f(x_i) \neq 1$ and we need to rescale so we get the sum of probabilities equal to unity. This is done by simply replacing $f(x_i)$ with $\frac{f(x_i)}{\sum_i^n f(x_i)}$. See also Jackson and Staunton (2001) for more information on implementing such binomial trees.

Computer algorithm

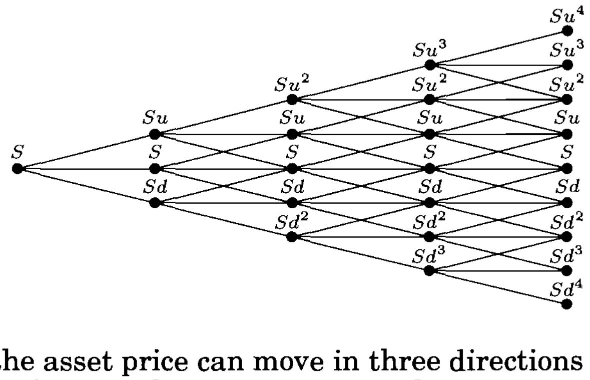
The computer code returns the value of a European call or put option from a Rubinstein skewness and kurtosis adjusted binomial tree (RubinsteinSKB). Setting the *Expansion* = "*e*" gives the value using Edgeworth expansion: *Expansion* = "*c*" gives the values using Gram-Charlier expansion. Similarly, setting the *CallPutFlag* = "*c*" returns a call value, and *CallPutFlag* = "*p*" returns a put value. Setting the *OutputFlag* = "*p*" returns the option value, *OutputFlag* = "*prob*" returns the probability distribution, and *OutputFlag* = "*St*" returns the node values(geometry of the tree). In the computer code $v = \sigma$ and $dt = \Delta t$.

```
Function RubinsteinSKB(Expansion As String, OutputFLag As String, _
  CallPutFlag As String, S As Double, X As Double, T As Double, _
  r As Double, b As Double, v As Double, _
  Skew As Double, Kurt As Double, n As Double) As Variant

        Dim u As Double, d As Double, p() As Double
        Dim Sum As Double, PSum As Double, dt As Double, A As Double
        Dim i As Integer, z As Integer
        Dim xi As Double

       Dim St() As Double

        z = 1
        If CallPutFlag = "p" Then
            z = -1
        End If

        ReDim p(0 To n)
       ReDim St(0 To n)

        dt = (T / n)
        u = Exp((b - v^2 / 2) * dt + v * Sqr(dt))
        d = Exp((b - v^2 / 2) * dt - v * Sqr(dt))

    Sum = 0
    PSum = 0
        For i = 0 To n Step 1
            xi = (2 * i - n) / Sqr(n)
            If Expansion = "e" Then 'Edgworth-Expansion
```

```
            p(i) = Application.Combin(n, i) * 0.5^n _
               * (1 + 1 / 6 * Skew * (xi^3 - 3 * xi) _
    + 1 / 24 * (Kurt - 3) * (xi^4 - 6 * xi^2 + 3) _
    + Skew^2 * (xi^6 - 15 * xi ^ 4 + 45 * xi^2 - 15) / 72)
        Else 'Gram -Charlier
            p(i) = Application.Combin(n, i) * 0.5^n _
               * (1 + 1 / 6 * Skew * (xi^3 - 3 * xi) _
    + 1 / 24 * (Kurt - 3) * (xi^4 - 6 * xi^2 + 3))
        End If
        PSum = PSum + p(i)
        St(i) = S * u^i * d^(n - i)
    Next
    For i = 0 To n Step 1
        p(i) = p(i) / PSum
        Sum = Sum + p(i) * Max(z * (S * u^i * d^(n - i) - X), 0)
    Next
If OutputFLag = "p" Then
    RubinsteinSKB = Exp(-r * T) * Sum
ElseIf OutputFLag = "prob" Then
    RubinsteinSKB = Application.Transpose(p())
 ElseIf OutputFLag = "St" Then
    RubinsteinSKB = Application.Transpose(St())
 End If

End Function
```

7.3 TRINOMIAL TREES

Trinomial trees were introduced in option pricing by Boyle (1986) and are similar to binomial trees. Trinomial trees can be used to price both European and American options on a single underlying asset.[2]



Because the asset price can move in three directions from a given node, compared with only two in a binomial tree, the number of time steps can be reduced and still attain the same accuracy as in the binomial tree. The main advantage is that trinomial trees offer more

[2]One-dimensional trinomial trees can also be used to price some derivatives on two correlated assets. However, a more efficient method, the three-dimensional lattice model, will be discussed later.

flexibility than binomial trees. The extra flexibility is useful when pricing complex derivatives, such as American barrier options.[3]

There are several ways to choose jump size and corresponding probabilities in a trinomial tree that will all give the same result when the number of time steps is large. To discretize a geometric Brownian motion, the jump sizes and probabilities must match the first two moments of the distribution (the mean and variance). One possibility is to build a trinomial tree where the asset price at each node can go up, stay at the same level, or go down. In that case, the up and down jump sizes are

$$u = e^{\sigma\sqrt{2\Delta t}} \qquad d = e^{-\sigma\sqrt{2\Delta t}},$$

and the probability of going up and down respectively are

$$p_u = \left(\frac{e^{b\Delta t/2} - e^{-\sigma\sqrt{\Delta t/2}}}{e^{\sigma\sqrt{\Delta t/2}} - e^{-\sigma\sqrt{\Delta t/2}}}\right)^2$$

$$p_d = \left(\frac{e^{\sigma\sqrt{\Delta t/2}} - e^{b\Delta t/2}}{e^{\sigma\sqrt{\Delta t/2}} - e^{-\sigma\sqrt{\Delta t/2}}}\right)^2$$

The probabilities must sum to unity. Thus, the probability of the asset price remaining unchanged is

$$p_m = 1 - p_u - p_d$$

T is the time to maturity in years, b is the cost-of-carry, $\Delta t = \frac{T}{n}$ is the length of each time step, and n is the number of time steps. After building the asset price tree, the value of the option can be found in the standard way by using backward induction. When the volatility σ is very low and the cost-of-carry is very high, p_u and p_d can sum to more than unity, and then naturally p_m will become negative. More precisely, we will get a negative probability $p_m < 0$ when

$$\sigma < \sqrt{\frac{b^2\Delta t}{2}}$$

Using this inequality, we can also find that we need to set the number of time steps to $n \geq \text{Integer}\left[\frac{b^2 T}{2\sigma^2}\right] + 1$ to avoid negative probabilities. We can alternatively avoid negative probabilities by choosing a more optimal state space. See Haug (2004) for a more detailed discussion on negative probabilities in tree models.

[3]See Ritchken (1995) and Cheuk and Vorst (1996).

Computer algorithm

The computer code returns the value of European and American call or put options. Setting the *OutputFlag* = "*p*" returns the option price, *OutputFlag* = "*d*" returns the option's delta, *OutputFlag* = "*g*" yields the option's gamma, *OutputFlag* = "*t*" yields the theta, *OutputFlag* = "*a*" returns an array containing the option value, delta, gamma, and theta. Setting the *AmeEurFlag* = "*a*" gives American option values, and *AmeEurFlag* = "*e*" gives European values. Similarly, setting the *CallPutFlag* = "*c*" returns a call value, and *CallPutFlag* = "*p*" returns a put value. In the computer code, $v = \sigma$ and $dt = \Delta t$.

```
Function TrinomialTree(OutputFlag As String, AmeEurFlag As String, _
  CallPutFlag As String, S As Double, X As Double, T As Double, r As Double, _
  b As Double, v As Double, n As Integer) As Variant


    Dim OptionValue() As Double
    ReDim OptionValue(0 To n * 2 + 1)

    Dim ReturnValue(3) As Double
    Dim dt As Double, u As Double, d As Double
    Dim pu As Double, pd As Double, pm As Double, Df As Double
    Dim i As Long, j As Long, z As Integer

     If CallPutFlag = "c" Then
        z = 1
     ElseIf CallPutFlag = "p" Then
        z = -1
    End If

    dt = T / n
    u = Exp(v * Sqr(2 * dt))
    d = Exp(-v * Sqr(2 * dt))
    pu = ((Exp(b * dt / 2) - Exp(-v * Sqr(dt / 2))) _
        / (Exp(v * Sqr(dt / 2)) - Exp(-v * Sqr(dt / 2))))^2

    pd = ((Exp(v * Sqr(dt / 2)) - Exp(b * dt / 2)) _
        / (Exp(v * Sqr(dt / 2)) - Exp(-v * Sqr(dt / 2))))^2
    pm = 1 - pu - pd
    Df = Exp(-r * dt)

    For i = 0 To (2 * n)
         OptionValue(i) = Max(0, z * (S * u^Max(i - n, 0) _
         * d^Max(n - i, 0) - X))
    Next

    For j = n - 1 To 0 Step -1

        For i = 0 To (j * 2)

            OptionValue(i) = (pu * OptionValue(i + 2) _
                + pm * OptionValue(i + 1) + pd * OptionValue(i)) * Df

            If AmeEurFlag = "a" Then
                OptionValue(i) = Max(z * (S * u^Max(i - j, 0) _
                    * d^Max(j - i, 0) - X), OptionValue(i))
```

```
            End If
        Next
    If j = 1 Then
            ReturnValue(1) = (OptionValue(2) - OptionValue(0)) _
            / (S * u - S * d)
            ReturnValue(2) = ((OptionValue(2) - OptionValue(1)) / (S * u - S) _
              - (OptionValue(1) - OptionValue(0)) / (S - S * d)) _
               / (0.5 * (S * u - S * d))
            ReturnValue(3) = OptionValue(1)
     End If
   Next

   ReturnValue(3) = (ReturnValue(3) - OptionValue(0)) / dt / 365

   ReturnValue(0) = OptionValue(0)
   If OutputFlag = "p" Then 'Option value
       TrinomialTree = ReturnValue(0)
   ElseIf OutputFlag = "d" Then 'Delta
       TrinomialTree = ReturnValue(1)
   ElseIf OutputFlag = "g" Then 'Gamma
       TrinomialTree = ReturnValue(2)
   ElseIf OutputFlag = "t" Then 'Theta
       TrinomialTree = ReturnValue(3)
  ElseIf OutputFlag = "a" Then ' All
       TrinomialTree = Application.Transpose(ReturnValue())
   End If

End Function
```

Example

To price an American put option with stock price 100, strike price 110, time to maturity of six months, risk-free rate equal to the cost-of-carry of 10%, volatility 27%, and 30 time steps: *TrinomialTree*("*p*", "*a*", "*p*", 100, 110, 0.5, 0.1, 0.1, 0.27, 30) returns an American put value of 11.6493.

Alternative Trinomial Tree

The above tree was a CRR-equivalent trinomial tree. As already indicated, there is an unlimited number of sample spaces and corresponding probability parameters that can be used to consistently price an option. Following is another popular set of parameters often used in the literature:

$$p_u = \frac{1}{6} + (b - \sigma^2/2)\sqrt{\frac{\Delta t}{12\sigma^2}}$$

$$p_d = \frac{1}{6} - (b - \sigma^2/2)\sqrt{\frac{\Delta t}{12\sigma^2}}$$

$$p_m = \frac{2}{3}$$

This implementation yields a negative up probability p_u if

$$\sigma > \sqrt{2b + \frac{2}{3\Delta t} + \frac{2\sqrt{1 + 6b\Delta t}}{3\Delta t}}$$

and a negative down probability p_d if

$$\sigma < \sqrt{2b + \frac{2}{3\Delta t} - \frac{2\sqrt{1 + 6b\Delta t}}{3\Delta t}}.$$

The up probability p_u becomes larger than unity if

$$\sigma < \sqrt{2b + \frac{50}{3\Delta t} - \frac{10\sqrt{25 + 6b\Delta t}}{3\Delta t}},$$

and similarly for the down probability when

$$\sigma > \sqrt{2b + \frac{50}{3\Delta t} + \frac{10\sqrt{25 + 6b\Delta t}}{3\Delta t}}$$

For example, cost-of-carry 20%, 20 time steps, and one year to maturity results in a negative down probability if the volatility is below 7.63%. This is a realistic scenario, although not a frequently occurring one. To avoid negative proabilities with this set of parameters, we can set the number of time steps n equal to or higher than

$$\text{Integer}\left[3bT\left(-1 + \frac{b}{\sigma^2} + \frac{\sigma^2}{4b}\right)\right] + 1$$

7.4 EXOTIC OPTIONS IN TREE MODELS

7.4.1 Options on Options

In Chapter 4 we looked at analytical formulas for options on European options. The underlying option can naturally be American style, and the option on the option can be either European or American. We can, for example, have an American option on a European option. We thus have four combinations in total:

- European on European
- European on American
- American on European
- American on American

American on American is naturally the most valuable, *ceteris paribus*. Any of these combinations can easily be valued in a binomial or trinomial tree. Basically all we need to do is to add a few lines of code.

Computer algorithm

The algorithm can be used to calculate the value and greeks for options on options. *OutputFlag* = "*p*" returns the option value, *OutputFlag* = "*d*" gives the delta, *OutputFlag* = "*g*" gives the gamma, *OutputFlag* = "*t*" gives the theta, and *OutputFlag* = "*a*" returns a column array with the price, delta, gamma, and theta. *CompoundEurAmeFlag* = "*e*" gives a European-style compound option; *CompoundEurAmeFlag* = "*a*" gives an American-style compound option. *AmeEurFlag* = "*e*" sets the underlying option to European, and *AmeEurFlag* = "*a*" sets the underlying option to American. *CompoundTypeFlag* = "*cc*" gives a call on a call option, *CompoundTypeFlag* = "*cp*" gives a call on a put option, *CompoundTypeFlag* = "*pc*" gives a put on a call option, while *CompoundTypeFlag* = "*pp*" gives a put on a put option.

```
Function BinomialCompoundOption(OutputFlag As String, _
  CompoundEurAmeFlag As String, AmeEurFlag As String, _
  CompoundTypeFlag As String, S As Double, X1 As Double, _
  X2 As Double, t1 As Double, T2 As Double,  r As Double, _
  b As Double, v As Double, n As Integer) As Variant


    Dim OptionValue() As Double, ReturnValue(3) As Double
    Dim u As Double, d As Double, p As Double
    Dim dt As Double, Df As Double
    Dim i As Integer, j As Integer, z As Integer, y As Integer, w As Integer

    ReDim OptionValue(0 To n + 1)

    If CompoundTypeFlag = "cc" Or CompoundTypeFlag = "pc" Then
        z = 1
    Else
        z = -1
    End If

    If CompoundTypeFlag = "cc" Or CompoundTypeFlag = "cp" Then
        y = 1
    Else
        y = -1
    End If

    w = 1
    dt = T2 / n
    u = Exp(v * Sqr(dt))
    d = 1 / u
    p = (Exp(b * dt) - d) / (u - d)
    Df = Exp(-r * dt)

    For i = 0 To n
         OptionValue(i) = Max(0, z * (S * u^i * d^(n - i) - X1))
    Next

    For j = n - 1 To 0 Step -1
        For i = 0 To j

            OptionValue(i) = (p * OptionValue(i + 1) + (1 - p) _
```

```
                * OptionValue(i)) * Df

            If AmeEurFlag = "a" Then
                OptionValue(i) = Max((z * (S * u^i * d^(j - i) _
                    - X1)), OptionValue(i))
            End If

            If t1 >= dt * j And w = 1 Then
                    OptionValue(i) = Max(y * (OptionValue(i) - X2), 0)
                    If i = j Then
                        w = -1
                    End If
            End If
            If w = -1 And CompoundEurAmeFlag = "a" Then
                OptionValue(i) = Max(y * (OptionValue(i) _
                  - X2), OptionValue(i))
            End If

        Next

         If j = 2 Then
                ReturnValue(2) = ((OptionValue(2) - OptionValue(1)) _
                / (S * u^2 - S) - (OptionValue(1) - OptionValue(0)) _
        / (S - S * d^2)) / (0.5 * (S * u^2 - S * d^2))
                ReturnValue(3) = OptionValue(1)
        End If
         If j = 1 Then
                ReturnValue(1) = (OptionValue(1) - OptionValue(0)) _
                  / (S * u - S * d)
         End If
    Next
    ReturnValue(3) = (ReturnValue(3) - OptionValue(0)) / (2 * dt) / 365
    ReturnValue(0) = OptionValue(0)
    If OutputFlag = "p" Then 'Option value
        BinomialCompoundOption = ReturnValue(0)
    ElseIf OutputFlag = "d" Then 'Delta
        BinomialCompoundOption = ReturnValue(1)
    ElseIf OutputFlag = "g" Then 'Gamma
        BinomialCompoundOption = ReturnValue(2)
    ElseIf OutputFlag = "t" Then 'Theta
        BinomialCompoundOption = ReturnValue(3)
   ElseIf OutputFlag = "a" Then
        BinomialCompoundOption = Application.Transpose(ReturnValue())
    End If

End Function
```

7.4.2 Barrier Options Using Brownian Bridge Probabilities

We can easily value a large number of barrier options combining Brownian bridge barrier hit probabilities in a binomial or trinomial tree. The probability that the asset has hit a barrier when it reaches a given end node in a binomial tree is equal to the probability of reaching the end node multiplied by the barrier hit probability for that node.

The probability of reaching an end node is

$$\frac{n!}{i!(n-i)!}p^i(1-p)^{n-i}$$

and the probability of hitting a barrier from below is

$$p^H(i,n) = \begin{cases} \exp\left\{\dfrac{-2}{\sigma^2 n\Delta t}\left|\ln\left(\dfrac{S}{H}\right)\ln\left(\dfrac{Su^i d^{n-i}}{H}\right)\right|\right\} & \text{when } Su^i d^{n-i} < H \\ 1 & \text{when } Su^i d^{n-i} \geq H \end{cases}$$

The probability of a barrier hit in the option's lifetime for a given node is given by

$$p^H = \sum_{i=0}^{n} \frac{n!}{i!(n-i)!}p^i(1-p)^{n-i}p^H(i,n)$$

In the case where the barrier is initially lower than the asset price and the option is down-and-in or down-and-out, simply replace the barrier hit probability for each node by

$$p_H(i,n) = \begin{cases} \exp\left\{\dfrac{-2}{\sigma^2 n\Delta t}\left|\ln\left(\dfrac{S}{H}\right)\ln\left(\dfrac{Su^i d^{n-i}}{H}\right)\right|\right\} & \text{when } Su^i d^{n-i} > H \\ 1 & \text{when } Su^i d^{n-i} \leq H \end{cases}$$

This is a very accurate way to take into account barriers in a binomial or trinomial tree, and it is also very easy to implement.

Computer algorithm

The code below is a binomial tree that can be used to calculate European up-and-out options as well as down-and-out options. By changing the payoff function slightly you will be able to value all types of exotic options with a barrier, for example, powered barrier options. The code below also takes into account a cash rebate that you get at maturity if the option is knocked-out.

```
Function BinomialBridgeBarrier(CallPutFlag As String, S As Double, _
    X As Double, H As Double, Rebate As Double, T As Double, _
                r As Double, b As Double, v As Double, n As Integer) As Double

    Dim OptionValue As Double
    Dim u As Double, d As Double, p As Double
    Dim dt As Double
    Dim i As Integer, z As Integer
    Dim BarrierHitProb As Double, RebateValue As Double
    Dim St As Double, PathProb As Double

    If CallPutFlag = "c" Then
        z = 1
```

```
        ElseIf CallPutFlag = "p" Then
        z = -1
    End If

    dt = T / n
    u = Exp((b - v^2 / 2) * dt + v * Sqr(dt))
    d = Exp((b - v^2 / 2) * dt - v * Sqr(dt))
    p = 0.5

    OptionValue = 0
    PathProb = 0

    For i = 0 To n
            St = S * u^i * d^(n - i)
          If S > H Then
          '//Probability of hitting  barrier below
                If St <= H Then
                      BarrierHitProb = 1
                Else
                      BarrierHitProb = Exp(-2 / (v^2 * T) _
                        * Abs(Log(H / S) * Log(H / St)))
                End If
            ElseIf S < H Then
            '// Probability of hitting the barrier above
                If St >= H Then
                    BarrierHitProb = 1
                Else
                    BarrierHitProb = Exp(-2 / (v^2 * T) _
                    * Abs(Log(S / H) * Log(St / H)))
                End If
            End If

            PathProb = Application.Combin(n, i) * p^i * (1 - p)^(n - i)
            OptionValue = OptionValue + (1 - BarrierHitProb) _
        * PathProb * Max(0, z * (St - X))
            RebateValue = RebateValue + BarrierHitProb * Rebate * PathProb
    Next

            BinomialBridgeBarrier = (OptionValue + RebateValue) * Exp(-r * T)

End Function
```

7.4.3 American Barrier Options in CRR Binomial Tree

The analytical barrier option pricing formulas presented in Chapter 4 are applicable only to European options. As demonstrated above, it is straightforward to value a large variety of complex European barrier options in a European binomial tree.

American barrier options, on the other hand, can only be priced in full-grown binomial trees. The reason is that we need to check if the option is optimal to exercise at any node when rolling back through the tree. For European options, we are only concerned by the end nodes. The accuracy will, in general, increase with the number of time steps

TABLE 7-1

Number of Time Steps to Use in a Down-and-Out Option

($S = 50, H = 45, T = 0.5, \sigma = 0.36$)

i	1	2	3	4	5	6	7	8	9	10
$F(i)$	5	23	52	93	145	210	286	373	472	583

in the tree when you are pricing standard options in binomial trees. However, when you are pricing barrier options, this is not necessarily true. If the barrier does not coincide with the nodes, monitoring of the barrier in the tree will be inaccurate. Boyle and Lau (1994) show that it is possible to adjust the number of time steps in a CRR binomial tree to make the barrier fall exactly on or very close to the nodes. To achieve this, the number of time steps should be set equal to

$$F(i) = \frac{i^2\sigma^2 T}{[\ln(\frac{S}{H})]^2} \qquad i = 1, 2, 3 \ldots \tag{7.15}$$

Example

Consider pricing a down-and-out option in a CRR binomial tree. Time to expiration is six months, the stock price is 50, the barrier is 45, and the volatility is 36%. How many time steps should you use to get an accurate value? Using Equation (7.15) and the input parameters above results in the values shown in Table 7-1. To attain maximum accuracy for the value of this specific barrier option, one should use 5, 23, 52, ... time steps for the corresponding 1, 2, 3, ... steps i used.

There are several limitations to this method: It works only for options with a single barrier, and the published approach admits only constant volatility. Also, if the barrier level varies over time or we are using a tree with other parameters than the CRR tree, the Boyle and Lau (1994) method will not work properly. In such cases there is a much more flexible method developed by Derman, Bardhan, Ergener, and Kani (1995) that can be used for almost any tree model.

For an American-style down-and-in call option or up-and-in put (standard barrier option) the method described under the section "Standard American Barrier Options" in Chapter 4, can be used in combination with binomial or trinomial trees. Computer code and Excel spreadsheet using a trinomial tree are included on the accompanying CD.

7.4.4 European Reset Options Binomial

Haug and Haug (2001) developed a modified binomial tree that can be used to value various reset options. In the setting of Rendleman and Bartter (1979), the probability of going up or down in a node is set equal to $\frac{1}{2}$. One could alternatively have used the approach of Cox, Ross, and Rubinstein (1979). With the former choice, if p is the probability that the asset price moves up and $(1-p)$ is the probability it moves down, then $p = \frac{1}{2} = (1-p)$. The corresponding sizes of the up and down moves at each time step, Δt apart, are

$$u = e^{(b-\sigma^2/2)\Delta t + \sigma\sqrt{\Delta t}}, \qquad d = e^{(b-\sigma^2/2)\Delta t - \sigma\sqrt{\Delta t}}$$

Let n denote the number of time steps and i the state.

European Reset Strike Options

In a plain vanilla reset call (put) option, the strike is reset to the asset price at a predetermined future time if the asset price is below (above) the initial strike price. The strike can more generally be reset to any function of the asset price at future dates. This makes the strike path-dependent.

For a European reset strike option, the value is simply equal to the sum of the payoffs multiplied by the corresponding path-dependent probabilities, discounted at the risk-free interest rate:

$$c_r(S, X) = e^{-rT} \sum_{j=0}^{m} \sum_{i=j}^{n-m+j} \frac{m!(n-m)!}{j!(m-j)!(i-j)!(n-m-i+j)!} \left(\frac{1}{2}\right)^n g(Su^i d^{n-i}, X_c),$$

where m is the number of time steps until the reset time, n is the number of time steps to maturity and $g(S, X) = \max[S - X, 0]$, and $X_c = \min[\alpha S u^j d^{m-j}, X]$. The constant α determines how much in- or out-of-the-money the reset strike is.

The method can be made more efficient by considering only paths that are in-the-money at maturity. This is achieved by replacing $\sum_{i=j}^{n-m+j}$ with $\sum_{i=a_c(j)}^{n-m+j}$, where $a_c(j) = j + \max[0, a_j + 1]$,

$$a_j = \text{Int}\left\{\frac{\ln(X_c) - \ln(Su^j d^{n-j})}{\ln(u) - \ln(d)}\right\},$$

and $\text{Int}(x)$ is the integer part of any real number x.[4]

A put reset option p_r is valued by the same expression as for c_r, but with $g(S, X) = \max[X - S, 0]$ and $X_p = \max[\alpha S u^j d^{m-j}, X]$. Similar

[4]To derive $a_c(j)$, consider the security price at state (j, m), $S_\tau = Su^j d^{m-j}$, and solve for those i that ensure that $S_T > X_c$: At the subtree starting at (j, m), we initially sum over $j \le i \le n - m + j$. Substituting $k = i - j$, $S_T = S_\tau u^k d^{(n-m)-k}$ where $0 \le k \le n - m$. The truncation a_j is obtained by solving the inequality $S_T > X_c$ for k, ensuring it is an integer.

TABLE 7-2

Comparison of Closed-Form Solution and the Binomial Reset Strike Method

The parameters used are $S = 100$, $X = 100$, $T = 1$, $r = 0.1$, $b = 0.1$, and $n = 150$. BS denotes the formula of Black and Scholes, for plain vanilla options; GW denotes the reset formula of Gray and Whaley; HH denotes the binomial reset method of Haug and Haug.

		$\tau = 0.25$		$\tau = 0.5$		$\tau = 0.75$	
Volatility	BS	GW	HH	GW	HH	GW	HH
				Call options			
10%	10.3081	10.9337	10.9293	10.8329	10.8384	10.6559	10.6626
20%	13.2697	14.5827	14.5935	14.5658	14.5846	14.2735	14.2936
30%	16.7341	18.5853	18.6258	18.6627	18.6637	18.3078	18.3288
				Put options			
10%	0.7919	1.1690	1.1649	1.3729	1.3794	1.4969	1.5063
20%	3.7534	4.9384	4.9539	5.3269	5.3511	5.3790	5.4064
30%	7.2179	9.2388	9.2876	9.7922	9.8030	9.7543	9.7854

to the call, the formula for the put can be made more efficient by only taking into account paths that are in-the-money at maturity: replace $\sum_{i=j}^{n-m+j}$ with $\sum_{i=j}^{a_p(j)}$, where $a_p(j) = \min[n - m + j, j + \max[0, a_j]]$

For a standard reset strike option, one could have used the closed-form formula published by Gray and Whaley (1999); see Chapter 4 on exotic options. The main advantage of the approach suggested here is its flexibility. Consider, for instance, a reset power option, with payoff at maturity equal to $g(S, X) = \max[S^2 - X^2, 0]$. This contract is easily valued by simply replacing the payoff function.

Table 7-2 compares the closed-form solution of Gray and Whaley with the Haug and Haug (2001) binomial method, using 150 time steps. The first column reports values of plain vanilla options, using the formula of Black and Scholes (1973). It is clear that the Haug and Haug method yields values that are very close to those of the closed-form solution.

Reset Strike Barrier Options

We now consider barrier options with strikes that are reset. These are standard reset options where the strike can be reset at a predefined future date, with one or more predefined barriers that can knock the option in or out.

To value these options it is necessary to compute the probability of hitting a barrier H. The probability of hitting the barrier along each path can be found analytically using Brownian bridge probabilities; see Brockhaus, Ferraris, Gallus, Long, Martin, and Overhaus (1999). Consider first an up-and-out call option where the barrier covers the entire life of the option. If the asset price hits the barrier before maturity, the option expires worthless. Its value is given by

$$c_{rb}(S, X) = e^{-rT} \sum_{j=0}^{m} \sum_{i=j}^{n-m+j} \frac{m!(n-m)!}{j!(m-j)!(i-j)!(n-m-i+j)!} \times \left(\frac{1}{2}\right)^n g(Su^i d^{n-i}, X_c)[1 - p^H(j, m; i, n-m)] \tag{7.16}$$

where m is the number of time steps until the reset time, n is the number of time steps to maturity, and $g(S, X) = \max[S - X, 0]$, $X_c = \min[\alpha S u^j d^{m-j}, X]$, $p^H(j, m; i, n-m) = p^H(j, m) + p^H(i, n-m) - p^H(j, m)p^H(i, n-m)$ is the probability of hitting the barrier when going through (j, m) and ending up in (i, n), and

$$p^H(i, n) = \begin{cases} \exp\left\{\dfrac{-2}{\sigma^2 n \Delta t}\left|\ln\left(\dfrac{S}{H}\right)\ln\left(\dfrac{Su^i d^{n-i}}{H}\right)\right|\right\} & \text{when } Su^i d^{n-i} < H \\ 1 & \text{when } Su^i d^{n-i} \geq H \end{cases}$$

The probability of hitting a barrier that covers only the period after the reset is given by $p^H(i, n-m)$. Similarly, the probability for hitting a barrier that covers only the time period before the reset time is given by $p^H(j, m)$.

Down-and-out barriers can be incorporated in the same way, by using the appropriate probabilities of barrier hits. Simply define

$$p_H(i, n) = \begin{cases} \exp\left\{\dfrac{-2}{\sigma^2 n \Delta t}\left|\ln\left(\dfrac{S}{H}\right)\ln\left(\dfrac{Su^i d^{n-i}}{H}\right)\right|\right\} & \text{when } Su^i d^{n-i} > H \\ 1 & \text{when } Su^i d^{n-i} \leq H, \end{cases}$$

and $p_H(j, m; i, n-m)$ as above. Now simply substitute the latter in place of $p^H(j, m; i, n-m)$. For a barrier that covers only the period after the reset, the probability of hitting the barrier is given by $p_H(i, n-m)$. A barrier that covers only the period before the reset induces a hit probability equal to $p_H(j, m)$.

If the contract specifies different barriers H_S and H_E for the periods before and after reset, respectively, the probability of not hitting any of the barriers is $1 - p_{H_S}(j, m) - p_{H_E}(i, n-m) + p_{H_S}(j, m)p_{H_E}(i, n-m)$ (this works for up-and-out options too, of course). For barrier put options, it is only necessary to change the payoff function $g(S, X)$ from $\max[S - X, 0]$ to $\max[X - S, 0]$, in the usual way.

To value in-options one can use the in/out barrier parity:

In-barrier option = Long out-barrier option − Short plain vanilla option

Consider next a cash rebate K that is paid out at maturity to the holder of the option if the barrier is hit during its lifetime. We take this into account by multiplying the rebate by the probability of hitting the barrier, and then discount the product at the risk-free rate: $e^{-rT}Kp^H(i,n)$. Use $p_H(\cdot)$ for a down-and-out option. Use the probability of the complementary event to price a rebate that is paid if the barrier is not hit during the option's lifetime.

In practice, the barrier is monitored at discrete points in time. Discrete monitoring will naturally decrease the probability of barrier hits, relative to continuous monitoring. Broadie, Glasserman, and Kou (1997) developed an approximate correction for pricing formulas for discrete barrier options, as discussed in Chapter 4 on exotic options. It can also be used for our reset binomial method.

Reset Barrier Options

Rather than resetting the strike, we can just as easily reset the barrier. Consider the case of an up-and-out reset barrier option, where the barrier is reset the first time it is hit. If the initial barrier covers the entire lifetime of the option, the option price is given by

$$e^{-rT}\sum_{j=0}^{m}\sum_{i=j}^{n-m+j}\frac{m!(n-m)!}{j!(m-j)!(i-j)!(n-m-i+j)!}\left(\frac{1}{2}\right)^n \times g(Su^id^{n-i},X)[1-p^{H(j)}(j,m;i,n-m)],$$

where $H(j)=\min(\alpha Su^jd^{m-j},H)$, H is the original barrier, and α is a positive constant that determines how the barrier is reset as a function of S_t (this can easily be generalized). Other types of reset barrier options can be valued in a similar way. The necessary adjustments are similar to those in the previous discussion.

Reset Time Options

Someone who is long a call or put when the underlying moves are in an unfavorable direction will benefit from extending the time to maturity of the option. With the above reset binomial method, it is easy to value options where the time to maturity is reset. Consider a call option, and assume the time to maturity is reset at a future date $\tau = m\Delta t$ if the asset price is a predetermined percent out-of-the-money (alternatively, in-the-money). Its price is given by

$$c_r^{\tau}(S, X) = \sum_{j=0}^{m} \sum_{i=j}^{n-m+j} D_c(i, j) \frac{m!(n-m)!}{j!(m-j)!(i-j)!(n-m-i+j)!} \times \left(\frac{1}{2}\right)^n \max[S_c(i, j) - X, 0], \quad (7.17)$$

where

$$S_c(i, j) = \begin{cases} Su^i d^{n-i} & \text{when } \alpha S u^j d^{m-j} < X \\ Su^j d^{m-j} & \text{when } \alpha S u^j d^{m-j} \geq X \end{cases}$$

and

$$D_c(i, j) = \begin{cases} e^{-rn\Delta t} & \text{when } \alpha S u^j d^{m-j} < X \\ e^{-rm\Delta t} & \text{when } \alpha S u^j d^{m-j} \geq X \end{cases}$$

For a put option, the price is given by

$$p_r^{\tau}(S, X) = \sum_{j=0}^{m} \sum_{i=j}^{n-m+j} D_p(i, j) \frac{m!(n-m)!}{j!(m-j)!(i-j)!(n-m-i+j)!} \times \left(\frac{1}{2}\right)^n \max[X - S_p(i, j), 0], \quad (7.18)$$

where

$$D_p(i, j) = \begin{cases} e^{-rn\Delta t} & \text{when } \alpha S u^j d^{m-j} > X \\ e^{-rm\Delta t} & \text{when } \alpha S u^j d^{m-j} \leq X \end{cases}$$

and

$$S_p(i, j) = \begin{cases} Su^i d^{n-i} & \text{when } \alpha S u^j d^{m-j} > X \\ Su^j d^{m-j} & \text{when } \alpha S u^j d^{m-j} \leq X \end{cases}$$

and

$$D_p(i, j) = \begin{cases} e^{-rn\Delta t} & \text{when } \alpha S u^j d^{m-j} > X \\ e^{-rm\Delta t} & \text{when } \alpha S u^j d^{m-j} \leq X \end{cases}$$

Longstaff (1990) derives a closed-form solution for these reset time options, known as writer-extendible options and covered in Chapter 4. As before, the binomial reset method offers more flexibility and can be used to value a larger class of reset time options. With the latter method, reset time options can be combined with reset strikes and barriers. One can, for instance, extend the method to a call where the time, strike, and barrier are reset as a function of the asset price at a predetermined future date. Table 7-3 gives an indication of how accurate this method is. It compares the method just presented to Longstaff's closed-form solution.

For a given number of time steps, the reset time method is somewhat less accurate than the reset strike method. In other words, it

TABLE 7-3

Comparison of Closed-Form Solution and the Binomial Reset Time Method

The parameters used are $S = 100$, $X = 100$, $T = 1$, $r = 0.1$, $b = 0.1$, and $n = 200$. Longstaff denotes the formula of Longstaff (1990). HH denotes the binomial reset time method of Haug and Haug.

	$\tau = 0.25$		$\tau = 0.5$		$\tau = 0.75$	
Volatility	Longstaff	HH	Longstaff	HH	Longstaff	HH
			Call options			
10%	5.2526	5.4007	6.6644	6.6167	8. 3941	8.3271
20%	8.1299	8.3793	9.7562	9.7096	11.4691	11.3562
30%	10.9954	10.7393	12.9565	13.0889	14.8446	15.0490
			Put options			
10%	1.2216	1.2112	1.1417	1.1532	0.9844	1.0015
20%	3.8839	3.8111	4.0780	4.1074	3.9962	4.0557
30%	6.7096	6.8574	7.2482	7.1952	7.3589	7.2203

is slower to converge. This is especially pronounced when the time to reset is short relative to the time to maturity. In this case, the grid is too coarse at the time of reset. This can be ameliorated either by increasing the total number of time steps in the tree or by locally making the grid finer (Figlewski and Gao, 1999). The latter technique offers better computational efficiency but is more complex to implement.

7.4.5 American Asian Options in a Tree

Valuing an arithmetic average rate option in a standard lattice will, under normal circumstances, lead to a nonrecombining tree. The number of nodes in nonrecombining trees grows exponentially with the number of time steps. These trees, therefore, tend to be slow in many applications. To my knowledge, no closed-form solution has been published for American Asian options.[5] Valuing American-style Asian options (aka Hawaiian options) requires a special tree implementation, first described by Hull and White (1993) and later improved by Cho and Lee (1997), and Chalasani, Jha, Egriboyun, and Varikooty (1999). The method reduces the number of states by using an interpolation technique when "rolling back" through the tree. The method is reasonably efficient but is still only an approximation that does

[5] Hansen and Jørgensen (1997) have published a model for American Asian options using numerical integration.

not necessarily converge to the analytical solution as the number of time steps is increased. Dai, Huang, and Lyuu (2002) claim that the Hull and White (1993) value represents an upper bound for the option value. Computer code in VBA and an Excel spreadsheet for the Hull-White Asian option tree is included on the accompanying CD. See also the book of Clewelow and Strickland (1998) for more details on implementing the method. Dai and Lyuu (2002) have developed a more accurate method that seems to be preferable to the methods just discussed.

7.5 THREE-DIMENSIONAL BINOMIAL TREES

Rubinstein (1994b) has published a method to construct a three-dimensional binomial model that can be used to price most types of options that depend on two assets—both European and American.[6] Examples of options that can be valued are as follows:

Spread options:	call: $\max[0, Q_1S_1 - Q_2S_2 - X]$
	put: $\max[0, X + Q_2S_2 - Q_1S_1]$
Options on the maximum:	call: $\max[0, \max(Q_1S_1, Q_2S_2) - X]$
	put: $\max[0, X - \max(Q_1S_1, Q_2S_2)]$
Options on the minimum:	call: $\max[0, \min(Q_1S_1, Q_2S_2) - X]$
	put: $\max[0, X - \min(Q_1S_1, Q_2S_2)]$
Dual-strike options:	call: $\max[0, (Q_1S_1 - X_1), (Q_2S_2 - X_2)]$
	put: $\max[0, (X_1 - Q_1S_1), (X_2 - Q_2S_2)]$
Reverse dual-strike options:	call: $\max[0, (Q_1S_1 - X_1), (X_2 - Q_2S_2)]$
	put: $\max[0, (X_1 - Q_1S_1), (Q_2S_2 - X_2)]$
Portfolio options:	call: $\max[0, (Q_1S_1 + Q_2S_2) - X]$
	put: $\max[0, X - (Q_1S_1 + Q_2S_2)]$
Options to exchange one asset for another:	$\max[0, Q_2S_2 - Q_1S_1]$
Relative performance options:	call: $\max\left[0, \frac{Q_1S_1}{Q_2S_2} - X\right]$
	put: $\max\left[0, X - \frac{Q_1S_1}{Q_2S_2}\right]$
Product options:	call: $\max[0, Q_1S_1Q_2S_2 - X]$
	put: $\max[0, X - Q_1S_1Q_2S_2]$

where Q_1 and Q_2 are the fixed quantities of the two different assets. In the three-dimensional binomial model, asset 1 can increase with a factor of u or decrease with a factor of d at each time step. If asset

[6]An alternative would be to use a three-dimensional trinomial tree as described by Boyle (1988) and later simplified by Cho and Lee (1995).

TABLE 7-4

Examples of Call Values

The parameters used are $S_1 = 122$, $S_2 = 120$, $X = 3, r = 0.1, b_1 = b_2 = 0$, and $n = 100$

		$T = 0.1$			$T = 0.5$		
σ_1	σ_2	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$
		European Values					
0.20	0.20	4.7554	3.8008	2.5551	10.7566	8.7080	6.0286
0.25	0.20	5.4297	4.3732	3.0098	12.2031	9.9377	7.0097
0.20	0.25	5.4079	4.3469	2.9743	12.1521	9.8811	6.9323
		American Values					
0.20	0.20	4.7630	3.8067	2.5590	10.8754	8.8029	6.0939
0.25	0.20	5.4385	4.3802	3.0145	12.3383	10.0468	7.0858
0.20	0.25	5.4166	4.3538	2.9790	12.2867	9.9897	7.0082

1 increases with u, asset 2's price can change by A or B. If asset 1 decreases with d, asset 2's price can change by C or D. By setting $A \neq C$ and $B \neq D$, one can construct nonzero correlation between the two assets.

$$u = \exp(\mu_1 \Delta t + \sigma_1 \sqrt{\Delta t}) \qquad d = \exp(\mu_1 \Delta t - \sigma_1 \sqrt{\Delta t}),$$

where

$$\mu_1 = b_1 - \sigma_1^2/2 \qquad \mu_2 = b_2 - \sigma_2^2/2$$

and

$$A = \exp[\mu_2 \Delta t + \sigma_2 \sqrt{\Delta t}(\rho + \sqrt{1 - \rho^2})]$$

$$B = \exp[\mu_2 \Delta t + \sigma_2 \sqrt{\Delta t}(\rho - \sqrt{1 - \rho^2})]$$

$$C = \exp[\mu_2 \Delta t - \sigma_2 \sqrt{\Delta t}(\rho - \sqrt{1 - \rho^2})]$$

$$D = \exp[\mu_2 \Delta t - \sigma_2 \sqrt{\Delta t}(\rho + \sqrt{1 - \rho^2})]$$

Table 7-4 shows European and American spread option values generated using a three-dimensional binomial tree with 100 time steps.

Number of Nodes in Two-Factor Binomial Tree

By counting the number of time steps from 0, as we have been doing above, we find that the number of nodes in the two-factor binomial tree is $\frac{(n+1)(n+2)(2n+3)}{6}$. Similarly, counting the number of time steps from 1 results in $\frac{n(n+1)(2n+1)}{6}$ nodes.

Application

American spread options are traded on the New York Mercantile Exchange on oil products. The heating oil crack is an option on the spread between heating oil and crude oil. The gasoline crack is an option on the spread between unleaded gasoline and crude oil. Crack spread options are useful to oil refineries for hedging purposes.

Maximum, minimum, and dual-strike options have been traded in the OTC market on commodities and stock indexes. The three-dimensional binomial model is extremely flexible. Adding a new type of option to the model can, in most cases, be achieved by a small adjustment in the payoff function.

Computer algorithm

The function can be used to build a complete three-dimensional binomial tree. Setting the value of *TypeFlag* determines what kind of option value that is estimated: *TypeFlag* = 1 returns the value of a spread option, *TypeFlag* = 2 returns the value of an option on the maximum of two assets, *TypeFlag* = 3 returns the value of an option on the minimum of two assets, *TypeFlag* = 4 returns the value of a dual-strike option, *TypeFlag* = 5 returns the value of a reverse dual-strike option, *TypeFlag* = 6 gives the value of a two-asset portfolio option, *TypeFlag* = 7 returns the value of an option to exchange one asset for another, *TypeFlag* = 8 gives the value of an outperformance option, while *TypeFlag* = 9 returns the value of an product option. Setting the *AmeEurFlag* equal to "*e*" gives European option values, and "*a*" gives American option values. Setting the *CallPutFlag* equal to "*c*" or "*p*" gives a call or put value, respectively.

```
Public Function ThreeDimensionalBinomial(TypeFlag As Integer, _
  AmeEurFlag As String, CallPutFlag As String, S1 As Double, _
  S2 As Double, Q1 As Double, Q2 As Double, X1 As Double, _
  X2 As Double, T As Double, r As Double, _
  b1 As Double, b2 As Double, v1 As Double, v2 As Double, _
  rho As Double, n As Integer) As Double

    Dim OptionValue() As Double
    Dim dt As Double, u As Double, d As Double
    Dim mu1 As Double, mu2 As Double
    Dim Y1 As Double, y2 As Double
    Dim NodeValueS1 As Double, NodeValueS2 As Double
    Dim i As Integer, j As Integer, m As Integer

    ReDim OptionValue(0 To n + 1, 0 To n + 1)

    dt = T / n
    mu1 = b1 - v1^2 / 2
    mu2 = b2 - v2^2 / 2
    u = Exp(mu1 * dt + v1 * Sqr(dt))
    d = Exp(mu1 * dt - v1 * Sqr(dt))

    For j = 0 To n
```

```
        Y1 = (2 * j - n) * Sqr(dt)
        NodeValueS1 = S1 * u^j * d ^ (n - j)
        For i = 0 To n
            NodeValueS2 = S2 * Exp(mu2 * n * dt) * Exp(v2 * (rho * Y1_
     + Sqr(1 - rho^2) * (2 * i - n) * Sqr(dt)))
            OptionValue(j, i) = PayoffFunction(TypeFlag, _
            CallPutFlag, NodeValueS1, NodeValueS2, Q1, Q2, X1, X2)
        Next
    Next
    For m = n - 1 To 0 Step -1
        For j = 0 To m
            Y1 = (2 * j - m) * Sqr(dt)
            NodeValueS1 = S1 * u^j * d^(m - j)
            For i = 0 To m
                y2 = rho * Y1 + Sqr(1 - rho^2) * (2 * i - m) * Sqr(dt)
                NodeValueS2 = S2 * Exp(mu2 * m * dt) * Exp(v2 * y2)
                OptionValue(j, i) = 0.25 * (OptionValue(j, i) _
                  + OptionValue(j + 1, i) + OptionValue(j, i + 1) _
                  + OptionValue(j + 1, i + 1)) * Exp(-r * dt)
                If AmeEurFlag = "a" Then
                    OptionValue(j, i) = Max(OptionValue(j, i), _
                    PayoffFunction(TypeFlag, CallPutFlag, NodeValueS1, _
                    NodeValueS2, Q1, Q2, X1, X2))
                End If
            Next
        Next
    Next

    ThreeDimensionalBinomial = OptionValue(0, 0)

End Function

Function PayoffFunction(TypeFlag As Integer, CallPutFlag As String, _
  S1 As Double, S2 As Double, Q1 As Double, Q2 As Double, _
    X1 As Double, X2 As Double) As Double

    Dim z As Integer


    If CallPutFlag = "c" Then
        z = 1
        ElseIf CallPutFlag = "p" Then
        z = -1
    End If

    If TypeFlag = 1 Then '// Spread option
        PayoffFunction = Max(0, z * (Q1 * S1 - Q2 * S2) - z * X1)
    ElseIf TypeFlag = 2 Then '// Option on the maximum of two assets
        PayoffFunction = Max(0, z * Max(Q1 * S1, Q2 * S2) - z * X1)
    ElseIf TypeFlag = 3 Then '// Option on the minimum of two assets
        PayoffFunction = Max(0, z * Min(Q1 * S1, Q2 * S2) - z * X1)
    ElseIf TypeFlag = 4 Then '// Dual strike option
        PayoffFunction = Application.Max(0, z * (Q1 * S1 - X1), _
          z * (Q2 * S2 - X2))
    ElseIf TypeFlag = 5 Then '// Reverse-dual strike option
        PayoffFunction = Application.Max(0, z * (Q1 * S1 - X1), _
          z * (X2 - Q2 * S2))
    ElseIf TypeFlag = 6 Then '// Portfolio option
        PayoffFunction = Max(0, z * (Q1 * S1 + Q2 * S2) - z * X1)
```

```
    ElseIf TypeFlag = 7 Then   '// Exchange option
        PayoffFunction = PayoffFunction = Max(0, Q2 * S2 - Q1 * S1)
  ElseIf TypeFlag = 8 Then   '// Outperformance option
       PayoffFunction = Max(0, z * (Q1 * S1 / (Q2 * S2) - X1))
    ElseIf TypeFlag = 9 Then   '// Product option
       PayoffFunction = Max(0, z * (Q1 * S1 * Q2 * S2 - X1))
   End If

End Function
```

Two-Asset European Binomial

For non-path-dependent European options, it is possible to simplify Rubinstein's three-dimensional binomial tree. In this case, we need only be concerned with the end nodes of the tree when computing option values. This makes the implementation simpler and thus more efficient than for the full version of the method.

The number of paths to a node when counting from zero is

$$\frac{n!n!}{i!(n-i)!j!(n-j)!},$$

where j is the number of price increases of asset one and i is the number of price increases of asset two. The probability of ending at a specific end node is

$$\frac{n!n!}{i!(n-i)!j!(n-j)!}0.25^n$$

The value of any non-path-dependent European two-asset option can then simply be computed as

$$c = e^{-rT}\sum_{j=0}^{n}\sum_{i=0}^{n}\frac{n!n!}{i!(n-i)!j!(n-j)!}0.25^n g[S_1(T), S_2(T)],$$

where

$$S_1(T) = S_1 u^j d^{n-j}$$

$$S_2(T) = S_2\exp[\mu_2 T + \sigma_2(\rho(2j-n) + \sqrt{1-\rho^2}(2i-n))\sqrt{\Delta t}],$$

where

$$u = \exp(\mu_1\Delta t + \sigma_1\sqrt{\Delta t}) \qquad d = \exp(\mu_1\Delta t - \sigma_1\sqrt{\Delta t})$$

$$\mu_1 = b_1 - \sigma_1^2/2 \qquad \mu_2 = b_2 - \sigma_2^2/2,$$

and

$$g[S_1(T), S_2(T)]$$

can be any payoff at maturity that depends on the price of the two assets.

Computer algorithm

The following algorithm implements a three-dimensional European binomial tree. The value of *TypeFlag* determines what kind of option value is being estimated: *TypeFlag* = 1 returns the value of a spread option, *TypeFlag* = 2 returns the value of an option on the maximum of two assets, *TypeFlag* = 3 returns the value of an option on the minimum of two assets, *TypeFlag* = 4 returns the value of a dual-strike option, *TypeFlag* = 5 returns the value of a reverse dual-strike option, *TypeFlag* = 6 gives the value of a two-asset portfolio option, *TypeFlag* = 7 returns the value of an option to exchange one asset for another, *TypeFlag* = 8 gives the value of an outperformance option, while *TypeFlag* = 9 returns the value of an product option. Setting the *CallPutFlag* equal to "*c*" or "*p*" gives a call or put value, respectively.

```
Function ThreeDimensionalBinomiaEuropean(TypeFlag As Integer, _
  AmeEurFlag As String, CallPutFlag As String, S1 As Double, S2 As Double, _
  Q1 As Double, Q2 As Double, X1 As Double, X2 As Double, T As Double, _
  r As Double, b1 As Double, b2 As Double, v1 As Double, v2 As Double, _
  rho As Double, n As Integer) As Double

    Dim dt As Double, u As Double, d As Double
    Dim my1 As Double, my2 As Double
    Dim Y1 As Double, y2
    Dim NodeValueS1 As Double, NodeValueS2 As Double
    Dim i As Integer, j As Integer
    Dim sum As Double
    Dim PatheProbability As Double

    dt = T / n
    my1 = b1 - v1^2 / 2
    my2 = b2 - v2^2 / 2
    u = Exp(my1 * dt + v1 * Sqr(dt))
    d = Exp(my1 * dt - v1 * Sqr(dt))

    For j = 0 To n
        NodeValueS1 = S1 * u^j * d^(n - j)
        For i = 0 To n
             NodeValueS2 = S2 * Exp(my2 * T + v2 * (rho * (2 * j - n) _
        + Sqr(1 - rho^2) * (2 * i - n)) * Sqr(dt))
            PatheProbability = Application.Combin(n, i) _
               * Application.Combin(n, j) * 0.25^n
            sum = sum + PatheProbability _
               * PayoffFunction(TypeFlag, CallPutFlag, _
        NodeValueS1, NodeValueS2, Q1, Q2, X1, X2)
        Next
    Next

    ThreeDimensionalBinomiaEuropean = sum * Exp(-r * T)

End Function
```

7.6 IMPLIED TREE MODELS

The implied tree model represents a development in option pricing that has received much attention both from practitioners and academics. The main idea is to use information from liquid options with different strikes and maturities to build an arbitrage-free model that contains all relevant valuation information implied by market prices. The idea of implied tree models was published in 1994 by Dupire (1994), Derman and Kani (1994), and Rubinstein (1994a). The method was later discussed and extended by Barle and Cakici (1995), Rubinstein (1995a), Derman, Kani, and Chriss (1996), Buchen and Kelly (1996), Chriss (1996), Jackwerth and Rubinstein (1996), and others.

The implied tree model discretizes an asset price process and lets the local volatility be a function of both the price level of the underlying asset and time:

$$dS = \mu(t)Sdt + \sigma(S, t)Sdz$$

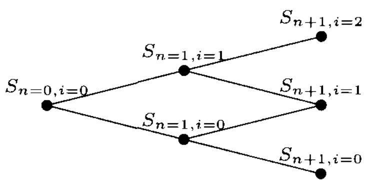
This differs from the geometric Brownian motion used in the Black-Scholes-Merton (BSM) economy:

$$dS = \mu Sdt + \sigma Sdz$$

The local volatility in the BSM economy is constant throughout the lifetime of the option. The volatility function in the implied tree $\sigma(S, t)$ is estimated numerically from the volatility smile given by the prices of liquid options. In this way, the implied tree model is calibrated to be arbitrage-free relative to observed options prices. The model can be used to price exotic options and other derivatives whose prices are not used to calibrate the model. The next two subsections show how to build implied binomial and trinomial trees.

7.6.1 Implied Binomial Trees

In this section we concentrate on the Derman and Kani (1994) implied binomial model. The following figure illustrates an implied binomial tree where the asset prices for time step $n = 1$ has already been solved for.



In general, if we at time n wish to find the unknown parameters at time $n+1$, then there are $2n+3$ unknown parameters to solve for:[7]

- $n+2$ asset prices $S_{n+1,i}$
- $n+1$ unknown risk-neutral transition probabilities $p_{n,i}$ from node (n,i) to node $(n+1, i+1)$

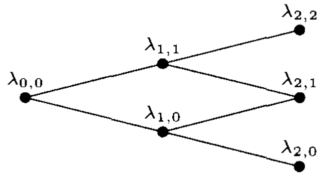
There are $2n+2$ known quantities:

- $n+1$ forward prices $F_{n,i} = S_{n,i}e^{b\Delta t}$
- $n+1$ option prices $c_{n,i}$ expiring at time T_{n+1}

where n is the time step and i is the state. The remaining degree of freedom is used to center the implied binomial tree around the center of the standard Cox, Ross, and Rubinstein (1979) binomial tree.

The implied binomial tree is constructed by forward induction and the use of Arrow-Debreu prices λ. Arrow-Debreu prices represent prices of primitive securities: $\lambda_{n,i}$ is the price today of a security that has a cash flow of unity in state i at time n and a cash flow of zero elsewhere. The Arrow-Debreu prices for the next step $n+1$ are given by

$$e^{r\Delta t}\lambda_{n+1,i} = \begin{cases} p_{n,n}\lambda_{n,n} & \text{when } i = n+1 \\ p_{n,i-1}\lambda_{n,i-1} + (1-p_{n,i})\lambda_{n,i} & \text{when } 1 \le i \le n+1 \\ (1-p_{n,0})\lambda_{n,0} & \text{when } i = 0 \end{cases}$$



By solving for the $(2n+3)$ unknown parameters using the $(2n+2)$ known parameters and by centering the tree around the center of the CRR tree, Derman and Kani get the following equations for the asset prices above the central node:

$$S_{n+1,i+1} = \frac{S_{n+1,i}\left[e^{r\Delta t}c(S_{n,i};T_{n+1}) - \sum\right] - \lambda_{n,i}S_{n,i}(F_{n,i} - S_{n+1,i})}{\left[e^{r\Delta t}c(S_{n,i};T_{n+1}) - \sum\right] - \lambda_{n,i}(F_{n,i} - S_{n+1,i})}, \quad (7.19)$$

[7]This is somewhat different from the Derman and Kani (1994) paper, which has $2n+1$ unknown parameters. The reason is that we start counting the initial node as time step 0, not 1 as in the Derman and Kani paper. The result is, of course, the same. We do this to be consistent with the other tree models described in this chapter.

where $\sum = \sum_{j=i+1}^{n} \lambda_{n,j}(F_{n,j} - S_{n,i})$ and $F_{n,j} = S_{n,j}e^{b\Delta t}$. The term $c(S_{n,i}; T_{n+1})$ is the price of a European call with strike equal to the known asset price $S_{n,i}$ and time to maturity T_{n+1}. The asset prices below the central node is given by

$$S_{n+1,i} = \frac{S_{n+1,i+1}\left[e^{r\Delta t}p(S_{n,i}; T_{n+1}) - \sum\right] + \lambda_{n,i}S_{n,i}(F_{n,i} - S_{n+1,i+1})}{\left[e^{r\Delta t}p(S_{n,i}; T_{n+1}) - \sum\right] + \lambda_{n,i}(F_{n,i} - S_{n+1,i+1})}, \tag{7.20}$$

where $\sum = \sum_{j=0}^{i-1} \lambda_{n,j}(S_{n,i} - F_{n,j})$, and $p(S_{n,i}; T_{n+1})$ is the price of a European put with strike equal to the known asset price $S_{n,i}$ and time to maturity T_{n+1} (not to be confused with the probability $p_{n,i}$). The transition probability at any time step is given by

$$p_{n,i} = \frac{F_{n,i} - S_{n+1,i}}{S_{n+1,i+1} - S_{n+1,i}} \tag{7.21}$$

How to Start Building a New Level of the Tree

If we know $S_{n+1,i}$ at one initial node, we can use these equations to find the implied asset price for all nodes above and below the center of the tree.

1. If the number of time steps n already solved for is odd: The initial central node $S_{n+1,i}$ is set equal to the central node of the CRR tree.
2. If the number of time steps n already solved for is even: Use the logarithmic CRR centering condition $S_{n+1,i} = S^2/S_{n+1,i+1}$, where S is today's asset price. Substituting this relation into (7.19) gives

$$S_{n+1,i+1} = \frac{S\left[e^{r\Delta t}c(S; T_{n+1}) + \lambda_{n,i}S - \sum\right]}{\lambda_{n,i}F_{n,i} - e^{r\Delta t}c(S; T_{n+1}) + \sum}, \tag{7.22}$$

   where

$$\sum = \sum_{j=i+1}^{n} \lambda_{n,j}(F_{n,j} - S_{n,i})$$

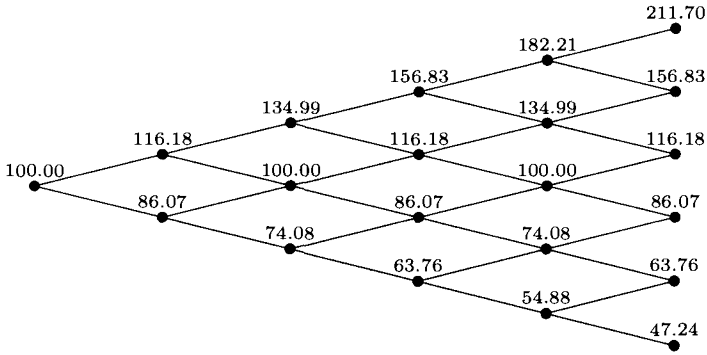
From the implied asset prices and probabilities, we can find the implied local volatilities at each node:

$$\sigma_{n,i} = \sqrt{p_{n,i}(1 - p_{n,i})}\ln(S_{n+1,i+1}/S_{n+1,i})$$

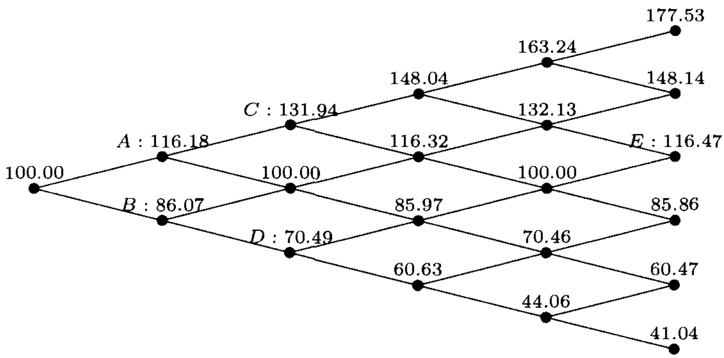
Implied Binomial Tree Example

Construct a five-year implied binomial tree with annual time steps. Assume that the stock price is 100, the risk-free rate is 5%, and the at-the-money volatility is 15% and increases (decreases) 0.5% with every 10-point drop (rise) in the strike price. The 90 strike will trade for 15.5% volatility, and the 110 strike for 14.5% volatility. The results below show a standard binomial tree together with the implied binomial tree, its transition probabilities, Arrow-Debreu prices, and local volatilities at each node.

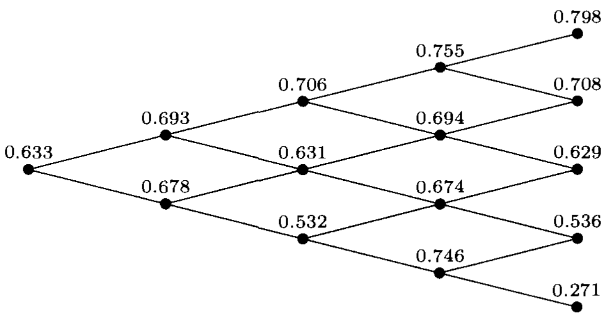
CRR binomial tree with constant volatility 15% where $u = e^{0.15\sqrt{1}} = 1.162$, $d = \frac{1}{u} = 0.861$, and a constant up probability equal to $p = \frac{e^{0.05\times 1} - 0.861}{1.162 - 0.861} = 0.633$:



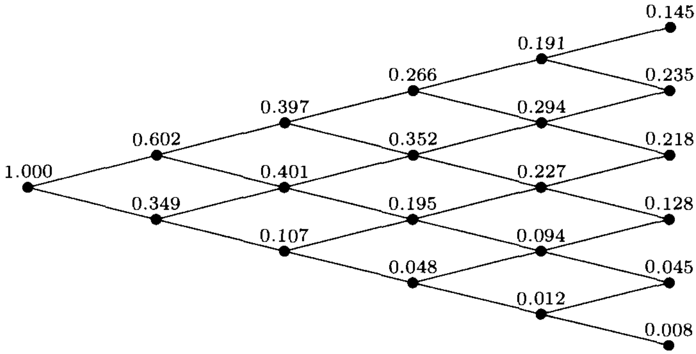
Derman-Kani Implied Binomial Tree



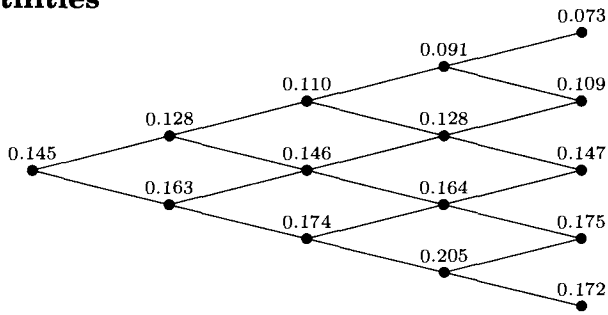
Transition Probabilities



Arrow-Debreu Price Tree



Local Volatilities



To illustrate the procedure of constructing an implied binomial tree, let's look at the calculations of a few nodes in detail. We start with the calculation of the implied stock price at node A:

$$S_A = S_{1,1} = \frac{100\left[e^{0.05\times 1}c(S;T_{n+1}) + 1\times 100 - \sum\right]}{1\times F_{n,i} - e^{0.05\times 1}c(S;T_{n+1}) + \sum} = 116.18,$$

where $c(X = 100; T = 1; \sigma = 15\%) = 9.74$ and $F_{n,i} = F_{0,0} = 100e^{0.05\times 1} = 105.13$. Since there are no stock prices above this node, the $\sum$-term is zero. The implied stock price at node B is

$$S_B = S_{1,0} = S^2/S_{1,1} = 100^2/116.18 = 86.07,$$

while the up transition probability leading to node A is

$$p_{0,0} = \frac{105.13 - 86.07}{116.18 - 86.07} = 0.633$$

The Arrow-Debreu prices are thus

$$\lambda_{1,1} = 0.633 \times 1e^{-0.05\times 1} = 0.602$$

and

$$\lambda_{1,0} = (1 - 0.633)1e^{-0.05\times 1} = 0.349$$

We can now solve for the stock prices and transition probabilities at the next time step. Since the number of time steps $n = 1$ already solved for is odd, the initial stock price is set equal to the central node of the CRR tree $S_{2,1} = 100$. The stock prices above the center node can now easily be found by using equation (7.19).

$$S_C = S_{2,2} = \frac{100\left[e^{0.05\times 1}c(S_{n,i}; T_{n+1}) - \sum\right] - 0.602 \times 116.18(F_{n,i} - 100)}{\left[e^{0.05\times 1}c(S_{n,i}; T_{n+1}) - \sum\right] - 0.0602(F_{n,i} - 100)}$$
$$= 131.94,$$

where $c(X = 116.18; T = 2; \sigma = 14.19\%) = 6.25$, and $F_{n,i} = F_{1,1} = 116.18e^{0.05\times 1} = 122.14$. Since there are no stock prices above this node, the $\sum$-term is zero. Below the central node we get

$$S_D = S_{2,0} = \frac{100\left[e^{0.05\times 1}p(S_{n,i}; T_{n+1}) - \sum\right] + 0.349 \times 86.07(F_{n,i} - 100)}{\left[e^{0.05\times 1}p(S_{n,i}; T_{n+1}) - \sum\right] + 0.349(F_{n,i} - 100)}$$
$$= 70.49,$$

where $p(X = 86.07; T = 2; \sigma = 15.7\%) = 1.67$ and $F_{n,i} = F_{1,0} = 86.07e^{0.05\times 1} = 90.48$. Since there are no stock prices below this node, the $\sum$-term is zero. The two transition probabilities leading to these nodes are

$$p_{1,1} = \frac{122.14 - 100.00}{131.94 - 100.00} = 0.693$$
$$p_{1,0} = \frac{90.48 - 70.49}{100.00 - 70.49} = 0.678$$

At node E, the number of time steps $n + 1 = 5$ is odd. We can now find the implied stock price using equation (7.22):

$$S_E = S_{5,3} = \frac{100[e^{0.05\times 1}c(S; T_{n+1}) + 0.227 \times 100 - \sum]}{0.227F_{n,i} - e^{0.05\times 1}c(S; T_{n+1}) + \sum} = 116.47,$$

where $c(X = 100.00; T = 5; \sigma = 15\%) = 26.17$ and $F_{n,i} = 100e^{0.05\times 1} = 105.13$. Since there are two nodes above this node, the $\sum$-term is equal to

$$\sum = \sum_{j=2+1}^{4} = 0.294(132.13e^{0.05\times 1} - 100)$$
$$+ 0.191(163.24e^{0.05\times 1} - 100) = 25.15$$

7.6.2 Implied Trinomial Trees

The implied binomial model can run into problems when one is matching some common volatility structures. In particular, negative transition probabilities can occur in several cases. One will then need to override the input data. However, the more data one overrides the less information from market prices will be reflected in the model.

A trinomial tree offers more flexibility and can match a larger class of volatility structures than its binomial counterpart. In an implied trinomial model, the state space can be chosen independently of the probabilities. The option prices are then used only to solve for the transition probabilities. We concentrate on constructing an implied trinomial tree using the method described by Derman, Kani, and Chriss (1996).

The transition probabilities above the center node of the tree are given by

$$p_{n,i} = \frac{e^{r\Delta T} c(S_{n,i}, T_{n+1}) - \sum_{j=i+1}^{2n} \lambda_{n,j}(F_{n,j} - S_{n+1,i+1})}{\lambda_{n,i}(S_{n+1,i+2} - S_{n+1,i+1})} \tag{7.23}$$

$$q_{n,i} = \frac{F_{n,i} - p_{n,i}(S_{n+1,i+2} - S_{n+1,i+1}) - S_{n+1,i+1}}{S_{n+1,i} - S_{n+1,i+1}} \tag{7.24}$$

The transition probabilities below and including the center node of the tree are given by

$$q_{n,i} = \frac{e^{r\Delta T} p(S_{n,i}, T_{n+1}) - \sum_{j=0}^{i-1} \lambda_{n,j}(S_{n+1,i+1} - F_{n,j})}{\lambda_{n,i}(S_{n+1,i+1} - S_{n+1,i})} \tag{7.25}$$

$$p_{n,i} = \frac{F_{n,i} + q_{n,i}(S_{n+1,i+1} - S_{n+1,i}) - S_{n+1,i+1}}{S_{n+1,i+2} - S_{n+1,i+1}} \tag{7.26}$$

In case of negative transition probabilities, one should try to choose another state space that better fits the volatility structure. Alternatively, one can override the negative probabilities. One way to ensure that the probabilities lie between zero and one is to set the probabilities equal to

If $S_{n+1,i+1} < F_{n,i} < S_{n+1,i+2}$

$$p_{n,i} = \frac{1}{2}\left(\frac{F_{n,i} - S_{n+1,i+1}}{S_{n+1,i+2} - S_{n+1,i+1}} + \frac{F_{n,i} - S_{n+1,i}}{S_{n+1,i+2} - S_{n+1,i}}\right)$$

$$q_{n,i} = \frac{1}{2}\left(\frac{S_{n+1,i+2} - F_{n,i}}{S_{n+1,i+2} - S_{n+1,i}}\right)$$

and if $S_{n+1,i} < F_{n,i} < S_{n+1,i+1}$, then

$$p_{n,i} = \frac{1}{2}\left(\frac{F_{n,i} - S_{n+1,i}}{S_{n+1,i+2} - S_{n+1,i}}\right)$$

$$q_{n,i} = \frac{1}{2}\left(\frac{S_{n+1,i+2} - F_{n,i}}{S_{n+1,i+2} - S_{n+1,i}} + \frac{S_{n+1,i+1} - F_{n,i}}{S_{n+1,i+1} - S_{n+1,i}}\right)$$

The transition probabilities at each time step can be used to calculate implied local volatilities for corresponding nodes in the tree.

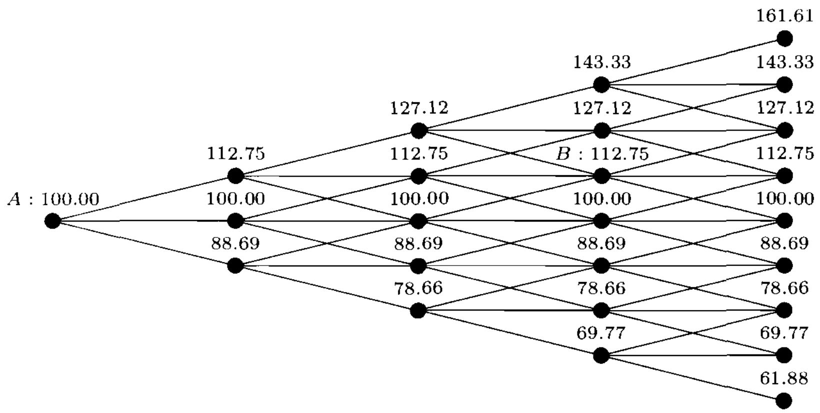
$$\sigma_{n,i} = \Big\{\big[p_{n,i}(S_{n+1,i+2} - F_0)^2 + (1 - p_{n,i} - q_{n,i})(S_{n+1,i+1} - F_0)^2 + q_{n,i}(S_{n+1,i} - F_0)^2\big]/(F_0^2 \Delta t)\Big\}^{1/2}, \tag{7.27}$$

where $F_0 = p_{n,i}S_{n+1,i+2} + (1 - p_{n,i} - q_{n,i})S_{n+1,i+1} + q_{n,i}S_{n+1,i}$.

Example

Assume the stock price is 100, the time to maturity is two years, the risk-free rate is 8%, the dividend yield is 6%, the at-the-money volatility is 12%. Assume moreover a volatility structure where the volatility increases (decreases) with 0.04% for every one-point decrease (increase) in the strike price. A strike price of 110 would then trade for 11.6% and a strike of 90 for 12.4% volatility. Build a two-year implied trinomial tree model with four time steps. $S = 100$, $T = 2$, $r = 0.08$, $b = 0.08 - 0.06 = 0.02$, $\sigma = 0.12$, $n = 4$, $\Delta t = 2/4 = 0.5$.

By choosing a trinomial equivalent CRR binomial tree ($u = e^{\sigma\sqrt{2\Delta t}}$ and $d = \frac{1}{u}$), we get the following stock price tree (state space):



By using equations (7.23), (7.24), (7.25), and (7.26), we can solve for the transition probabilities, the Arrow-Debreu prices, and the local volatilities at each node. To illustrate the calculation procedure, let's look at the calculation at node A:

$$q_{0,0} = \frac{e^{0.08\times 0.5}p(100, 0.5) - \sum_{j=0}^{i-1} \lambda_{n,j}(S_{n+1,i+1} - F_{n,j})}{1(100 - 88.69)} = 0.2240$$

The put price with strike 100, time to maturity six months, and volatility 12% is equal to 2.4333 in a standard trinomial tree with the same state space. The summation term $\sum_{j=0}^{i-1} \lambda_{n,j}(S_{n+1,i+1} - F_{n,j})$ is equal to zero, since there are no nodes below this node.

$$p_{0,0} = \frac{F_{n,i} + 0.2240(100.00 - 88.69) - 100.00}{112.75 - 100.00} = 0.2775$$

$F_{n,i} = F_{0,0} = 100e^{0.02\times 0.5} = 101.01$. The local volatility at this node is

$$\sigma_{0,0} = \Big\{[0.2775(112.75 - F_0)^2 + (1 - 0.2775 - 0.2240)(100.00 - F_0)^2$$
$$+ 0.2240(88.69 - F_0)^2]/(F_0^2 \times 0.5)\Big\}^{1/2} = 0.1194,$$

where

$$F_0 = (0.2775 \times 112.75 + 0.2240 \times 88.69$$
$$+ (1 - 0.2775 - 0.2240)100.00)e^{0.02\times 0.5} = 102.02$$

As a last example, let's look at the calculations at node B:

$$p_{3,4} = \frac{e^{0.08\times 0.5}c(112.75, 2) - \sum_{j=i+1}^{2n} \lambda_{n,j}(F_{n,j} - S_{n+1,i+1})}{0.2440(127.12 - 112.75)} = 0.2476$$

The price of a call option with strike 112.75, time to maturity two years, and volatility $0.12 + (100.00 - 112.75)0.0004 = 0.1149$ is equal to 2.8198. $\sum_{j=i+1}^{2n} \lambda_{n,j}(F_{n,j} - S_{n+1,i+1})$ is equal to

$$\sum_{j=4+1}^{2\times 3} = 0.1024(127.12e^{0.02\times 0.5} - 112.75)$$
$$+ 0.0145(143.33e^{0.02\times 0.5} - 112.75) = 2.07,$$

while the down probability equals

$$q_{3,4} = \frac{F_{n,i} - 0.2476(127.12 - 112.75) - 112.75}{100.00 - 112.75} = 0.1903, \tag{7.28}$$

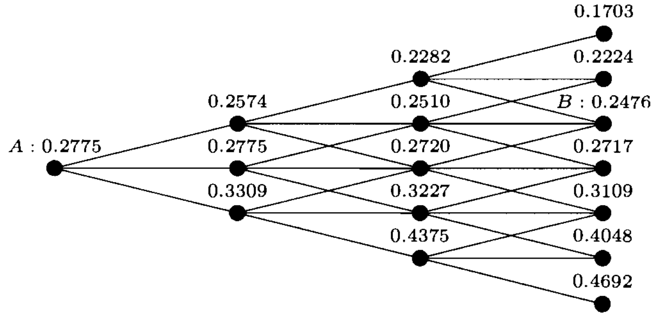
and $F_{n,i} = F_{3,4} = 112.75^{0.02\times 0.5} = 113.88$. The local volatility is

$$\sigma_{n,i} = \Big\{[0.2476(127.12 - F_0)^2 + (1 - 0.2476 - 0.1903)(112.75 - F_0)^2$$
$$+ 0.1903(100.00 - F_0)^2]/(F_0^2 \times 0.5)\Big\} = 0.1116,$$

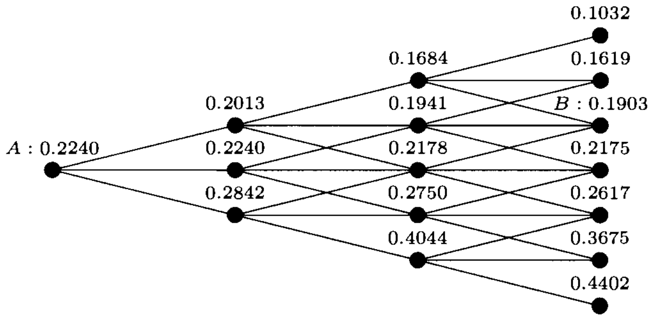
where

$$F_0 = (0.2476 \times 127.12 + 0.1903 \times 100.00 + (1 - 0.2476 - 0.1903)112.75)e^{0.02\times 0.5} = 115.03$$

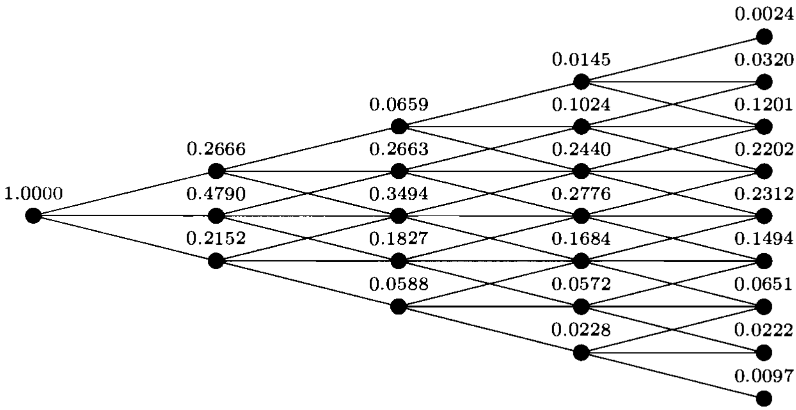
Up-Transition Probabilities



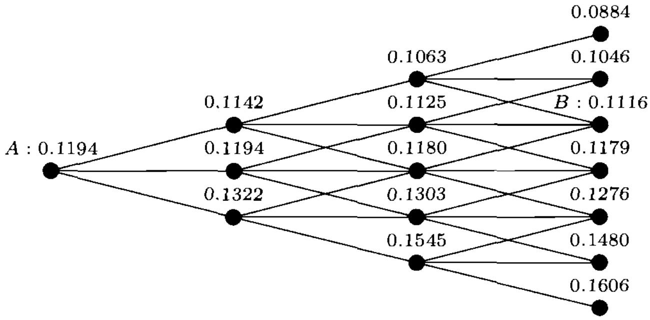
Down-Transition Probabilities



Arrow-Debreu Price Tree



Implied Local Volatilities



Computer algorithm

The computer code can be used to build a complete implied trinomial tree. For simplicity, the volatility smile is set to be a linear function of the strike price. The input parameter *Skew* will decide the steepness of the volatility smile/skew. For instance, setting the *Skew* equal to 0.0004 will give a volatility smile/skew as in the numerical example above. In practice, the volatility function should naturally be adjusted to reflect the real implied volatilities. By specifying the "*ReturnFlag*," the code will return

"**UPM**" A matrix of implied up transition probabilities.

"**DPM**" A matrix of implied down transition probabilities.

"**LVM**" A matrix of implied local volatilities.

"**ADM**" A matrix of Arrow-Debreu prices at a single node.

"**DPni**" The implied down transition probability at a single node.

"**ADni**" The Arrow-Debreu price at a single node (at time step *STEPn* and state *STATEn*).

"**LVni**" The local volatility at a single node.

"**c**" The value of a European call option.

"**p**" The value of a European put option.

The code calls the *TrinomialTree*(·) function to calculate option prices used to calibrate the implied trinomial tree. The chosen state space in the computer code is the trinomial equivalent CRR binomial state space. However, the state space could easily be changed by modifying the code slightly.

```
Function ImpliedTrinomialTree(ReturnFlag As String, STEPn As Integer, _
    STATEi As Integer, S As Double, X As Double, T As Double, _
                r As Double, b As Double, v As Double, _
        Skew As Double, nSteps As Integer)

    Dim ArrowDebreu() As Double
    Dim LocalVolatility() As Double
    Dim UpProbability() As Double
    Dim DownProbability() As Double
    Dim OptionValueNode() As Double
    Dim dt As Double, u As Double, d As Double
    Dim Df As Double, pi As Double, qi As Double
    Dim Si1 As Double, Si As Double, Si2 As Double
```

```
Dim vi As Double, Fj As Double, Fi As Double, Fo As Double
Dim sum As Double, OptionValue As Double
Dim i As Integer, j As Integer, n As Integer, z As Integer

ReDim OptionValueNode(0 To nSteps * 2) As Double
ReDim ArrowDebreu(0 To nSteps, 0 To nSteps * 2) As Double
ReDim UpProbability(0 To nSteps - 1, 0 To nSteps * 2 - 2) As Double
ReDim DownProbability(0 To nSteps - 1, 0 To nSteps * 2 - 2) As Double
ReDim LocalVolatility(0 To nSteps - 1, 0 To nSteps * 2 - 2) As Double

dt = T / nSteps
u = Exp(v * Sqr(2 * dt))
d = 1 / u
Df = Exp(-r * dt)
ArrowDebreu(0, 0) = 1
For n = 0 To nSteps - 1
    For i = 0 To n * 2
        sum = 0
        Si1 = S * u^Max(i - n, 0) * d^Max(n * 2 - n - i, 0)
        Si = Si1 * d
        Si2 = Si1 * u
        Fi = Si1 * Exp(b * dt)
        vi = v + (S - Si1) * Skew
        If i < (n * 2) / 2 + 1 Then
            For j = 0 To i - 1
                Fj = S * u^Max(j - n, 0) * d^Max(n * 2 - n - j, 0) _
                    * Exp(b * dt)
                sum = sum + ArrowDebreu(n, j) * (Si1 - Fj)
            Next
            OptionValue = TrinomialTree("e", "p", S, Si1, (n + 1) _
              * dt, r, b, vi, n + 1)
            qi = (Exp(r * dt) * OptionValue - sum) _
              / (ArrowDebreu(n, i) * (Si1 - Si))
            pi = (Fi + qi * (Si1 - Si) - Si1) / (Si2 - Si1)
        Else
            OptionValue = TrinomialTree("e", "c", S, Si1, (n + 1) _
              * dt, r, b, vi, n + 1)
            sum = 0
            For j = i + 1 To n * 2
                Fj = S * u^Max(j - n, 0) _
                  * d^Max(n * 2 - n - j, 0) * Exp(b * dt)
                sum = sum + ArrowDebreu(n, j) * (Fj - Si1)
            Next
            pi = (Exp(r * dt) * OptionValue - sum) _
              / (ArrowDebreu(n, i) * (Si2 - Si1))
            qi = (Fi - pi * (Si2 - Si1) - Si1) / (Si - Si1)
        End If

        '// Replacing negative probabilities
        If pi < 0 Or pi > 1 Or qi < 0 Or qi > 1 Then
            If Fi > Si1 And Fi < Si2 Then
                pi = 1 / 2 * ((Fi - Si1) / (Si2 - Si1) _
                  + (Fi - Si) / (Si2 - Si))
                qi = 1 / 2 * ((Si2 - Fi) / (Si2 - Si))
            ElseIf Fi > Si And Fi < Si1 Then
                pi = 1 / 2 * ((Fi - Si) / (Si2 - Si))
                qi = 1 / 2 * ((Si2 - Fi) / (Si2 - Si) _
                  + (Si1 - Fi) / (Si1 - Si))
            End If
```

```
        End If
            DownProbability(n, i) = qi
            UpProbability(n, i) = pi
        '// Calculating local volatilities
            Fo = (pi * Si2 + qi * Si + (1 - pi - qi) * Si1)
            LocalVolatility(n, i) = Sqr((pi * (Si2 - Fo)^2 _
             + (1 - pi - qi) * (Si1 - Fo)^2 + qi _
              * (Si - Fo)^2) / (Fo^2 * dt))

        '// Calculating Arrow-Debreu prices
        If n = 0 Then
            ArrowDebreu(n + 1, i) = qi * ArrowDebreu(n, i) * Df
            ArrowDebreu(n + 1, i + 1) = (1 - pi - qi) _
              * ArrowDebreu(n, i) * Df
            ArrowDebreu(n + 1, i + 2) = pi * ArrowDebreu(n, i) * Df
        ElseIf n > 0 And i = 0 Then
            ArrowDebreu(n + 1, i) = qi * ArrowDebreu(n, i) * Df
        ElseIf n > 0 And i = n * 2 Then
            ArrowDebreu(n + 1, i) = UpProbability(n, i - 2) _
            * ArrowDebreu(n, i - 2) * Df + (1 - UpProbability(n, i - 1) _
            - DownProbability(n, i - 1)) * ArrowDebreu(n, i - 1) * Df _
            + qi * ArrowDebreu(n, i) * Df
            ArrowDebreu(n + 1, i + 1) = UpProbability(n, i - 1) _
              * ArrowDebreu(n, i - 1) * Df + (1 - pi - qi) _
              * ArrowDebreu(n, i) * Df
            ArrowDebreu(n + 1, i + 2) = pi * ArrowDebreu(n, i) * Df
        ElseIf n > 0 And i = 1 Then
            ArrowDebreu(n + 1, i) = (1 - UpProbability(n, i - 1) _
    - DownProbability(n, i - 1)) * ArrowDebreu(n, i - 1) * Df _
              + qi * ArrowDebreu(n, i) * Df
        Else
        ArrowDebreu(n + 1, i) = UpProbability(n, i - 2) _
          * ArrowDebreu(n, i - 2) * Df _
           + (1 - UpProbability(n, i - 1) - DownProbability(n, i - 1)) _
     * ArrowDebreu(n, i - 1) * Df + qi * ArrowDebreu(n, i) * Df
    End If
    Next
Next

If ReturnFlag = "DPM" Then
    ImpliedTrinomialTree = Application.Transpose(DownProbability)
ElseIf ReturnFlag = "UPM" Then
    ImpliedTrinomialTree = Application.Transpose(UpProbability)
ElseIf ReturnFlag = "DPni" Then
    ImpliedTrinomialTree = (DownProbability(STEPn, STATEi))
ElseIf ReturnFlag = "UPni" Then
  ImpliedTrinomialTree = (UpProbability(STEPn, STATEi))
ElseIf ReturnFlag = "ADM" Then
    ImpliedTrinomialTree = Application.Transpose(ArrowDebreu)
ElseIf ReturnFlag = "LVM" Then
    ImpliedTrinomialTree = Application.Transpose(LocalVolatility)
ElseIf ReturnFlag = "LVni" Then
    ImpliedTrinomialTree = Application.Transpose _
    (LocalVolatility(STEPn, STATEi))
ElseIf ReturnFlag = "ADni" Then
    ImpliedTrinomialTree = (ArrowDebreu(STEPn, STATEi))
Else

'// Calculation of option price using the implied trinomial tree
```

```
        If ReturnFlag = "c" Then
            z = 1
        ElseIf ReturnFlag = "p" Then
            z = -1
        End If
        For i = 0 To (2 * nSteps)
                OptionValueNode(i) = Max(0, z * (S * u^Max(i - nSteps, 0) _
                * d^Max(nSteps - i, 0) - X))
        Next
         For n = nSteps - 1 To 0 Step -1
            For i = 0 To (n * 2)
                OptionValueNode(i) = (UpProbability(n, i) _
                * OptionValueNode(i + 2) + (1 - UpProbability(n, i) _
        - DownProbability(n, i)) * OptionValueNode(i + 1) _
        + DownProbability(n, i) * OptionValueNode(i)) * Df
            Next
        Next
        ImpliedTrinomialTree = OptionValueNode(0)
    End If

End Function
```

Smile Dynamics

The implied tree models offer useful intuition on how local and global volatilities are connected. The implied tree models mentioned so far all assume the local volatility is a deterministic function of time and the asset price. In practice, the volatility is typically partly deterministic and partly stochastic. One natural extension is therefore stochastic implied trees, where the volatility is stochastic. For more on this extension see Derman and Kani (1997). Another factor one needs to consider before using a volatility model in practice is the dynamics of the smile. The literature distinguishes between two main types of models, so called sticky strike and sticky delta. In a sticky-strike model the volatility for a given strike will stay the same as the asset price moves, while in a sticky-delta model the volatility for a given delta will stay the same. Which of the two models is best varies, see Derman (1999) for more details.

7.7 FINITE DIFFERENCE METHODS

The use of finite difference methods in finance was first described by Brennan and Schwartz (1978). Finite difference models, also called grid models, are simply a numerical technique to solve partial differential equations (PDE). Different finite difference methods can be used to price European and American options, as well as many types of exotic options.

Finite difference models are as we soon will see very similar to tree models. The finite difference method is basically a numerical approximation of the PDE. Here we will give an overview of the three most common finite difference techniques in option pricing:

- Explicit finite difference
- Implicit finite difference
- Crank-Nicolson finite difference

All the finite difference models described here build on the same main principle. First we build a grid with time along one dimension/axis and price along the other dimension/axis. Just as in a tree model, the time and price movements are discretized. Time increases in increments of Δt, while the asset changes in amounts of ΔS. These increments are then used to construct a grid of possible combinations of time and asset price levels. The finite difference technique is then used to approximately solve the relevant PDE on this grid. Just as in a tree model one starts at the "end" of the grid, at time T, and rolls back through the grid. The calculations done on the grid are a bit different, however.

The finite difference models can be used to solve a large class of PDEs, and thereby a large class of options. If we assume that the underlying asset follows a geometric Brownian motion, we get the following Black-Scholes-Merton PDE for any single asset derivatives (see Appendix A in Chapter 1 for how to come up with this PDE):

$$\frac{\partial f}{\partial t} + \frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2 S^2 + b\frac{\partial f}{\partial S}S = rf, \tag{7.29}$$

where f is the value of a derivative security for example a European call c or an American call C or some type of exotic option. We want to solve this PDE along the grid for the particular derivative instrument under consideration. How this is done depends on the chosen finite difference technique, as well as the derivative's contractual details. We start with the explicit finite difference method. Wilmott (2000), Randall and Tavella (2000), Topper (2005) and Duffy (2006) offer more detailed coverage of finite difference methods applied to option pricing.

7.7.1 Explicit Finite Difference

The explicit finite difference method is more or less a generalization of the trinomial tree. The method approximates the PDE in equation (7.29) by using numerical differentiation, Hull and White (1990). The $\frac{\partial f}{\partial t}$ is approximated by using the forward difference (this naturally because time can only move forward):

$$\frac{\partial f}{\partial t} \approx \frac{f_{j+1,i} - f_{j,i}}{\Delta t},$$

where $f_{j,i}$ is the value of the derivative instrument at time step j and price level i. The delta, $\frac{\partial f}{\partial S}$, and the gamma, $\frac{\partial^2 f}{\partial S^2}$, are approximated by central differences (the asset price can naturally move in both directions):

$$\frac{\partial f}{\partial S} \approx \frac{f_{j+1,i+1} - f_{j+1,i-1}}{2\Delta S} \qquad \frac{\partial^2 f}{\partial S^2} \approx \frac{f_{j+1,i+1} - 2f_{j+1,i} + f_{j+1,i-1}}{\Delta S^2}$$

Replacing the partial derivatives in equation (7.29) with these approximations, we get

$$\frac{f_{j+1,i} - f_{j,i}}{\Delta t} + \frac{1}{2}\frac{f_{j+1,i+1} - 2f_{j+1,i} + f_{j+1,i-1}}{\Delta S^2}\sigma^2 S^2 + b\frac{f_{j+1,i+1} - f_{j+1,i-1}}{2\Delta S}S = rf_{j,i},$$

which can be rewritten as

$$f_{j,i} = \frac{1}{1 + r\Delta t}(p_u f_{j+1,i+1} + p_m f_{j+1,i} + p_d f_{j+1,i-1}),$$

where

$$p_u = \frac{1}{2}(\sigma^2 i^2 + bi)\Delta t$$

$$p_m = 1 - \sigma^2 i^2 \Delta t$$

$$p_d = \frac{1}{2}(\sigma^2 i^2 - bi)\Delta t$$

This is very similar to the trinomial tree backward equations, and it can be shown that both a trinomial and also binomial tree actually is a special case of a finite difference method. See, for example, Heston and Zhou (2000) and James (2003).

Explicit Finite Difference Implementation

First we have to decide on the price step and time step sizes. A good size of price and time steps seems to be $\Delta S = \frac{2X}{n_j}$ and $\Delta t = \frac{\Delta S^2}{\sigma^2 4X^2}$, where n_j is chosen by the user. The number of time steps is then set to $N = Integer(\frac{T}{\Delta t}) + 1$.

The current asset price will often fall between the nodes of the finite difference grid. To get accurate option values, we need to adjust for this. One way to do this is simply to interpolate the current value from the two closest grid points (Wilmott, 2000). Let us define the nearest grid point below the current asset price as $z = integer(\frac{S}{\Delta S})$. After rolling back through the grid, we can find the current option value as a weighted sum (interpolation) of the two closest grid points, where the weight given to the nearest grid points, respectively, just above, and below the current asset price is

$$w = \frac{S - z\Delta S}{\Delta S}$$

The current option value at time zero is therefore

$$c = (1 - w)c(0, z) + wc(0, z + 1)$$

Alternatively, using the interpolation method just described, we can ensure that a grid point always falls exactly on the current asset price.[8] This can easily be done by, for example, setting $\Delta S = \frac{S}{n_j}$, where n_j is given by the user, and after this, setting $n_j = integer(\frac{X}{\Delta S})2$.

Computer algorithm

The computer code returns the value of a European or American call or put option using the explicit finite difference method. Setting the *AmeEurFlag* = "*a*" gives American option values, *AmeEurFlag* = "*e*" gives European values. Similarly, setting the *CallPutFlag* = "*c*" returns a call value, and *CallPutFlag* = "*p*" returns a put value. In the computer code, $v = \sigma$ and $dt = \Delta t$. In this implementation we make sure a grid point corresponds to the current asset price.

```
Function ExplicitFiniteDifference(AmeEurFlag As String, _
  CallPutFlag As String, S As Double, X As Double, T As Double, _
  r As Double, b As Double, v As Double, M As Integer) As Double

    Dim C() As Double, St() As Double
    Dim dt As Double, dS As Double
    Dim pu As Double, pm As Double, pd As Double, Df As Double
    Dim i As Integer, j As Integer, N As Integer, z As Integer
    Dim SGridPt As Integer

    z = 1
    If CallPutFlag = "p" Then z = -1

    dS = S / M
    M = Int(X / dS) * 2
    ReDim St(0 To M + 1)

    SGridPt = S / dS
    dt = dS^2 / (v^2 * 4 * X^2)
    N = Int(T / dt) + 1

    ReDim C(0 To N, 0 To M + 1)
    dt = T / N
    Df = 1 / (1 + r * dt)
```

[8]Thanks to Sam at the Wilmott forum for pointing this out to me, www.wilmott.com.

```
For i = 0 To M
      St(i) = i * dS ' // Asset price at maturity
      C(N, i) = Max(0, z * (St(i) - X)) '// At maturity
Next
For j = N - 1 To 0 Step -1
     For i = 1 To M - 1
     pu = 0.5 * (v^2 * i^2 + b * i) * dt
     pm = 1 - v^2 * i^2 * dt
     pd = 0.5 * (v^2 * i^2 - b * i) * dt
     C(j, i) = Df * (pu * C(j + 1, i + 1) _
        + pm * C(j + 1, i) + pd * C(j + 1, i - 1))
     If AmeEurFlag = "a" Then
         C(j, i) = Max(z * (St(i) - X), C(j, i))
     End If
     Next
     If z = 1 Then '//Call option
         C(j, 0) = 0
         C(j, M) = (St(i) - X)
     Else
         C(j, 0) = X
         C(j, M) = 0
     End If
Next
ExplicitFiniteDifference = C(0, SGridPt)

End Function
```

7.7.2 Implicit Finite Difference

The implicit finite difference method is closely related to the explicit finite difference method. The main difference is that we approximate $\frac{\partial f}{\partial S}$ and $\frac{\partial^2 f}{\partial S^2}$ in PDE (7.29) by central differentiation at time step j instead at $j+1$ as in the explicit finite difference method. This gives

$$\frac{f_{j+1,i} - f_{j,i}}{\Delta t} + \frac{1}{2}\frac{f_{j,i+1} - 2f_{j,i}f_{j,i-1}}{\Delta S^2}\sigma^2 S + b\frac{f_{j,i+1} - f_{j,i-1}}{2\Delta S}S = rf_{j,i},$$

which can be rewritten as

$$f_{j+1,i} = p_u f_{j,i+1} + p_m f_{j,i} + p_d f_{j,i-1}$$
$$p_u = \frac{1}{2}i(b + -v^2 i)\Delta t$$
$$p_m = 1 + (r + v^2 i^2)\Delta t$$
$$p_d = \frac{1}{2}i(-b - v^2 i)\Delta t$$

If we use M as the number of price steps in the grid, we now need to solve for $M-1$ unknown derivatives values, $f_{j,i}$, on the grid simultaneously:

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ p_u(0), & p_m(1) & p_d(2) & \cdots & \cdots & \cdots & 0 \\ 0 & p_u(1) & p_m(2) & p_d(3) & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & p_u(3) & p_m(4) & p_d(5) & 0 \\ 0 & 0 & 0 & 0 & p_u(4) & p_m(6) & p_d(7) \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_{j,M} \\ f_{j,M-1} \\ f_{j,M-2} \\ \cdots \\ f_{j,M-5} \\ f_{j,M-6} \\ f_{j,M-7} \end{bmatrix} = \begin{bmatrix} f_{j+1,M} \\ f_{j,M-1} \\ f_{j,M-2} \\ \cdots \\ f_{j+1,M-5} \\ f_{j+1,M-6} \\ f_{j+1,M-7} \end{bmatrix}$$

We can now easily find the option value by inverting the tridiagonal matrix. This gives us the European option values. To find the value of an American option, we now need to roll back through the grid and check if the exercise value at each grid point is higher than the already calculated grid option values.

Computer algorithm

The computer code returns the value of a European call or put option using the implicit finite difference method. Setting the *CallPutFlag* = "*c*" returns a call value, and *CallPutFlag* = "*p*" returns a put value. In the computer code $v = \sigma$ and $dt = \Delta t$. M is the number of price steps used and N is the number of time steps.

```
Function ImplicitFiniteDifference(AmeEurFlag As String, _
  CallPutFlag As String, S As Double, X As Double, _
  T As Double, r As Double, b As Double, v As Double, _
  N As Integer, M As Integer) As Double

    Dim p() As Variant, CT() As Variant, C As Variant
    Dim dS As Double, dt As Double
    Dim i As Integer, z As Integer, j As Integer
    Dim SGridPt As Integer

    z = 1
    If CallPutFlag = "p" Then z = -1

    '// Makes sure current asset price falls at grid point
    dS = 2 * S / M
    SGridPt = S / dS
    M = Int(X / dS) * 2
    dt = T / N

    ReDim CT(0 To M)
    ReDim p(0 To M, 0 To M)

    For j = 0 To M
        CT(j) = Max(0, z * (j * dS - X)) '// At maturity
        For i = 0 To M
            p(j, i) = 0
        Next
    Next
```

```
    p(0, 0) = 1
    For i = 1 To M - 1 Step 1
        p(i, i - 1) = 0.5 * i * (b - v^2 * i) * dt
        p(i, i) = 1 + (r + v^2 * i^2) * dt
        p(i, i + 1) = 0.5 * i * (-b - v^2 * i) * dt
    Next

    p(M, M) = 1
   C = Application.MMult(Application.MInverse(p()), _
      Application.Transpose(CT()))
   For j = N - 1 To 1 Step -1
       C = Application.MMult(Application.MInverse(p()), C)

       If AmeEurFlag = "a" Then '//American option
           For i = 1 To M
               C(i, 1) = Max(CDbl(C(i, 1)), z * ((i - 1) * dS - X))
            Next
       End If
    Next

    ImplicitFiniteDifference = C(SGridPt + 1, 1)

End Function
```

7.7.3 Finite Difference in ln(S)

Alternatively, we can rewrite the Black-Scholes-Merton PDE in terms of ln(S). Letting $x = \ln(S)$, we get

$$\frac{\partial f}{\partial t} + \frac{1}{2}\sigma^2\frac{\partial^2 f}{\partial x^2} + (b - \sigma^2/2)\frac{\partial f}{\partial x} = rf \tag{7.30}$$

This will, in general, make the finite difference implementation slightly more efficient, as the coefficients now no longer depend on the price step in the grid.

Explicit Finite Difference Method

For the explicit finite difference method, p_u, p_m and p_d now become independent of the price step, i, which makes the method somewhat more efficient. In the case of the explicit finite difference method, we now have

$$p_u = \frac{1}{2}\Delta t\left(\frac{\sigma^2}{\Delta x^2} + \frac{b - \sigma^2/2}{\Delta x}\right)$$

$$p_m = 1 - \Delta t\frac{\sigma^2}{\Delta x^2} - r\Delta t$$

$$p_d = \frac{1}{2}\Delta t\left(\frac{\sigma^2}{\Delta x^2} - \frac{b - \sigma^2/2}{\Delta x}\right)$$

It is numerically most efficient to set the price step $\Delta x = \sigma\sqrt{3\Delta t}$.

It is worth mentioning that even if rewriting the PDE in this way makes the implementation slightly more efficient, the drawback is that in practice, different conditions typically depend on S and not $\ln(S)$—for example, barrier options and reset-strike options. This can make the implementation less intuitive for some derivatives.

Computer algorithm

The computer code returns the value of a European or American call or put option using the explicit finite difference method for a PDE in $\ln(S)$. Setting the *AmeEurFlag* = "*a*" gives American option values, *AmeEurFlag* = "*e*" gives European values. Similarly, setting the *CallPutFlag* = "*c*" returns a call value, and *CallPutFlag* = "*p*" returns a put value. In the computer code, $v = \sigma$ and $dt = \Delta t$. N is the number of time steps and M is the number of price steps.

```
Function ExplicitFiniteDifferenceLnS(AmeEurFlag As String, _
  CallPutFlag As String, S As Double, X As Double, _
  T As Double, r As Double, b As Double, v As Double, _
  N As Integer, M As Integer) As Double

    Dim C() As Double, St() As Double
    Dim dt As Double, dx As Double
    Dim pu As Double, pm As Double, pd As Double
    Dim i As Integer, j As Integer, z As Integer

    ReDim C(0 To M / 2, 0 To M + 1)
    ReDim St(0 To M + 1)

    z = 1
    If CallPutFlag = "p" Then z = -1

    dt = T / N
    dx = v * Sqr(3 * dt)
    pu = 0.5 * dt * ((v / dx)^2 + (b - v^2 / 2) / dx)
    pm = 1 - dt * (v / dx)^2 - r * dt
    pd = 0.5 * dt * ((v / dx)^2 - (b - v^2 / 2) / dx)

    St(0) = S * Exp(-M / 2 * dx)
    C(N, 0) = Max(0, z * (St(0) - X))

    For i = 1 To M
        St(i) = St(i - 1) * Exp(dx)    ' // asset price at maturity
        C(N, i) = Max(0, z * (St(i) - X)) '// Option value at maturity
    Next

    For j = N - 1 To 0 Step -1
        For i = 1 To M - 1
            C(j, i) = pu * C(j + 1, i + 1) _
        + pm * C(j + 1, i) + pd * C(j + 1, i - 1)
            If AmeEurFlag = "a" Then '//American option
                C(j, i) = Max(C(j, i), z * (St(i) - X))
            End If
        Next
        C(j, M) = C(j, M - 1) + St(M) - St(M - 1) '//Upper boundary
        C(j, 0) = C(j, 1) '// Lower boundary
    Next
```

```
    ExplicitFiniteDifferenceLnS = C(0, M / 2)
End Function
```

7.7.4 The Crank-Nicolson Method

In the Crank-Nicolson method the approximation of the PDE equation is done by central differences at time step $j + \frac{1}{2}$ instead of at $j + 1$ as in the explicit finite difference method, or at point j as in the implicit finite difference method. Using the Crank-Nicolson method to approximate PDE (7.30) gives

$$
\begin{aligned}
-\frac{f_{j+1,i} - f_{j,i}}{\Delta t} = \frac{1}{2}\sigma^2 &\frac{(f_{j,i+1} - 2f_{j,i} f_{j,i-1}) + (f_{j+1,i+1} - 2f_{j+1,i} f_{j+1,i-1})}{2\Delta x^2} \\
&+ (b - \sigma^2/2)\frac{(f_{j+1,i+1} - f_{j+1,i-1}) + (f_{j,i+1} - f_{j,i-1})}{4\Delta x} \\
&- r\left(\frac{f_{j+1,i} + f_{j,i}}{2}\right)
\end{aligned}
$$

As we can see, the Crank-Nicolson method is a combination of the explicit and implicit methods. It is more efficient than the others. In combination with the same boundary conditions as in the implicit finite difference method, the Crank-Nicolson method will make up a tridiagonal system of equations. For an in-depth discussion of the Crank-Nicolson method applied to derivatives valuation, see Wilmott (2000).

Computer algorithm

```
Function CrankNicolson(AmeEurFlag As String, CallPutFlag As String, _
  S As Double, X As Double, T As Double, r As Double, _
  b As Double, v As Double, N As Integer, M As Integer) As Double


    Dim C() As Double, St() As Double, p() As Double, pmd() As Double
    Dim dt As Double, dx As Double
    Dim i As Integer, j As Integer, z As Integer
    Dim pu As Double, pm As Double, pd As Double

    ReDim pmd(0 To M)
    ReDim p(0 To M)
    ReDim C(0 To M / 2 + 1, 0 To M + 1)
    ReDim St(0 To M + 1)

    z = 1
    If CallPutFlag = "p" Then z = -1

    dt = T / N
    dx = v * Sqr(3 * dt)
    pu = -0.25 * dt * ((v / dx)^2 + (b - 0.5 * v^2) / dx)
```

```
    pm = 1 + 0.5 * dt * (v / dx)^2 + 0.5 * r * dt
    pd = -0.25 * dt * ((v / dx)^2 - (b - 0.5 * v^2) / dx)

    St(0) = S * Exp(-M / 2 * dx)
    C(0, 0) = Max(0, z * (St(0) - X))
    For i = 1 To M '// Option value at maturity
        St(i) = St(i - 1) * Exp(dx)
        C(0, i) = Max(0, z * (St(i) - X))
    Next

    pmd(1) = pm + pd
    p(1) = -pu * C(0, 2) - (pm - 2) * C(0, 1) _
    - pd * C(0, 0) - pd * (St(1) - St(0))
    For j = N - 1 To 0 Step -1

        For i = 2 To M - 1
            p(i) = -pu * C(0, i + 1) - (pm - 2) * C(0, i) _
    - pd * C(0, i - 1) - p(i - 1) * pd / pmd(i - 1)
            pmd(i) = pm - pu * pd / pmd(i - 1)
        Next


        For i = M - 2 To 1 Step -1
            C(1, i) = (p(i) - pu * C(1, i + 1)) / pmd(i)
        Next

            For i = 0 To M
                C(0, i) = C(1, i)
                If AmeEurFlag = "a" Then
                    C(0, i) = Max(C(1, i), z * (St(i) - X))
                 End If
            Next

    Next

    CrankNicolson = C(0, M / 2)

End Function
```

CHAPTER

8

MONTE CARLO SIMULATION

If you are out to describe the truth, leave elegance to the tailor.

Albert Einstein

8.1 STANDARD MONTE CARLO SIMULATION

Monte Carlo simulation is a numerical method that is useful in many situations when no closed-form solution is available. Monte Carlo simulating in option pricing, originally introduced by Boyle (1977), can be used to value most types of European options and, as we will see, also American options. The Monte Carlo method can be used to simulate a wide range of stochastic processes and is thus very general. To illustrate the use of Monte Carlo simulation, we will start with the processes where the natural logarithm of the underlying asset follows geometric Brownian motion. The process governing the asset price S is then given by

$$S + dS = S \exp\left[\left(\mu - \frac{1}{2}\sigma^2\right) dt + \sigma dz\right],$$

where dz is a Wiener process with standard deviation 1 and mean 0. To simulate the process, we consider its values at given discrete time intervals, Δt apart:

$$S + \Delta S = S \exp\left[\left(\mu - \frac{1}{2}\sigma^2\right) \Delta t + \sigma \epsilon_t \sqrt{\Delta t}\right],$$

where ΔS is the change in S in the chosen time interval Δt, and ϵ_t is a random drawing from a standard normal distribution.

The main drawback of Monte Carlo simulation is that it is computer-intensive. A minimum of 10,000 simulations are typically necessary to price an option with satisfactory accuracy. The standard error in the estimated value from the standard Monte Carlo simulation is normally related to the square root of the number of simulations. More precisely, if s is the standard deviation of the payoffs from n simulations, then the standard error is given by $\frac{s}{\sqrt{n}}$. This means that to double the accuracy, we will need to quadruple the number of simulations. So if we want to double the accuracy from 10,000 simulations, we will need 40,000 simulations, and so on. Several techniques are available to speed up the Monte Carlo simulation, something we will look into later in this chapter (variance reduction techniques). The computer code in this book is written in Visual Basic (VBA). Implementing the same code in C++ (or any other lower-level computer language) will increase the speed dramatically and is recommended for serious, large-scale use.

The steps in standard Monte Carlo simulation is simply to simulate n number of paths of the asset price. The value is then given as the discounted average of the simulated paths. In the case of non-path-dependent European style options, we are only interested in the end value of the asset, and the value of a standard call option is simply given by

$$c = \frac{e^{-rT}}{n} \sum_{i=1}^{n} \max[Se^{(b-\sigma^2/2)T+\sigma\epsilon_i\sqrt{T}} - X, 0],$$

where ϵ_i is a random drawing from a standard normal distribution, and similarly for a put option we have

$$p = \frac{e^{-rT}}{n} \sum_{i=1}^{n} \max[X - Se^{(b-\sigma^2/2)T+\sigma\epsilon_i\sqrt{T}}, 0]$$

For path-dependent options, we also need to divide each path into discrete time steps. Doing this for a standard European option would be unnecessarily computer-intensive, but just so you get an idea, here is what it would look like:

$$c = \frac{e^{-rT}}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} \max[S_{j-1}e^{(b-\sigma^2/2)\Delta t+\sigma\epsilon_{j,i}\sqrt{\Delta t}} - X, 0],$$

where m is the number of time steps, $\Delta t = T/m$, and S_0 is the initial asset price when $j = 1$. Also, when it comes to stochastic processes like mean reversion, we have to divide the path into discrete time steps even for standard European options, as we soon will look into.

Computer algorithm

This function can be used to price standard European call and put options.

```
Function MonteCarloStandardOption(CallPutFlag As String, _
    S As Double, X As Double, T As Double, r As Double, b As Double, _
    v As Double, nSimulations As Long) As Double

    Dim St As Double
    Dim sum As Double, Drift As Double, vSqrdt As Double
    Dim i As Long, z As Integer
```

```
    Drift = (b - v^2 / 2) * T
    vSqrdt = v * Sqr(T)

    If CallPutFlag = "c" Then
        z = 1
    ElseIf CallPutFlag = "p" Then
        z = -1
    End If

    For i = 1 To nSimulations
            St = S * Exp(Drift + vSqrdt * Application.NormInv(Rnd(), 0, 1))
        sum = sum + Max(z * (St - X), 0)
    Next

    MonteCarloStandardOption = Exp(-r * T) * sum / nSimulations

End Function
```

8.1.1 Greeks in Monte Carlo

One way to calculate Greeks that works for any model or implementation is to use finite difference approximations—described as numerical Greeks in Chapter 2. For example, to calculate the delta with two-sided finite difference method:

$$\Delta \approx \frac{c(S+\Delta S, X, T, r, b, \sigma) - c(S-\Delta S, X, T, r, b, \sigma)}{2\Delta S}$$

This involves running the Monte Carlo simulation twice, once using $S+\Delta S$ as initial asset price and once using $S-\Delta S$ as initial asset price. Vega, theta, rho, gamma, and any other Greek can be computed in similarly ways.

Finite difference approximations are unnecessary computer-intensive and not very accurate in combination with Monte Carlo methods. Curran (1993) suggests a more efficient and accurate way to calculate Greeks using Monte Carlo simulation for many derivatives. Let us for illustration purposes assume geometric Brownian motion. In the case of delta, the method starts by counting the number of paths that ends up in-the-money. For these paths we add up the asset prices at maturity, then divide this sum by the number of simulations times the initial asset price—that is:

$$\Delta = e^{-rT}\frac{\sum}{Sn}, \tag{8.1}$$

where $\sum$ is the sum of the asset prices at the end of each path for every in-the-money path and n is the number of simulations. In the case of gamma, we have

$$\Gamma = e^{-rT}\left(\frac{X}{S}\right)^2\frac{\sum}{4n}, \tag{8.2}$$

where $\sum$ now is the number of paths where $|S_T - X| < e$ for some e, and n is the number of simulations. Now that we know delta and gamma, we can find theta from the following relationship:

$$\Theta = rC - bS\Delta + \frac{1}{2}\sigma^2 S^2 \Gamma \tag{8.3}$$

To get the theta on the notation of a one-day time decay, we have to divide the calculation above by the number of days in the year, 365.

To calculate vega, we use the following relationship:

$$\text{Vega} = \Gamma \sigma S^2 T \tag{8.4}$$

To get the vega in terms of one volatility point we need to divide by 100.

Computer algorithm

This function can be used to price standard European call and put options with delta, gamma, theta, and vega. The function returns an array of these measures.

```
Function StandardMCWithGreeks(OutputFlag As String, _
 CallPutFlag As String, S As Double, X As Double, _
 T As Double, r As Double, b As Double, v As Double, _
 nSimulations As Long) As Variant

    Dim St As Double, Output() As Double
    Dim sum As Double, Drift As Double, vSqrdt As Double
    Dim DeltaSum As Double, GammaSum As Double
    Dim i As Long, z As Integer

   ReDim Output(0 To 4) As Double

    Drift = (b - v^2 / 2) * T
    vSqrdt = v * Sqr(T)

    If CallPutFlag = "c" Then
        z = 1
    ElseIf CallPutFlag = "p" Then
        z = -1
    End If

    For i = 1 To nSimulations
        St = S * Exp(Drift + vSqrdt * Application.NormInv(Rnd(), 0, 1))
        sum = sum + Max(z * (St - X), 0)
        If z = 1 And St > X Then DeltaSum = DeltaSum + St
        If z = -1 And St < X Then DeltaSum = DeltaSum + St
        If Abs(St - X) < 2 Then GammaSum = GammaSum + 1
    Next

    '// Option value:
    Output(0) = Exp(-r * T) * sum / nSimulations
     '// Delta:
    Output(1) = z * Exp(-r * T) * DeltaSum / (nSimulations * S)
    '// Gamma:
    Output(2) = Exp(-r * T) * (X / S)^2 * GammaSum / (4 * nSimulations)
    '// Theta:
    Output(3) = (r * Output(0) - b * S * Output(1) _
      - 0.5 * v^2 * S^2 * Output(2)) / 365
```

```
    '// Vega:
    Output(4) = Output(2) * v * S^2 * T / 100

    StandardMCWithGreeks = Application.Transpose(Output())

End Function
```

8.1.2 Monte Carlo for Callable Options

So far, we have only looked at using Monte Carlo simulation for a standard European option. There is no need to use simulation to value an option for which we have a closed-form solution. The above application was presented only to illustrate the use of Monte Carlo simulation. The strength of Monte Carlo simulation lies in its generality and that it is very powerful tool to value path-dependent options. One type of path-dependent option that we have not covered yet is the callable option, typically in the form of a callable warrant. These are call options where the owner is forced to exercise if the asset price has traded above a given barrier level, H, for a prespecified number of days in a row. Similarly, a callable put requires the owner to exercise if the asset price trades below a given barrier for a prespecified number of days in a row. While finding a closed-form solution for these options can be difficult, it is straightforward to implement this valuation problem with Monte Carlo methods. Below we have included a VBA implementation.[1]

Computer algorithm

This function can be used to price standard European call and put options that are callable if the asset price is *MovingDaysN* above the barrier H.

```
Function CallableWarrantNDays(CallPutFlag As String, S As Double, _
  X As Double, H As Double, T As Double, r As Double, b As Double, _
  v As Double, DaysPerYear As Integer, _
  nSimulations As Long, MovingDaysN As Integer) As Double

    Dim i As Long, j As Long
    Dim n As Long, Counter As Long
    Dim z As Integer
    Dim dt As Double, St As Double, sum As Double
    Dim Drift As Double, vSqrt As Double
    Dim BarrierHitProb As Double

    z = 1
    If CallPutFlag = "p" Then
        z = -1
    End If
```

[1]This is a very time-consuming simulation. For any practical purpose, such a simulation should be implemented in a fast language like C++.

```
    n = DaysPerYear * T
    dt = T / n
    Drift = (b - v * v * 0.5) * dt
    vSqrt = v * Sqr(dt)
    sum = 0

  For j = 1 To nSimulations
      BarrierHitProb = 0
      St = S
      Counter = 0
      For i = 2 To n '//starts at second fixing

          St = St * Exp(Drift + vSqrt _
            * Application.NormInv(Rnd(), 0, 1))

          If z = 1 Then '// call
              If St > H Then
                  Counter = Counter + 1
              Else
                  Counter = 0
              End If
          ElseIf z = -1 Then '//put
              If St < H Then
                  Counter = Counter + 1
              Else
                  Counter = 0
              End If
          End If

          If Counter = MovingDaysN Then
            sum = sum + Exp(-r * (i * dt)) * Max(z * (St - X), 0)
            BarrierHitProb = 1
            Exit For
        End If
    Next

      sum = sum + Exp(-r * T) * (1 - BarrierHitProb) _
        * Max(z * (St - X), 0)
  Next

  CallableWarrantNDays = sum / nSimulations

End Function
```

8.1.3 Two Assets

Monte Carlo simulation can easily be extended to options on two underlying assets:

$$S_1 + \Delta S_1 = S_1 \exp\left[\left(\mu_1 - \frac{1}{2}\sigma_1^2\right)\Delta t + \sigma_1 \alpha_{1,t}\sqrt{\Delta t}\right]$$

$$S_2 + \Delta S_2 = S_2 \exp\left[\left(\mu_2 - \frac{1}{2}\sigma_2^2\right)\Delta t + \sigma_2 \alpha_{2,t}\sqrt{\Delta t}\right]$$

Correlation between the two assets is allowed by setting

$$\alpha_{1,t} = \epsilon_{1,t}$$

$$\alpha_{2,t} = \rho\epsilon_{1,t} + \epsilon_{2,t}\sqrt{1-\rho^2},$$

where $\epsilon_{1,t}$ and $\epsilon_{2,t}$ are two independently random numbers from a standard normal distribution.

Most options on two correlated assets can more efficiently be priced analytically or in a three-dimensional binomial lattice, shown earlier in Chapter 7. One example where two-asset Monte Carlo simulation is useful is in the pricing of European Asian spread options.[2] These are options whose payoff depends on the difference between the arithmetic average of the two assets at expiration.

Computer algorithm

The computer code below shows how to price a European arithmetic average spread call or put option. The code is limited to pricing the option at the beginning of the averaging period. It is easy to extend the code to price the option before or into the averaging period.

```
Function MonteCarloAsianSpreadOption(CallPutFlag As String, S1 As Double, _
  S2 As Double, X As Double, T As Double, r As Double, b1 As Double, _
  b2 As Double, v1 As Double, v2 As Double, rho As Double, _
  nSteps As Long, nSimulations As Long) As Double

    Dim dt As Double, St1 As Double, St2 As Double
    Dim i As Long, j As Long, z As Integer
    Dim sum As Double, Drift1 As Double, Drift2 As Double
    Dim v1Sqrdt As Double, v2Sqrdt As Double
    Dim Epsilon1 As Double, Epsilon2 As Double
    Dim Average1 As Double, Average2 As Double

    If CallPutFlag = "c" Then
        z = 1
    ElseIf CallPutFlag = "p" Then
        z = -1
    End If

    dt = T / nSteps
    Drift1 = (b1 - v1^2 / 2) * dt
    Drift2 = (b2 - v2^2 / 2) * dt
    v1Sqrdt = v1 * Sqr(dt)
    v2Sqrdt = v2 * Sqr(dt)

    For i = 1 To nSimulations
        Average1 = 0
        Average2 = 0
        St1 = S1
        St2 = S2
        For j = 1 To nSteps
            Epsilon1 = Application.NormInv(Rnd(), 0, 1)
            Epsilon2 = rho * Epsilon1 _
              + Application.NormInv(Rnd(), 0, 1) * Sqr(1 - rho^2)
            St1 = St1 * Exp(Drift1 + v1Sqrdt * Epsilon1)
            St2 = St2 * Exp(Drift2 + v2Sqrdt * Epsilon2)
            Average1 = Average1 + St1
            Average2 = Average2 + St2
        Next
        Average1 = Average1 / nSteps
        Average2 = Average2 / nSteps
```

[2]For pricing Asian spread options, see Heenk, Kemna, and Vorst (1990).

```
        sum = sum + Max(z * (Average1 - Average2 - X), 0)
    Next

    MonteCarloAsianSpreadOption = Exp(-r * T) * sum / nSimulations

End Function
```

8.1.4 Three Assets

Monte Carlo simulation can easily be extended to options on three underlying assets:

$$S_1 + \Delta S_1 = S_1 \exp\left[\left(\mu_1 - \frac{1}{2}\sigma_1^2\right)\Delta t + \sigma_1 \alpha_{1,t}\sqrt{\Delta t}\right]$$
$$S_2 + \Delta S_2 = S_2 \exp\left[\left(\mu_2 - \frac{1}{2}\sigma_2^2\right)\Delta t + \sigma_2 \alpha_{2,t}\sqrt{\Delta t}\right]$$
$$S_3 + \Delta S_3 = S_3 \exp\left[\left(\mu_3 - \frac{1}{2}\sigma_3^2\right)\Delta t + \sigma_3 \alpha_{3,t}\sqrt{\Delta t}\right]$$

Correlation between the three assets is allowed by setting

$$\alpha_{1,t} = \epsilon_{1,t}$$
$$\alpha_{2,t} = \rho_{1,2}\epsilon_{1,t} + \epsilon_{2,t}\sqrt{1 - \rho_{1,2}^2}$$
$$\alpha_{3,t} = \frac{\epsilon_{3,t}}{g} + (\rho_{2,3} - \rho_{1,3}\rho_{1,2})\epsilon_{2,t} + \rho_{1,3}\epsilon_{1,t}\sqrt{\frac{1}{1 - \rho_{1,2}^2}},$$

where

$$g = \sqrt{\frac{1 - \rho_{1,3}^2}{1 - \rho_{1,2}^2 - \rho_{2,3}^2 - \rho_{1,3}^2 + 2\rho_{1,2}\rho_{1,3}\rho_{2,3}}}$$

and $\epsilon_{1,t}$, $\epsilon_{2,t}$, and $\epsilon_{3,t}$ are three independently drawn random numbers from a standard normal distribution. Moreover, $\rho_{1,2}$ is the correlation between the returns of asset one and two, $\rho_{1,3}$ is the correlation between the returns of asset one and three, and $\rho_{2,3}$ is the correlation between the returns of asset two and three.

Option on Maximum of Two Spread Options

To illustrate Monte Carlo simulation for an option on three correlated assets, let's look at an option that pays off the maximum of two spread options. That is, the payoff is $\max(z(S_1 - S_2 - X), z(S_3 - S_2 - X), 0)$, where z takes the value 1 for a call and -1 for a put. The computer code for such an option follows.

Computer algorithm

```
Function MonteCarloTripleAsset(CallPutFlag As String, S1 As Double, _
  S2 As Double, S3 As Double, X As Double, T As Double, r As Double, _
  b1 As Double, b2 As Double, b3 As Double, v1 As Double, v2 As Double, _
  v3 As Double, rho12 As Double, rho13 As Double, _
  rho23 As Double, nSimulations As Long) As Double

    Dim dt As Double, St1 As Double, St2 As Double, St3 As Double
    Dim i As Long, z As Integer
    Dim sum As Double, g As Double
    Dim Drift1 As Double, Drift2 As Double, Drift3 As Double
    Dim v1Sqrdt As Double, v2Sqrdt As Double, v3Sqrdt As Double
    Dim Epsilon1 As Double, Epsilon2 As Double, Epsilon3 As Double
    Dim alpha2 As Double, alpha3 As Double

    z = 1
    If CallPutFlag = "p" Then
        z = -1
    End If

    Drift1 = (b1 - v1 * v1 / 2) * T
    Drift2 = (b2 - v2 * v2 / 2) * T
    Drift3 = (b2 - v3 * v3 / 2) * T
    v1Sqrdt = v1 * Sqr(T)
    v2Sqrdt = v2 * Sqr(T)
    v3Sqrdt = v3 * Sqr(T)
      g = Sqr((1 - rho13 ^ 2) / (1 - rho12 ^ 2 - rho23 ^ 2 _
        - rho13 ^ 2 + 2 * rho12 * rho13 * rho23))

    sum = 0

    For i = 1 To nSimulations

        St1 = S1
        St2 = S2
        St3 = S3

        Epsilon1 = Application.NormInv(Rnd(), 0, 1)
        Epsilon2 = Application.NormInv(Rnd(), 0, 1)
        Epsilon3 = Application.NormInv(Rnd(), 0, 1)
        alpha2 = rho12 * Epsilon1 + Epsilon2 * Sqr(1 - rho12 ^ 2)
        alpha3 = Epsilon3 / g + (rho23 - rho13 * rho12) * Epsilon2 _
        + rho13 * Epsilon1 * Sqr(1 / (1 - rho12 ^ 2))

        St1 = St1 * Exp(Drift1 + v1Sqrdt * Epsilon1)
        St2 = St2 * Exp(Drift2 + v2Sqrdt * alpha2)
        St3 = St3 * Exp(Drift3 + v3Sqrdt * alpha3)

        sum = sum + Application.Max(z * (St1 - St2 - X), _
         z * (St3 - St2 - X), 0)
    Next

    MonteCarloTripleAsset = Exp(-r * T) * sum / nSimulations

End Function
```

8.1.5 *N* Assets, Cholesky Decomposition

To build in a correlation matrix for multiple assets in Monte Carlo simulation, one can utilize the Cholesky decomposition. Assume we have multiple uncorrelated random numbers $\epsilon_1, \epsilon_2, \epsilon_3, \ldots, \epsilon_N$, then Cholesky decomposition is used to transform these into correlated variables $\alpha_1, \alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_N$. Let us define ϵ and α as column vectors with ϵ_i and α_i in the rows. Now we can transform ϵ to α by

$$\alpha = M\epsilon,$$

where M is a matrix that must satisfy

$$MM^T = R$$

where R is a symmetric positive definite correlation matrix. That is, we are decomposing the correlation matrix into the product of two matrices. M multiplied by the transposed matrix M^T will naturally return the input matrix R. There are several ways to decompose the matrix; Cholesky decomposition is a popular method. For example, the two asset bivariate case is a special case of the multi-asset case

$$R = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

and

$$M = \begin{bmatrix} \frac{1}{\sqrt{1-\rho^2}} & \frac{-\rho}{\sqrt{1-\rho^2}} \\ 0 & 1 \end{bmatrix}$$

In the multi-asset case, we need a computer algorithm,[3] as follows.

Computer algorithm

The computer code takes the positive definite correlation matrix R as input and returns the M matrix.

```
Function CholeskyDecomposition(R As Object) As Variant

Dim a() As Double
Dim M() As Double
Dim i As Integer, j As Integer, n As Integer, h As Integer
Dim U As Double

'// Number of columns in input correlation matrix R n =
R.Columns.Count

ReDim a(1 To n, 1 To n)
ReDim M(1 To n, 1 To n)

For i = 1 To n
    For j = 1 To n
        a(i, j) = R(i, j)
        M(i, j) = 0
     Next
Next
```

[3]The computer algorithm here is based on Wilmott (2000), which gives a more detailed description of Cholesky decomposition.

```
For i = 1 To n
    For j = i To n
     U = a(i , j)
    For h = 1 To (i - 1)
        U = U - M(i , h) * M(j , h)
    Next
    If j = i Then
        M(i , i) = Sqr(U)
    Else
        M(j , i) = U / M(i , i)
    End If
    Next
Next

CholeskyDecomposition = M

End Function
```

8.2 MONTE CARLO OF MEAN REVERSION

One of the great strengths of Monte Carlo simulation is that it can easily be applied to any sensible stochastic process. We next look at an example of how to implement a popular choice for a mean reverting asset price:

$$dS = \kappa(\theta - S_t)dt + \sigma S_t^{\beta} dz,$$

where θ is the mean reversion level, κ is the speed of mean reversion, and β determines the structural form of the diffusion term; consider, for instance, $\beta = 1$, which yields the lognormal mean reversion model, and $\beta = 0$ yields the normally distributed mean reversion model (aka the Ornstein-Uhlenbeck process). When we want to do Monte Carlo simulation of a mean reversion stochastic process, we need to divide each sample path into discrete time steps, even if we only are going to value non-path-dependent European options.

Computer algorithm

This function can be used to price standard European call and put options assuming the asset price follows a mean reversion process.

```
Function MonteCarloMeanReverting(CallPutFlag As String, S As Double, _
X As Double, T As Double, r As Double, b As Double, v As Double, _
kappa As Double, theta As Double, beta As Double, _
nSteps As Long, nSimulations As Long) As Double

    Dim dt As Double, St As Double
    Dim sum As Double
    Dim i As Long, j As Long, z As Integer

    dt = T / nSteps

    If CallPutFlag = "c" Then
        z = 1
    ElseIf CallPutFlag = "p" Then
        z = -1
```

```
    End If

    For i = 1 To nSimulations
     St = S
        For j = 0 To nSteps
            St = St + kappa * (theta - St) * dt _
            + v * St^beta * v * Sqr(dt) _
            * Application.NormInv(Rnd(), 0, 1)
        Next
        sum = sum + Max(z * (St - X), 0)

    Next

    MonteCarloMeanReverting = Exp(-r * T) * sum / nSimulations

End Function
```

8.3 GENERATING PSEUDO-RANDOM NUMBERS

Random numbers, also called random variates, are an important input for Monte Carlo simulations. A computer cannot generate completely random numbers, but a good random-number generator should be good at simulating numbers that are hard to distinguish from real randomness.

Random Numbers From the Standard Normal Distribution

Most computer languages offer built-in functions that draw randomly a number Z uniformly distributed between 0 and 1. In Monte Carlo simulations we typically need to transform this into a random number from a standard normal distribution ϵ. Several methods are suggested in the literature.

One very simple and intuitive method is to use the following formula:

$$\epsilon = \sum_{i=1}^{12} Z_i - 6$$

The drawback is that it uses unnecessary computer time, as it uses many independent random numbers to generate one random number. Another drawback is that it is also not accurate in the tail of the distribution. To increase the accuracy in the tail of the distribution, we can simply increase the number of summands, for example, to

$$\epsilon = \sum_{i=1}^{24} Z_i - 12$$

This will, however, just add more computer time. A better alternative is what is known as the Box-Muller method:

$$\epsilon = \cos(2\pi Z_2)\sqrt{-2\ln(Z_1)}, \tag{8.5}$$

where cos is the cosine function, and Z_1 and Z_2 are two independent uniform random numbers between 0 and 1. The simple procedure above requires two random numbers to return one; however, the method can easily be implemented to return two random numbers, as shown in the code below. Also, using tangens instead of cosinus is known to increase the speed.

Computer algorithm

```
Function BoxMuller2(x1 As Double, x2 As Double) As Variant

    Dim T As Double, L As Double
    Dim ReturnVec(1 To 2)

    If x1 = 0 Then
        BoxMuller2(x1, x2) = BoxMuller(x2, x1)
    Else
     '/ Using tan(Pi*x2) instead of cos and sin increases the speed
        T = Tan(Pi * x2)
        L = Sqr(-2 * Log(x1))
        ReturnVec(1) = L * (1 - T * T) / (1 + T * T)
        ReturnVec(2) = L * 2 * T / (1 + T * T)
        BoxMuller2 = ReturnVec()
    End If

End Function
```

Another alternative is to use the inverse cumulative normal distribution function to do the job. This is a procedure that works for any distribution for which we are able to efficiently express the inverse cumulative distribution function. In the particular case of the normal distribution, this is possible, and

$$\epsilon = N^{-1}(Z)$$

This is a very efficient method. Since there is no closed-form solution for the inverse cumulative normal distribution function, we must make sure we use an accurate approximation. One method is presented in Chapter 13. Excel also has a built-in inverse cumulative normal distribution function that we, for convenience, have used in many of the Monte Carlo VBA examples.

Built-in Excel Function

In Excel there is a built-in random-number generator. Normally distributed random variates can be easily generated directly in an Excel spreadsheet by invoking "*NormInv*(*Rnd*())". Be aware that this is not a good random-number generator, but for simple single-asset options, it seems to work fine. In particular, for multi-asset options, you need a better implemented random-number generator. Note also that random-number generators that come with standard distributions of lower-level languages like C++ are generally of low quality.

8.4 VARIANCE REDUCTION TECHNIQUES

Standard Monte Carlo simulation is not very accurate when a "moderate" number of random variates are used. A large number of simulations is typically needed to get a reasonably accurate result. With this in mind, we next look at several methods that speed up and make Monte Carlo simulations more accurate. I discuss how to implement

- Antithetic variance reduction
- Intelligent Monte Carlo (IQ-MC, aka importance sampling)
- Quasi-random Monte Carlo simulation
- Combining quasi-random MC with importance sampling

8.4.1 Antithetic Variance Reduction

The antithetic variance reduction technique involves calculating the derivative's value twice for each simulation: A path is simulated in the normal way, and then a mirror path is generated by switching the sign of the random variates drawn to generate the original path. The two paths are then used to compute two different option values. One, therefore, has to simulate fewer random variates (half as many) to simulate a given number of option values. This is a very robust and simple variance reduction to implement, and can be used in almost any type of Monte Carlo simulation.

Computer algorithm antithetic variance reduction
This function can be used to price standard European call and put options using pseudo-random Monte Carlo simulation with antithetic variance reduction technique.

```
Function MonteCarloStandardOptionAntithetic(CallPutFlag As String, _
  S As Double, X As Double, T As Double, r As Double, b As Double, _
  v As Double, nSimulations As Long) As Double

    Dim St1 As Double, St2 As Double, Epsilon As Double
    Dim sum As Double, Drift As Double, vSqrdt As Double
    Dim i As Long, j As Long, z As Integer

    Drift = (b - v^2 / 2) * T
    vSqrdt = v * Sqr(T)

    If CallPutFlag = "c" Then
        z = 1
    ElseIf CallPutFlag = "p" Then
        z = -1
    End If

    For i = 1 To nSimulations
        Epsilon = Application.NormInv(Rnd(), 0, 1)
        St1 = S * Exp(Drift + vSqrdt * Epsilon)
        St2 = S * Exp(Drift + vSqrdt * (-Epsilon))
        sum = sum + (Max(z * (St1 - X), 0) + Max(z * (St2 - X), 0)) / 2
```

```
    Next
    MonteCarloStandardOptionAntithetic = Exp(-r * T) * sum / nSimulations
End Function
```

8.4.2 IQ-MC/Importance Sampling

This technique involves valuing options by simulation-only paths that end up in-the-money. The applicability of this technique, of course, requires that only these paths are relevant for the option value. In standard Monte Carlo simulation, in contrast, all paths are taken into account. For example, when one is valuing a deep out-of-the-money option using 10,000 simulations, it can be that only a couple of thousand or just a few hundred of the simulated paths end up in-the-money. All the paths ending out-of-the-money are "useless" for the valuation and just a waste of computer time. Standard Monte Carlo simulation is for this reason extremely inefficient for out-of-the-money options. A method known as importance sampling is based on the idea that only "important" paths need to be simulated, in the case of non-path-dependent options—that is, paths that end up in-the-money. This is, in other words, to make the Monte Carlo simulation "intelligent" and is why I like to call it IQ-MC (IQ Monte Carlo). Importance sampling, or what I will describe here as IQ-MC, was probably first described in relation to quantitative finance by Reider (1993) and later described by Boyle, Broadie, and Glasserman (1997); Glasserman, Heidelberg, and Shahabuddin (2000); and Su and Fu (2000), among others.

IQ-MC can be used to value options that depend on a wide variety of stochastic processes. We will for simplicity limit ourselves to geometric Brownian motion and standard European options. Standard European options under geometric Brownian motion can naturally be valued much more efficiently with the BSM formula. This section is just to illustrate how powerful and efficient IQ-MC is compared to standard Monte Carlo simulation.

IQ-MC can be done by the following steps:

Sample random numbers that will lead to in-the-money paths:

$$X = S\exp\left[\left(b - \frac{1}{2}\sigma^2\right)T + \sigma\sqrt{T}N^{-1}(\epsilon)\right]$$

solved with respect to ϵ gives

$$N^{-1}(\epsilon) = \frac{\ln(X/S) - (b - \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$\epsilon = N\left(\frac{\ln(X/S) - (b - \sigma^2/2)T}{\sigma\sqrt{T}}\right),$$

where $N(\cdot)$ is the cumulative normal distribution function.

For a call option, this means that only random numbers between $N(\frac{\ln(X/S)-(b-\sigma^2/2)T}{\sigma\sqrt{T}})$ and 1 will lead to in-the-money options. For put options, only random numbers between 0 and $N(\frac{\ln(X/S)-(b-\sigma^2/2)T}{\sigma\sqrt{T}})$ will lead to in-the-money paths.

These in-the-money paths then just have to be weighted by the fraction of paths that would go in-the-money assuming one used standard MC. For a call, we simply have to multiply by the risk-neutral probability of ending up in-the-money. For a European call this is $N(d_2)$, where

$$d_2 = \frac{\ln(X/S) + (b - \sigma^2/2)T}{\sigma\sqrt{T}},$$

while for a put option, we use $N(-d_2)$. This gives us the value of the call option

$$c = N(d_2)\sum_0^n \left\{S\exp\left[\left(b - \frac{1}{2}\sigma^2\right)T + \sigma\sqrt{T}N^{-1}(\epsilon_c)\right] - X\right\}, \tag{8.6}$$

where ϵ_c is a random number between $N(-d)$ and 1. If we have an algorithm for generating a random number ϵ between 0 and 1, all we need is to compute

$$\epsilon_c = [1 - N(d)]\epsilon + N(d)]$$

Similarly for a put, we have

$$p = N(-d_2)\sum_0^n \left\{X - S\exp\left[\left(b - \frac{1}{2}\sigma^2\right)T + \sigma\sqrt{T}N^{-1}(\epsilon_p)\right]\right\}, \tag{8.7}$$

where ϵ_p is a random number between 0 and $N(-d)$. If we have an algorithm to generate a random number ϵ between 0 and 1, all we need is to compute

$$\epsilon_p = N(d)\epsilon$$

Computer algorithm for standard European option using IQ-MC

This function can be used to price standard European call and put options. The code illustrates how to use IQ-MC (importance sampling). For improved accuracy and more complex derivatives, one should replace Excel's random-number generator, Rnd(), with a better random-number generator.

```
Function IQMC(CallPutFlag As String, S As Double, X As Double, _
  T As Double, r As Double, b As Double, v As Double, _
   nSimulations As Long) As Double

    Dim St As Double
    Dim sum As Double, Drift As Double, vSqrdt As Double
    Dim i As Long, z As Integer
    Dim d As Double, d2 As Double, Epsilon As Double

    Drift = (b - v^2 / 2) * T
    vSqrdt = v * Sqr(T)
    d = (Log(X / S) - (b - v^2 / 2) * T) / (v * Sqr(T))
    d2 = (Log(S / X) + (b - v^2 / 2) * T) / (v * Sqr(T))

    If CallPutFlag = "c" Then
        z = 1
    ElseIf CallPutFlag = "p" Then
        z = -1
    End If

    For i = 1 To nSimulations
        If z = -1 Then
            Epsilon = CND(d) * Rnd()
        Else
            Epsilon = (1 - CND(d)) * Rnd() + CND(d)
        End If
        St = S * Exp(Drift + vSqrdt * Application.NormInv(Epsilon, 0, 1))
        sum = sum + Max(z * (St - X), 0)
    Next
    IQMC = Exp(-r * T) * sum / nSimulations * CND(z * d2)

End Function
```

8.4.3 IQ-MC Two Correlated Assets

IQ-MC can easily be extended to options on two correlated assets. Here we illustrate the procedure to value a two-asset correlation option. A closed-form solution for such an option is given in Chapter 5, covering two-asset exotic options. However, this gives us a good opportunity to test out how accurate our implementation is. The implementation follows below.

```
Function IQMC2Asset(CallPutFlag As String, S1 As Double, S2 As Double, _
        x1 As Double, x2 As Double, T As Double, r As Double, _
        b1 As Double, b2 As Double, v1 As Double, v2 As Double, _
        rho As Double, nSimulations As Long) As Double

    Dim dt As Double, St1 As Double, St2 As Double
    Dim i As Long, j As Long, z As Integer
    Dim sum As Double, Drift1 As Double, Drift2 As Double
    Dim v1Sqrdt As Double, v2Sqrdt As Double
    Dim y1 As Double, y2 As Double, dd As Double
    Dim Epsilon1 As Double, Epsilon2 As Double
    Dim d As Double, d2 As Double

    If CallPutFlag = "c" Then
        z = 1
    ElseIf CallPutFlag = "p" Then
        z = -1
    End If
```

```
    Drift1 = (b1 - v1^2 / 2) * T
    Drift2 = (b2 - v2^2 / 2) * T
    v1Sqrdt = v1 * Sqr(T)
    v2Sqrdt = v2 * Sqr(T)

    y1 = (Log(S1 / x1) + (b1 - v1^2 / 2) * T) / (v1 * Sqr(T))
    y2 = (Log(S2 / x2) + (b2 - v2^2 / 2) * T) / (v2 * Sqr(T))
    d = (Log(x1 / S1) - (b1 - v1^2 / 2) * T) / (v1 * Sqr(T))
    dd = (Log(x2 / S2) - (b2 - v2^2 / 2) * T) / (v2 * Sqr(T))

    For i = 1 To nSimulations

        If z = -1 Then
            Epsilon1 = CND(d) * Rnd()
        Else
            Epsilon1 = (1 - CND(d)) * Rnd() + CND(d)
        End If
        Epsilon1 = Application.NormInv(Epsilon1, 0, 1)
        Epsilon2 = CND((dd - rho * Epsilon1) / Sqr(1 - rho * rho))
        If z = 1 Then
            Epsilon2 = (1 - Epsilon2) * Rnd() + Epsilon2
        Else
            Epsilon2 = Epsilon2 * Rnd()
        End If
        Epsilon2 = rho * Epsilon1 + Application.NormInv(Epsilon2, 0, 1) _
         * Sqr(1 - rho ^ 2)
        St1 = S1 * Exp(Drift1 + v1Sqrdt * Epsilon1)
        St2 = S2 * Exp(Drift2 + v2Sqrdt * Epsilon2)
        sum = sum + z * (St2 - x2)
    Next

    IQMC2Asset = Exp(-r * T) * sum / nSimulations _
     * CBND(z * y1, z * y2, rho)

End Function
```

8.4.4 Quasi-Random Monte Carlo

Quasi-random numbers, also known as low-discrepancy sequences, are nonrandom series of numbers. Quasi-random numbers are much more evenly spread out than random numbers. This make the method much more efficient than standard Monte Carlo simulation.

Halton Numbers

Halton (1960) numbers are a simple way of generating quasi-random numbers. Halton numbers are not very efficient for high-dimensional problems, but the method illustrates very well the added efficiency in using quasi-random numbers instead of standard pseudo-random Monte Carlo simulation. Quasi-random numbers better suited for high-dimensional problems are described by Sobol (1967) and Faure (1982). For more details on quasi-random numbers in financial applications see also Wilmott (2000), Jäckel (2002), and Glasserman (2003).

Computer algorithm

Below is the computer code for generating quasi-random Halton numbers.

```
Function Halton(n, b)
    Dim n0, n1, r As Integer
    Dim H As Double
    Dim f As Double
    n0 = n
    H = 0
    f = 1 / b
    While (n0 > 0)
        n1 = Int(n0 / b)
        r = n0 - n1 * b
        H = H + f * r
        f = f / b
        n0 = n1
    Wend
    Halton = H
End Function
```

Computer algorithm

Below is the computer code for a European option using quasi-random Monte Carlo simulation with Halton numbers.

```
Function HaltonMonteCarloStandardOption(CallPutFlag As String, _
    S As Double, X As Double, T As Double, r As Double, b As Double, _
    v As Double, nSimulations As Long) As Double

    Dim St As Double
    Dim sum As Double, Drift As Double, vSqrdt As Double
    Dim i As Long, z As Integer

    Drift = (b - v^2 / 2) * T
    vSqrdt = v * Sqr(T)

    If CallPutFlag = "c" Then
        z = 1
    ElseIf CallPutFlag = "p" Then
        z = -1
    End If

    For i = 1 To nSimulations
        St = S * Exp(Drift + vSqrdt _
          * BoxMuller(Halton(i, 3), Halton(i, 5)))
        sum = sum + Max(z * (St - X), 0)
    Next

    HaltonMonteCarloStandardOption = Exp(-r * T) * sum / nSimulations

End Function

Function BoxMuller(x As Double, y As Double) As Double

    BoxMuller = Sqr(-2 * Log(x)) * Cos(2 * Application.Pi() * y)

End Function
```

Quasi-Random IQ-MC

By combining quasi-random Monte Carlo simulation with importance sampling, we get a very intelligent form of simulation that has little or nothing to do with random numbers. The method is very powerful and efficient (super IQ-MC). Below is an example of how to implement it for a standard option.

Computer algorithm

This function can be used to price standard European call and put options. The code illustrates how to use Super IQ-MC (quasi-random MC in combination with importance sampling). Only the in-the-money paths are simulated using quasi-random Halton numbers. The computer code calls the Halton function given in the section describing quasi-random number generation.

```
Function SuperIQMC(CallPutFlag As String, S As Double, _
    X As Double, T As Double, r As Double, b As Double, _
    v As Double, nSimulations As Long) As Double

    Dim St As Double
    Dim sum As Double, Drift As Double, vSqrdt As Double
    Dim i As Long, z As Integer
    Dim d As Double, d2 As Double, Epsilon As Double

    Drift = (b - v^2 / 2) * T
    vSqrdt = v * Sqr(T)
    d = (Log(X / S) - (b - v^2 / 2) * T) / (v * Sqr(T))
    d2 = (Log(S / X) + (b - v^2 / 2) * T) / (v * Sqr(T))

    If CallPutFlag = "c" Then
        z = 1
    ElseIf CallPutFlag = "p" Then
        z = -1
    End If

    For i = 1 To nSimulations
        If z = -1 Then
            Epsilon = CND(d) * Halton(i, 5)
        Else
            Epsilon = (1 - CND(d)) * Halton(i, 5) + CND(d)
        End If
        St = S * Exp(Drift + vSqrdt * Application.NormInv(Epsilon, 0, 1))
        sum = sum + Max(z * (St - X), 0)
    Next
    SuperIQMC = Exp(-r * T) * sum / nSimulations * CND(z * d2)

End Function
```

8.5 AMERICAN OPTION MONTE CARLO

In general, Monte Carlo simulation is limited to value European options/derivatives instruments. However, Boyle, Broadie, and Glasserman (1997) have developed a sophisticated method for valuing American derivatives using Monte Carlo simulation. Here I will simply give a quick example of how to implement it in VBA code.

Computer algorithm

The VBA algorithm shown here shows, how to implement the Boyle, Broadie, and Glasserman American Monte Carlo simulation for single-asset options. The implementation is based on programming done by Meier (2000). To understand the code the best, you should play with the accompanying spreadsheet.

```
Function BroadieGlasserman(CallPutFlag As String, S As Double, _
  X As Double, T As Double, r As Double, b As Double, Sig As Double, _
  m As Integer, Branches As Integer, nSimulations As Integer) As Double

    Dim Drift As Double, SigSqrdt As Double, Discdt As Double
    Dim z As Integer
    Dim i As Integer, j As Integer, i1 As Integer, i2 As Integer
    Dim Estimator As Integer, Simulation As Integer
    Dim EstimatorSum As Double, Sum1 As Double, Sum2 As Double
    Dim w() As Integer, v() As Double, Estimators() As Double
    ReDim w(1 To m) As Integer, v(1 To Branches, 1 To m) As Double
    ReDim Estimators(1 To 2) As Double

    z = 1
    If CallPutFlag = "Put" Then z = -1

    Drift = (b - Sig ^ 2 / 2) * T / (m - 1)
    SigSqrdt = Sig * Sqr(T / (m - 1))
    Discdt = Exp(-r * T / (m - 1))

    For Estimator = 1 To 2
        EstimatorSum = 0
        For Simulation = 1 To nSimulations
            v(1, 1) = S
            w(1) = 1
            For j = 2 To m
                v(1, j) = v(1, j - 1) * Exp(Drift + SigSqrdt _
                * NormInv(Rnd, 0, 1))
                w(j) = 1
            Next j
            j = m
            Do While j > 0
                If j = m Then
                    v(w(j), j) = Max(z * (v(w(j), j) - X), 0)
                    If w(j) < Branches Then
                        v(w(j) + 1, j) = v(w(j - 1), j - 1) _
             * Exp(Drift + SigSqrdt * NormInv(Rnd, 0, 1))
                        w(j) = w(j) + 1
                    ElseIf w(j) = Branches Then
                        w(j) = 0
                        j = j - 1
                    End If
                ElseIf j < m Then
                    If Estimator = 1 Then 'the high estimator
                        Sum1 = 0
                        For i1 = 1 To Branches
                            Sum1 = Sum1 + Discdt * v(i1, j + 1)
                        Next i1
                        v(w(j), j) = Max(Max(z * (v(w(j), j) - X), 0), _
                Sum1 / Branches)
                    ElseIf Estimator = 2 Then 'the low estimator
                        Sum1 = 0 'value determinant
                        For i1 = 1 To Branches
                            Sum2 = 0 'decision determinant
                            For i2 = 1 To Branches
                                If i1 <> i2 Then Sum2 = Sum2 _
                                 + Discdt * v(i2, j + 1)
```

```
                        Next i2
                        If Max(z * (v(w(j), j) - X), 0) >= Sum2 _
                          / (Branches - 1) Then
                            Sum1 = Sum1 + Max(z * (v(w(j), j) - X), 0)
                        Else
                            Sum1 = Sum1 + Discdt * v(i1, j + 1)
                        End If
                    Next i1
                    v(w(j), j) = Sum1 / Branches
                End If
                If w(j) < Branches Then
                    If j > 1 Then
                        v(w(j) + 1, j) = v(w(j - 1), j - 1) _
            * Exp(Drift + SigSqrdt * NormInv(Rnd, 0, 1))
                        w(j) = w(j) + 1
                        For i = j + 1 To m
                            v(1, i) = v(w(j), j) _
            * Exp(Drift + SigSqrdt * NormInv(Rnd, 0, 1))
                            w(i) = 1
                        Next i
                        j = m
                    Else
                        j = 0
                    End If
                ElseIf w(j) = Branches Then
                    w(j) = 0
                    j = j - 1
                End If
            End If
        Loop
        EstimatorSum = EstimatorSum + v(1, 1)
    Next Simulation
    Estimators(Estimator) = EstimatorSum / nSimulations
Next Estimator
BroadieGlasserman = 0.5 * Max(Max(z * (S - X), 0), _
    Estimators(2)) + 0.5 * Estimators(1)

End Function


Function NormInv(n1 As Double, n2 As Double, n3 As Double) As Double

    NormInv = Application.NormInv(n1, n2, n3)

End Function
```

CHAPTER

9

OPTIONS ON STOCKS THAT PAY DISCRETE DIVIDENDS

Anyone who has never made a mistake has never tried anything new.

Albert Einstein

To find the value of options on a stock index, it is typically suffice to assume the index pays out a continuous dividend yield. Models that can deal with this situation were covered in Chapters 1, 3, and 7. Assuming a continuous payout rate for options on a single stock is clearly less realistic and will consequently yield unacceptable accuracy in the value estimate. A high degree of accuracy can be attained by assuming the stock pays out known discrete dividends during the option's lifetime. Valuation of options on an asset, typically a stock, paying a discrete dividend is considered a relatively complex valuation problem that has received a lot of attention, but also a lot of confusion. Despite options on stocks paying discrete dividends have been trading actively since at least the early 1970s, it was first in 2002 that Haug, Haug, and Lewis (2003) came up with a "closed-form" benchmark model for valuing such options. Although most options on single stocks are American, we start with the European case. Even before that discussion, we will, however, shortly describe one of the main methods described in the literature. Because there is much confusion concerning the valuation of options on stocks paying discrete dividends, this chapter is more wordy than what is typical for this book.

Escrowed Dividend Model The simplest escrowed dividend approach makes a simple adjustment to the Black-Scholes-Merton (BSM) formula. The adjustment consists of replacing the stock price S by the stock price minus the present value of the dividend $S - e^{-rt_D}D$, where D is the size of the cash dividend to be paid at time t_D. Because the initial stock price is reduced below the actual, observed stock price, this approach typically leads to too little absolute price volatility ($\sigma_t S_t$) in the period before the dividend is paid. Moreover, it is just an approximation used to fit the ex-dividend price process into the geometric Brownian motion (GBM) assumption of the BSM formula. The approach will, in general, undervalue call options, and the mispricing is larger the later in the option's lifetime that the dividend is paid. The approximation suggested by Black (1975) for American options suffers from the same problem, as does the Roll-Geske-Whaley (RGW) model Roll (1977), Geske (1979a), and Whaley (1981). The RGW model uses this approximation of the stock price process and applies a compound option approach to take into account the possibility of early exercise. Not only does this yield a poor approximation in certain circumstances, but it can open up arbitrage opportunities!

Several papers discuss the weakness of the escrowed dividend approach. In the case of European options, suggested fixes are often based on adjustments of the volatility in the BSM model, in combination with the escrowed dividend adjustment. We next discuss three such approaches, all of which assume that the stock price can be described by a GBM.

9.1 EUROPEAN OPTIONS ON STOCK WITH DISCRETE CASH DIVIDEND

9.1.1 The Escrowed Dividend Model

European options on a stock that pays out one or more cash dividends during the option's lifetime can be priced by the BSM formula, by simply replacing S with S minus the present value of the dividends. In general, the stock price minus the present value of the dividends can be written as

$$S - D_1 e^{-rt_1} - D_2 e^{-rt_2} \cdots - D_n e^{-rt_n}, \qquad (t_n < T),$$

where D_1 is dividend payout one, t_1 is the time to this dividend payout, t_2 is the time until the second dividend payout, and so on, and T is the time to maturity of the option. Although simple, this approach can lead to significant mispricing and arbitrage opportunities. In particular, it will underprice options where the dividend is close to the option's expiration date. The antithesis of this insight is that the approach may be accurate when there is a single dividend payment just after the option contract is initiated.

Example

Consider a European call option on a stock that will payout a dividend two, three, and six months from now. The current stock price is 100, the strike price is 90, the time to maturity on the option is nine months, the risk-free rate is 10%, and the volatility is 25%. Hence, $S = 100$, $X = 90$, $T = 0.75$, $r = 0.1$, $\sigma = 0.25$, $D_1 = D_2 = 2$, $t_1 = 0.25$, $t_2 = 0.5$.

The stock price minus the net present value of the cash dividends is

$$100 - 2e^{-0.1\times 0.25} - 2e^{-0.1\times 0.5} = 96.1469$$

Next, use the BSM formula:

$$d_1 = \frac{\ln(96.1469/90) + (0.1 + 0.25^2/2)0.75}{0.25\sqrt{0.75}} = 0.7598$$

$$d_2 = d_1 - 0.25\sqrt{0.75} = 0.5433$$

$$N(d_1) = N(0.7598) = 0.7763 \qquad N(d_2) = N(0.5433) = 0.7065$$

$$c \approx 96.1469N(d_1) - 90e^{-0.1\times 0.75}N(d_2) = 15.6465$$

9.1.2 Simple Volatility Adjustment

To reduce the volatility problem mentioned in the escrowed dividend model, several adjustments to the volatility has been suggested in the literature. An adjustment popular among practitioners is to replace the volatility σ with $\sigma_2 = \frac{\sigma S}{S-De^{-rt_D}}$; see, for instance, Chriss (1997). This approach increases the volatility relative to the basic escrowed divided process. However, the adjustment yields too high volatility if the dividend is paid out early in the option's lifetime. The approach typically overprices call options in this situation and can give arbitrage opportunities.

9.1.3 Haug-Haug Volatility Adjustment

The following is a volatility adjustment that has been suggested, used in combination with the escrowed dividend model. The adjustment seems to have been discovered independently by Haug and Haug (1998), as well as by Beneder and Vorst (2001). σ in the BSM formula is replaced with σ_{adj}, and the stock price minus the present value of the dividends until expiration is substituted for the stock price.

$$\begin{aligned}\sigma_{\text{adj}}^2 &\approx \left(\frac{S\sigma}{S - \sum_{i=1}^{n} D_i e^{rt_i}}\right)^2 (t_1 - t_0) + \left(\frac{S\sigma}{S - \sum_{i=2}^{n} D_i e^{rt_i}}\right)^2 \\ &\quad \times (t_2 - t_1) + \cdots + \sigma^2(T - t_n) \\ &= \sum_{j=1}^{n} \left(\frac{S\sigma}{S - \sum_{i=j}^{n} D_i e^{rt_i}}\right)^2 (t_j - t_{j-1}) + \sigma^2(T - t_n) \end{aligned} \tag{9.1}$$

This method seems to work better than, for instance, the volatility adjustment discussed by Chriss (1997). It is still simply a rough approximation, though, with little theory behind it. For this reason, there is no guarantee for it to be accurate in all circumstances.

Computer algorithm

Below is the computer code for calculating the Haug-Haug volatility adjustment to be used in combination with the escrowed dividend process.

```
Function HaugHaugVol(S As Double, T As Double, r As Double, _
    Dividends As Variant, DividendTimes As Variant, v As Double) As Double

    Dim SumDividends As Double, sumVolatilities As Double
    Dim n As Integer, j As Integer, i As Integer

    n = Application.Count(Dividends) ' number of dividends

    sumVolatilities = 0
    For j = 1 To n + 1
        SumDividends = 0
        For i = j To n
            SumDividends = SumDividends + Dividends(i) _
             * Exp(-r * DividendTimes(i))
        Next
        If j = 1 Then
                sumVolatilities = sumVolatilities _
                 + (S * v / (S - SumDividends))^2 * DividendTimes(j)
        ElseIf j < n + 1 Then
            sumVolatilities = sumVolatilities _
                + (S * v / (S - SumDividends)^2 _
        * (DividendTimes(j) - DividendTimes(j - 1))
        Else
            sumVolatilities = sumVolatilities _
            + v^2 * (T - DividendTimes(j - 1))
        End If
    Next
    HaugHaugVol = Sqr(sumVolatilities / T)

End Function
```

9.1.4 Bos-Gairat-Shepeleva Volatility Adjustment

Bos, Gairat, and Shepeleva (2003) suggest the following volatility adjustment to be used in combination with the escrowed dividend adjustment:

$$\sigma(S,X,T)^2 \approx \sigma^2 + \sigma\sqrt{\frac{\pi}{2T}}\left\{4e^{\frac{z_1^2}{2}-s}\sum_{i=1}^{n} D_i e^{-rt_i}\left[N(z_1) - N\left(z_1 - \sigma\frac{t_i}{\sqrt{T}}\right)\right]\right. \tag{9.2}$$

$$\left. + e^{\frac{z_2^2}{2}-2s}\sum_{i}^{n}\sum_{j}^{n} D_i D_j e^{-r(t_i+t_j)}\left[N(z_2) - N\left(z_2 - \frac{2\sigma\min(t_i,t_j)}{\sqrt{T}}\right)\right]\right\},$$

where n is the number of dividends in the option's lifetime, $s = \ln(S)$, $x = \ln[(X + D_T)e^{-rT}]$, where $D_T = \sum_i^n D_i e^{-rt_i}$, and

$$z_1 = \frac{s-x}{\sigma\sqrt{T}} + \frac{\sigma\sqrt{T}}{2} \qquad z_2 = z_1 + \sigma\sqrt{T}$$

The method is quite accurate for most cases. For very large or with multiple dividend payouts, the method can yield significant mispricing, however. See Haug, Haug, and Lewis (2003) for a more detailed discussion.

Computer algorithm

```
Function BosGaiShepVol(S As Double, X As Double, T As Double, r As Double, _
    v As Double, Optional DividendTimes As Object, _
    Optional Dividends As Object)

    Dim n As Integer, i As Integer, j As Integer
    Dim sum1 As Double, sum2 As Double
    Dim z1 As Double, z2 As Double
    Dim ti As Double, tj As Double
    Dim dt As Double

    n = Application.Count(Dividends)
    dt = 0
    For i = 1 To n
        dt = dt + Dividends(i) * Exp(-r * DividendTimes(i))
    Next

    S = Log(S)
    X = Log((X + dt) * Exp(-r * T))

    z1 = (S - X) / (v * Sqr(T)) + v * Sqr(T) / 2
    z2 = (S - X) / (v * Sqr(T)) + v * Sqr(T)

    sum1 = 0
    sum2 = 0

    For i = 1 To n
        ti = DividendTimes(i)
        sum1 = sum1 + Dividends(i) * Exp(-r * ti) _
         * (CND(z1) - CND(z1 - v * ti / Sqr(T)))
        For j = 1 To n
            tj = DividendTimes(j)
            sum2 = sum2 + Dividends(i) * Dividends(j) * Exp(-r * (ti + tj)) _
        * (CND(z2) - CND(z2 - 2 * v * Min(ti, tj)))
        Next
    Next

    BosGaiShepVol = Sqr(v^2 + v * Sqr(Pi / (2 * T)) _
        * (4 * Exp(z1^2 / 2 - S) _
      * sum1 + Exp(z2^2 / 2 - 2 * S) * sum2))

End Function
```

9.1.5 Bos-Vandermark

A slightly different way to implement the escrowed dividend method is to adjust the stock price and strike Bos and Vandermark (2002).

$$c \approx c_{BSM}(S - X_n, X + X_f, T, r, b, \sigma), \tag{9.3}$$

where $c_{BSM}(\cdot)$ is the BSM formula and

$$X_n = \sum_{i=1}^{n} \frac{T - t_i}{T} D_i e^{-rt_i} \qquad X_f = \sum_{i=1}^{n} \frac{t_i}{T} D_i e^{-rt_i},$$

where n is the number of cash dividends in the option's lifetime, t_i is the years to dividend payout i, and D_i is the cash dividend at time t_i. As usual, T is the number of years to maturity on the option.

This approach seems to work better than the approximations mentioned above. It still suffers from approximation errors for large dividends, just like the Bos, Gairat, and Shepeleva (2003) approximation. Numerical investigations indicate that this is a fast and efficient approximation that should work in most practical cases. The main drawback is that it works only for European options, while in practice most options on an asset with discrete dividends are American.

Computer algorithm

Below is the code using the BSM formula with the Bos-Vandermark adjustment for discrete cash dividends. The variable *Dividends* can be an array of the cash dividends, and *DividendTimes* is correspondingly an array with the times the dividends are paid, measured in years to maturity.

```
Function GBlackScholesBVCashDividend(CallPutFlag As String, S As Double, _
  X As Double, T As Double, r As Double, b As Double, v As Double, _
  Optional Dividends As Object, Optional DividendTimes As Object) As Double

        Dim i As Integer, n As Integer
        Dim Xn As Double, Xf As Double

        n = Application.Count(Dividends)
        Xn = 0
        Xf = 0

        For i = 1 To n
            Xn = Xn + (T - DividendTimes(i)) / T * Dividends(i) _
           * Exp(-r * DividendTimes(i))
            Xf = Xf + (DividendTimes(i)) / T * Dividends(i) _
           * Exp(-r * DividendTimes(i))
        Next

        GBlackScholesBVCashDividend = GBlackScholes(CallPutFlag, _
          S - Xn, X + Xf * Exp(r * T), T, r, b, v)

End Function
```

9.2 NON-RECOMBINING TREE

The motivation for using tree models is that they easily handle American options with discrete dividends.

An alternative to the escrowed dividend approximation is to use non-recombining lattice methods. If implemented as a binomial or trinomial tree, one builds a new tree from each node on each dividend payment date. A problem with all non-recombining lattices is that they are time-consuming to evaluate. This problem is amplified with multiple dividends. As pointed out by Haug, Haug, and Lewis (2003), the literature on non-recombining trees does not account for the fact that a constant dividend D can't be paid at arbitrarily low stock prices.[1] Even if the method is "dead wrong" by allowing negative stock prices, numerical investigation indicates that a non-recombining tree still is fairly accurate (and efficient) in the case of one or two dividend payments.

Computer algorithm

```
Function BinomialDiscreteDividends(CallPutFlag As String, _
  AmeEurFlag As String, S As Double, X As Double, T As Double, r As Double,
  v As Double, n As Integer, Optional CashDividends As Variant, _
  Optional DividendTimes As Variant)

    Dim TmpDividendTimes() As Variant
    Dim TmpCashDividends() As Variant
    Dim StockPriceNode() As Double
    Dim OptionValueNode() As Double
    Dim NoOfDividends As Integer, Binary As Integer
    Dim Df As Double, dt As Double
    Dim u As Double, d As Double, uu As Double
    Dim p As Double, z As Double
    Dim i As Integer, j As Integer
    Dim StepsBeforeDividend As Integer
    Dim DividendAmount As Double
    Dim ValueNotExercising As Double

    If IsMissing(DividendTimes) Or IsEmpty(DividendTimes) Then
        NoOfDividends = 0
    Else
        ' // Counts the number of dividend payments
        NoOfDividends = Application.Count(DividendTimes)
    End If

    If NoOfDividends = 0 Then
        ' // If the number of dividends is zero use standard binomial model
        BinomialDiscreteDividends = CRRBinomial(AmeEurFlag, CallPutFlag, S, _
          X, T, r, r, v, n)
        Exit Function
    End If

    ReDim TmpDividendTimes(1 To NoOfDividends) As Variant
    ReDim TmpCashDividends(1 To NoOfDividends) As Variant
```

[1] One exception is Wilmott, Dewynne, and Howison (1993, p. 399), who mention the problem and suggest to let the company go bankrupt if the dividend is larger than the asset price.

```
If CallPutFlag = "c" Then
    Binary = 1  ' // call option
ElseIf CallPutFlag = "p" Then
    Binary = -1 ' // put option
End If

dt = T / n
Df = Exp(-r * dt)
u = Exp(v * Sqr(dt))
d = 1 / u
uu = u^2
p = (Exp(r * dt) - d) / (u - d)

DividendAmount = CashDividends(1)

For i = 1 To NoOfDividends - 1 Step 1
    TmpCashDividends(i) = CashDividends(i + 1)
    TmpDividendTimes(i) = DividendTimes(i + 1) - DividendTimes(1)
Next

StepsBeforeDividend = Int(DividendTimes(1) / T * n)

ReDim StockPriceNode(1 To StepsBeforeDividend + 2) As Double
ReDim OptionValueNode(1 To StepsBeforeDividend + 2) As Double

StockPriceNode(1) = S * d^StepsBeforeDividend

For i = 2 To StepsBeforeDividend + 1 Step 1
    StockPriceNode(i) = StockPriceNode(i - 1) * uu
Next

'// Calculate option values for nodes time step just before dividend
For i = 1 To StepsBeforeDividend + 1 Step 1

    ValueNotExercising = BinomialDiscreteDividends(CallPutFlag, AmeEurFlag, _
StockPriceNode(i) - DividendAmount, X, T - DividendTimes(1), r, v, n _
- StepsBeforeDividend, TmpCashDividends, TmpDividendTimes)
     If AmeEurFlag = "a" Then
        OptionValueNode(i) = Max(ValueNotExercising, Binary _
        * (StockPriceNode(i) - X))
     ElseIf AmeEurFlag = "e" Then
        OptionValueNode(i) = ValueNotExercising
    End If
Next

'//Option values before dividend payment "standard binomial"
For j = StepsBeforeDividend To 1 Step -1
    For i = 1 To j + 1 Step 1
        StockPriceNode(i) = d * StockPriceNode(i + 1)
        If AmeEurFlag = "a" Then
            OptionValueNode(i) = Max((p * OptionValueNode(i + 1) _
    + (1 - p) * OptionValueNode(i)) _
    * Df, Binary * (StockPriceNode(i) - X))
        ElseIf AmeEurFlag = "e" Then
                OptionValueNode(i) = (p * OptionValueNode(i + 1) _
     + (1 - p) * OptionValueNode(i)) * Df
        End If
    Next
```

```
    Next

    BinomialDiscreteDividends = OptionValueNode(1)

End Function
```

9.3 BLACK'S METHOD FOR CALLS ON STOCKS WITH KNOWN DIVIDENDS

Black (1975) describes an approximation to the value of an American call on a dividend paying stock. This is basically the escrowed dividend method, where the stock price in the BSM formula is replaced with the stock price minus the present value of the dividend. To take into account the possibility of early exercise, one also computes an option value just before the dividend payment, without subtracting the dividend. Next, one takes the maximum of the two option values. The method has the same drawbacks as the escrowed dividend model, in addition to not optimally taking into account the value of early exercise possibilities.

9.4 THE ROLL, GESKE, AND WHALEY MODEL

Roll (1977), Geske (1979a), and Whaley (1981) developed a formula for the valuation of an American call option on a stock paying a single dividend of D, with time to dividend payout t. It has for a long time been considered a closed-form solution for American call options on dividend-paying stocks. However, as pointed out by Beneder and Vorst (2001), Haug, Haug, and Lewis (2003), and others, the model is based on the escrowed dividend price process and is seriously flawed, resulting in arbitrage opportunities among other problems.

$$C \approx (S - De^{-rt})N(b_1) + (S - De^{-rt})M\left(a_1, -b_1; -\sqrt{\frac{t}{T}}\right)$$
$$- Xe^{-rT}M\left(a_2, -b_2; -\sqrt{\frac{t}{T}}\right) - (X - D)e^{-rt}N(b_2), \qquad (9.4)$$

where

$$a_1 = \frac{\ln[(S - De^{-rt})/X] + (r + \sigma^2/2)T}{\sigma\sqrt{T}}$$
$$a_2 = a_1 - \sigma\sqrt{T}$$
$$b_1 = \frac{\ln[(S - De^{-rt})/I] + (r + \sigma^2/2)t}{\sigma\sqrt{t}}$$
$$b_2 = b_1 - \sigma\sqrt{t},$$

where $N(x)$ is the cumulative normal distribution function and $M(a, b; \rho)$ is the cumulative bivariate normal distribution function with upper integral limits a and b, and correlation coefficient ρ, as described in Chapter 13. I is the critical ex-dividend stock price I that solves

$$c(I, X, T-t) = I + D - X,$$

where $c(I, X, T-t)$ is the value of a European call with stock price I and time to maturity $T-t$. If $D \leq X(1-e^{-r(T-t)})$ or $I = \infty$, it will not be optimal to exercise the option before expiration, and the price of the American option can be found by using the BSM formula where the stock price is replaced with the stock price minus the present value of the dividend payment $S - De^{-rt}$.

This model was for many years considered a brilliant closed-form solution. As indicated above, the approach has considerable flaws that lead to significant arbitrage opportunities, and thus renders it more or less useless for all practical purposes. See Haug, Haug, and Lewis (2003) for more details on its shortcomings.

Example

Consider an American-style call option on a stock that will pay a dividend of 4 in exactly three months. The stock price is 80, the strike price is 82, time to maturity is four months, the risk-free interest rate is 6%, and the volatility is 30%. $S = 80$, $X = 82$, $t = 0.25$, $T = 0.3333$, $r = 0.06$, $D = 4$, $\sigma = 0.3$.

$$a_1 = \frac{\ln[(80 - 4e^{-0.06\times 0.25})/82] + (0.06 + 0.3^2/2)0.3333}{0.3\sqrt{0.3333}} = -0.2321$$

$$a_2 = a_1 - 0.3\sqrt{0.3333} = -0.4053$$

The critical stock price I solves

$$c(I, 82, 0.3333 - 0.25) = I + 4 - 82$$

The solution, given by a numerical search algorithm, is $I = 80.1173$. Moreover:

$$b_1 = \frac{\ln[(80 - 4e^{-0.06\times 0.25})/80.1173] + (0.06 + 0.3^2/2)0.25}{0.3\sqrt{0.25}} = -0.1715$$

$$b_2 = b_1 - 0.3\sqrt{0.25} = -0.3215$$

$$M\left(a_1, -b_1; -\sqrt{\frac{0.25}{0.3333}}\right) = 0.0703 \quad M\left(a_2, -b_2; -\sqrt{\frac{0.25}{0.3333}}\right) = 0.0632$$

$$N(b_1) = N(-0.1715) = 0.4319 \qquad N(b_2) = N(-0.3215) = 0.3739$$

$$C \approx (80 - 4e^{-0.06\times 0.25})N(b_1) + (80 - 4e^{-0.06\times 0.25})M\left(a_1, -b_1; -\sqrt{\frac{0.25}{0.3333}}\right)$$

$$- 82e^{-0.06\times 0.3333}M\left(a_2, -b_2; -\sqrt{\frac{0.25}{0.3333}}\right)$$

$$- (82 - 4)e^{-0.06\times 0.25}N(b_2) = 4.3860$$

The value of a similar European call is 3.5107.

Example of Arbitrage Opportunities

Consider the case of an initial stock price of 100, strike 130, risk-free rate 6%, volatility 30%, one year to maturity, and an expected dividend payment of seven in 0.9999 years. Using this input, the RGW model posits a value of 4.3007. Consider now another option, expiring just before the dividend payment, say, in 0.9998 years. Since this in effect is an American call on a non-dividend-paying stock, it is not optimal to exercise it before maturity. In the absence of arbitrage, the value must therefore equal the BSM price of 4.9183. This is, however, an arbitrage opportunity! The arbitrage occurs because the RGW model is misspecified, in that the dynamics of the stock price process depends on the timing of the dividend. Similar examples have been discussed by Beneder and Vorst (2001) and Frishling (2002). This is not just an esoteric example, as several well-known software systems use the RGW model and other similar misspecified models.

Computer algorithm

```
Function RollGeskeWhaley(S As Double, X As Double, t1 As Double, _
    T2 As Double, r As Double, d As Double, v As Double) As Double

    'tl time to dividend payout
    'T2 time to option expiration

    Dim Sx As Double, i As Double
    Dim a1 As Double, a2 As Double, b1 As Double, b2 As Double
    Dim HighS As Double, LowS As Double, epsilon As Double
    Dim ci As Double, infinity As Double

    infinity = 1000000000
    epsilon = 1e-08
    Sx = S - d * Exp(-r * t1)
    If d <= X * (1 - Exp(-r * (T2 - t1))) Then '// Not optimal to exercise
        RollGeskeWhaley = GBlackScholes("c", Sx, X, T2, r, r, v)
        Exit Function
    End If
    ci = GBlackScholes("c", S, X, T2 - t1, r, r, v)
    HighS = S
    While (ci - HighS - d + X) > 0 And HighS < infinity
        HighS = HighS * 2
```

```
        ci = GBlackScholes("c", HighS, X, T2 - t1, r, r, v)
    Wend
    If HighS > infinity Then
        RollGeskeWhaley = GBlackScholes("c", Sx, X, T2, r, r, v)
        Exit Function
    End If

    LowS = 0
    i = HighS * 0.5
    ci = GBlackScholes("c", i, X, T2 - t1, r, r, v)

    '// Search algorithm to find the critical stock price I
    While Abs(ci - i - d + X) > epsilon And HighS - LowS > epsilon
        If (ci - i - d + X) < 0 Then
            HighS = i
        Else
            LowS = i
        End If
        i = (HighS + LowS) / 2
        ci = GBlackScholes("c", i, X, T2 - t1, r, r, v)
    Wend
    a1 = (Log(Sx / X) + (r + v^2 / 2) * T2) / (v * Sqr(T2))
    a2 = a1 - v * Sqr(T2)
    b1 = (Log(Sx / i) + (r + v^2 / 2) * t1) / (v * Sqr(t1))
    b2 = b1 - v * Sqr(t1)

    RollGeskeWhaley = Sx * CND(b1) + Sx * CBND(a1, -b1, -Sqr(t1 / T2)) _
    - X * Exp(-r * T2) * CBND(a2, -b2, -Sqr(t1 / T2)) _
    - (X - d) * Exp(-r * t1) * CND(b2)

End Function
```

where $CND(\cdot)$ is the cumulative normal distribution function and $CBND(\cdot)$ is the cumulative bivariate normal distribution function described in Chapter 13.

Example: *RollGeskeWhaley*(80, 82, 0.25, 0.3333, 0.06, 4, 0.3) returns a call value of 4.3860 as in the numerical example above.

9.5 BENCHMARK MODEL FOR DISCRETE CASH DIVIDEND

This section is based on the paper by Haug, Haug, and Lewis (2003) (HHL). It describes what can be considered a benchmark model for the discrete dividend option valuation problem.

9.5.1 A Single Dividend

Let us start with a European-style or American-style equity option on a stock that pays a discrete dividend at time $t = t_D$. The simpler problem is to first specify a price process whereby any dividends are reinvested immediately back into the security—this is the so-called cum-dividend process S_t. In general, S_t is not the market price of the security, but instead is the market price of a hypothetical mutual fund that only invests in the security. To distinguish the concepts, we will write the market price of the security at time t as Y_t, which we will sometimes call the ex-dividend process. Of course, if there are no dividends, then $Y_t = S_t$ for all t. Even if the company pays a dividend, we can always arrange things so that $Y_0 = S_0$, which guarantees that $Y_t = S_t$ for all $t < t_D$.

The HHL model allows S_t to follow a very general continuous-time stochastic process. For example, they mention the following processes, written with dynamics under risk-adjusted probabilities (and therefore with an expected instantaneous rate of return equal to the risk-free rate. To keep things simple, they consider a world with a constant rate r).

Example (Cum-Dividend) Processes

(P1) GBM: $dS_t = rS_t\, dt + \sigma S_t\, dB_t$, where σ is a constant volatility and B is a standard Brownian motion.

(P2) Jump-diffusion: $dS_t = (r - \lambda k)S_t\, dt + \sigma S_t\, dB_t + S_t\, dJ_t$, where dJ_t is a Poisson-driven jump process with mean jump arrival rate λ and mean jump size k.

(P3) Jump-diffusion with stochastic volatility: $dS_t = (r - \lambda k)S_t\, dt + \sigma_t S_t\, dB_t + S_t dJ_t$, where σ_t follows its own separate, possibly correlated, diffusion or jump-diffusion.

Consider an option at time t, expiring at time T, and assume for a moment that there are no dividends so that $Y_t = S_t$ for all $t \le T$. In that case, clearly, models (P1) and (P2) are one-factor models: The option value $V(S_t, t)$ depends only upon the current state of one random variable. Model (P3) is a two-factor model, $V(S_t, \sigma_t, t)$. Obviously, "n-factor" models are possible in principle, for arbitrary n, and the Haug-Haug-Lewis treatment will apply to those, too.

Choosing a Dividend Policy

Consider the case where a company declares a single discrete dividend of size D, where the "ex-dividend date" is at time t_D. We consider an unprotected European-style option with time to expiration after time t_D, so that the option holder will not receive the dividend. Since option prices depend upon the market price of the security, we must now write $V(Y_t, t)$ for one-factor models.

Note that the company "declares" a dividend D. This means that it is the company's stated intention to pay the amount D *if that is possible*. When will it be impossible? It is natural to assume that the company cannot pay out more equity than exists. For simplicity, imagine a world where there are no distortions from taxes or other frictions, so that a dollar of dividends is valued the same as a dollar of equity. In such a world, if the company pays a dividend D, the stock price at the ex-dividend date must drop by the same amount: $Y(t_D) = Y(t_D^-) - D = S(t_D^-) - D$, where t_D^- is the time instantaneously before the ex-dividend date t_D. Since stock prices represent the price of a limited liability security, we must have $Y(t_D) \geq 0$, so the model is inconsistent if $S(t_D^-) < D$. The above models do not take this into account.

The HHL model uses the following minimal modification of the company's dividend policy. HHL assumes that the company will indeed pay out its declared amount D if $S^- > D$, abbreviating $S^- = S(t_D^-)$. However, in the case where $S^- < D$, the company is assumed to pay some *lesser* amount $\Delta(S^-)$ whereby $0 \leq \Delta(S^-) \leq S^-$. Numerical results for two natural policy choices, namely $\Delta(S^-) = S^-$ (liquidator), and $\Delta(S^-) = 0$ (survivor) are reported below. The first case allows liquidation because the ex-dividend stock price (at least in all of the sample models P1–P3 above) would be absorbed at zero. The second case (and, indeed, any model where $\Delta(S) < S$) allows survival because the stock price process can then attain strictly positive values after the dividend payment.

These choices, liquidation versus survival, sound dramatically different. In cases of financial distress, where indeed the stock price is very low, they would be. But such cases are relatively rare. As a practical matter, the choice of $\Delta(S)$ for $S < D$ has a negligible financial effect for most applications; the main point is that *some* choice must be made to fully specify the model. There is little financial effect in most applications because the probability that an initial stock price S_0 becomes as small as a declared dividend D is typically negligible.

To restate the idea in terms of a stochastic differential equation (SDE) for the security price process, the model now considers the actual dividend paid as the random variable $\mathcal{D}(S)$, where

$$\mathcal{D}(S) = \begin{cases} D, & \text{if } S > D \\ \Delta(S) \leq S, & \text{if } S \leq D \end{cases} \tag{9.5}$$

In (9.5) D is the declared (or projected) dividend—a constant, independent of S. The functional form for $\mathcal{D}(S)$ is any function that preserves limited liability. Then, the market price of the security evolves, using GBM as the prototype, as the SDE:

$$dY_t = \left[rY_t - \delta(t - t_D)\mathcal{D}(Y_{t_D^-})\right] dt + \sigma Y_t dB_t, \tag{P1a}$$

where $\delta(t - t_D)$ is Dirac's delta function centered at t_D. The same SDE drift modification occurs for (P2), (P3), or any other security price process you wish to model.

It's worth stressing that the Brownian motion B_t that appears in (P1) and (P1a) have *identical realizations*. You might want to picture a realization of B_t for $0 \le t \le T$. Your mental picture will ensure that $Y_t = S_t$ for all $t < t_D$ and $Y_{t_D} = S_{t_D} - \mathcal{D}(S_{t_D})$. Note that Y_t is completely determined by knowledge of S_t alone for all $t \le t_D$ (the fact that $Y_{t_D} = f(S_{t_D})$, where f is a deterministic function, will be crucial later).

To utilize this setup to value options, you need to be able to solve for the option value and the transition density for the cum-dividend stock price, $\phi(S_0, S_t, t)$, in the *absence* of dividends.[2] You need not have these functions in so-called "closed form," but merely have available some method of obtaining them. This method may be an analytic formula, a lattice method, a Monte Carlo procedure, a series solution, or whatever.

The Main Result

Now write $V_E(S_t, t; D, t_D)$ for the time-t fair value of a European-style option that expires at time T, in the presence of a discrete dividend D paid at time t_D. The last two arguments are the main parameters in the fully specified dividend policy $\{t_D, \mathcal{D}(S)\}$, where $t < t_D < T$. If there is no dividend between time t and the option expiration T, we simply drop the last two arguments and write $V_E(S_t, t)$. So, to be clear about notation, when you see an option value $V(\cdot)$ that has *only two arguments*, this will be a formula that you know in the absence of dividends, like the BSM formula. Again, the strike price X, option expiration T, and other parameters and state variables have been suppressed for simplicity. With this notation the main result is as follows:

The adoption by a company of a single discrete dividend policy $\{t_D, \mathcal{D}(S)\}$, causes the fair value of a European-style option to change from $V_E(S_0, 0)$ to $V_E(S_0, 0; D, t_D)$, where

$$V_E(S_0, 0; D, t_D) = e^{-rt_D} \int_0^\infty V_E(S - \mathcal{D}(S), t_D)\phi(S_0, S, t_D)\, \underset{\cdot}{S} \qquad (9.6)$$

For more details including mathematical proofs see Haug, Haug, and Lewis (2003).

Example

Take GBM, where the dividend policy is $\Delta(S) = S$ (liquidator) for $S \le D$. Then (9.6) for a call option becomes

$$C_E(S_0, 0; D, t_D) = e^{-rt_D} \int_D^\infty C_E(S - D, t_D)\phi(S_0, S, t_D)\underset{\cdot}{S} \qquad (9.7)$$

---

[2]The transition density is the probability density for an initial state (stock price plus other state variables) S_0 to evolve to the final state S_t in a time t. This evolution occurs under the risk-adjusted, cum-dividend process (or measure) such as the ones given under "Example (cum-dividend) processes" above. For GBM, $\phi(S_0, S_t, t)$ is the familiar lognormal density.

Note that the call price in the integrand of (9.6) is zero for $S - \mathcal{D}(S) = 0$ ($S \leq D$). In (9.7), $\phi(S_0, S, t)$ is simply the (no-dividend) lognormal density and $C_E(S - D, t_D)$ is simply the no-dividend BSM formula with time-to-go $T - t_D$. For example, suppose $S_0 = X = 100$, $T = 1$ (year), $r = 0.06$, $\sigma = 0.3$, and $D = 7$. Then consider two cases: (i) $t_D = 0.01$, and (ii) $t_D = 0.99$. We find from (9.7) the high precision results: (i) $C_E(100, 0; 7, 0.01) = 10.59143873835989$ and (ii) $C_E(100, 0; 7, 0.99) = 11.57961536099359$.

American-Style Options
It is well known that for an American-style call option with a discrete dividend, early exercise is only optimal instantaneously prior to the ex-dividend date Merton (1973). This result, of course, applies to the present model. Hence, to value an American-style call option with a single discrete dividend, you merely replace (9.6) with

$$C_A(S_0, 0; D, t_D) = e^{-rt_D} \int_0^\infty \max\{(S - X)^+, C_E(S - \mathcal{D}(S), t_D)\}\phi(S_0, S, t_D)S \qquad (9.8)$$

Early exercise is never optimal unless there is a finite solution S^* to $S^* - X = C_E(S^* - D, t_D)$, where we are assuming that $X > D$ (a virtual certainty in practice).

For American-style put options, as is also well known, it can be optimal to exercise at any time prior to expiration, even in the absence of dividends. So, in this case, you are generally forced to a numerical solution, evolving the stock price according to your model. This is the well-known backward iteration. What may differ from what you are used to is that you must allow for an instantaneous drop of $\mathcal{D}(S)$ on the ex-date.

9.5.2 Multiple Dividends

With the sequence of dividends $\{(D_i, t_i)\}_{i=1}^n$, $t_1 < t_2 < \ldots < t_n$, the argument behind formula (9.6) still holds. Simply repeat it iteratively, starting at time t_{n-1} by applying (9.6) to the last dividend (D_n, t_n). While straightforward, this procedure involves evaluating an n-fold integral. I therefore show a simpler way to compute it in the next section.

9.5.3 Applications

To illustrate the application of the pricing formula, we now specialize the option contracts as well as the stock price process.

European Call and Put Options The following put-call parity holds:

For a general cum-dividend price process S_t and dividend policy $\mathcal{D}(S)$ as in (9.5),

$$C_E(S_0, 0; D, t_D) + e^{-rT} X + e^{-rt_D}\bar{D} = P_E(S_0, 0; D, t_D) + S_0, \quad (9.9)$$

where

$$\bar{D} = D - \int_0^D \phi(S_0, S, t_D)(D - \Delta(S))dS$$

is the expected received dividend.

For the case of GBM stock price and liquidator dividend, $\Delta(S) = S$ for $S < D$, the value of a European call option can be written explicitly as

$$C_E(S_0, 0; D, t_D) = e^{-rt_D} \int_d^\infty C_E\left(S_0 e^{(r-\sigma^2/2)t_D + \sigma\sqrt{t_D}x} - D, t_D\right) \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} dx \quad (9.10)$$

$$d = \frac{\ln(D/S_0) - \left(r - \sigma^2/2\right) t_D}{\sigma\sqrt{t_D}}$$

A similar expression can be written down for the put option, but this is really not necessary in light of (9.9).

Tables 9-1 and 9-2 report option prices for European call options for small and large dividends. The tables use the symbols:

BSM is the plain vanilla Black–Scholes–Merton model.

M73 is the BSM model with $S - e^{-rt_D}D$ substituted for S—the escrowed dividend adjustment Merton (1973).

Vol1 is identical to M73, but with an adjusted volatility. The volatility of the asset is replaced with $\sigma_2 = \frac{\sigma S}{S - e^{-rt_D}D}$. See, for example, Chriss (1997).

Vol2 is a slightly more sophisticated volatility adjustment than Vol1, the Haug-Haug volatility adjustment.

Vol3 is the volatility adjustment suggested by Bos, Gairat, and Shepeleva (2003).

BV adjusts the strike and stock price, to take into account the effects of the discrete dividend payment (Bos and Vandermark, 2002).

Num is a non-recombining binomial tree with 500 time steps, and no adjustment to prevent the event that $S - D < 0$.

HHL(9.7) is the *exact* solution in (9.7).

TABLE 9-1

European Calls with Dividend of 7

($S = 100, T = 1, r = 6\%, \sigma = 30\%$)

t	BSM	Mer73	Vol1	Vol2	Vol3	BV	Num	HHL(9.7)
				$X = 100$				
0.0001	14.7171	10.5805	11.4128	10.5806	10.5806	10.5806	10.5829	10.5806
0.5000	14.7171	10.6932	11.5001	11.1039	11.0781	11.0979	11.1079	11.1062
0.9999	14.7171	10.8031	11.5855	11.5854	11.5383	11.5887	11.5704	11.5887
				$X = 130$				
0.0001	4.9196	3.0976	3.7403	3.0977	3.0977	3.0977	3.0987	3.0977
0.5000	4.9196	3.1437	3.7701	3.4583	3.4383	3.4159	3.4368	3.4383
0.9999	4.9196	3.1889	3.7993	3.7993	3.7616	3.7263	3.7140	3.7263
				$X = 70$				
0.0001	34.9844	28.5332	28.9113	28.5332	28.5332	28.5332	28.5343	28.5332
0.5000	34.9844	28.7200	29.0832	28.9009	28.8893	28.9350	28.9218	28.9215
0.9999	34.9844	28.9016	29.2504	29.2504	29.2286	29.3257	29.3140	29.3257

Table 9-1 illustrates that the M73 adjustment is inaccurate, especially in the case when the dividend is paid close to the option's

TABLE 9-2

European Calls with Dividend of 50

($S = 100, T = 1, r = 6\%, \sigma = 30\%$)

t	BSM	Mer73	Vol1	Vol2	Vol3	BV	Num	HHL(9.7)
				$X = 100$				
0.0001	14.7171	0.1282	2.9961	0.1283	0.1282	0.1283	0.1273	0.1283
0.5000	14.7171	0.1696	3.0678	1.4323	0.5755	0.8444	1.0687	1.0704
0.9999	14.7171	0.2192	3.1472	3.1469	1.1566	2.1907	2.1825	2.1908
				$X = 130$				
0.0001	4.9196	0.0094	1.3547	0.0094	0.0094	0.0094	0.0092	0.0094
0.5000	4.9196	0.0133	1.3556	0.4313	0.0947	0.1516	0.2264	0.2279
0.9999	4.9196	0.0184	1.3609	1.3607	0.2510	0.6120	0.6072	0.6120
				$X = 70$				
0.0001	34.9844	1.6510	7.0798	1.6517	1.6513	1.6514	1.6515	1.6517
0.5000	34.9844	1.9982	7.3874	4.9953	3.3697	4.2808	4.7304	4.7299
0.9999	34.9844	2.3780	7.7100	7.7096	4.9966	7.2247	7.2122	7.2248

expiration. Moreover, the Vol1 adjustment, often used by practitioners, yields significantly inaccurate values when the dividend is close to the beginning of the option's lifetime. Both Vol2 and BV do much better at accurately pricing the options. Vol3 yields values very close to the BV model. The non-recombining tree (Num) and the "exact" HHL solution (9.7) give very similar values in all cases. However, the non-recombining tree is not ensured to converge to the HHL solution (9.7) in all situations, unless the non-recombining tree is set up to prevent negative stock prices in the nodes where $S - D < 0$. This problem will typically be relevant only with a very high dividend. For low to moderate cash dividends, one can assume that even the "naive" non-recombining tree and the HHL solution agree to economically significant accuracy.

Table 9-2 shows that the BV and the non-recombining tree have significant differences when there's a significant dividend in the middle of the option's lifetime. The latter is closer to the true value. The Vol3 model strongly underprices the option when the dividend is this high.

American Call and Put Options Most traded stock options are American. Tables 9-3 to 9-5 offer a numerical comparison of stock options with a single cash dividend payment. The tables use the following models that differ from the European options considered above.

TABLE 9-3

American Calls with Dividend of 7

($D = 7, S = 100, T = 1, r = 6\%, \sigma = 30\%$)

t	B75	RGW	Num	HHL(9.8)
t		$X = 100$		
0.0001	10.5805	10.5805	10.5829	10.5806
0.5000	10.6932	11.1971	11.6601	11.6564
0.9999	14.7162	13.9468	14.7053	14.7162
		$X = 130$		
0.0001	3.0976	3.0976	3.0987	3.0977
0.5000	3.1437	3.1586	3.4578	3.4595
0.9999	4.9189	4.3007	4.9071	4.9189
		$X = 70$		
0.0001	30.0004	30.0004	30.0000	30.0004
0.5000	32.3034	32.3365	32.4604	32.4608
0.9999	34.9839	34.7065	34.9737	34.9839

TABLE 9-4

American Calls with Dividend of 30

($D = 30, S = 100, T = 1, r = 6\%, \sigma = 30\%$)

	B75	RGW	Num	HHL(9.8)
t		$X = 100$		
0.0001	2.0579	2.0579	2.0574	2.0583
0.5000	9.8827	7.5202	9.9296	9.9283
0.9999	14.7162	11.4406	14.7053	14.7162
		$X = 130$		
0.0001	0.3345	0.3345	0.3322	0.3346
0.5000	1.6439	0.6742	1.7851	1.7855
0.9999	4.9189	2.4289	4.9071	4.9189
		$X = 70$		
0.0001	30.0004	30.0004	30.0000	30.0004
0.5000	32.3034	32.0762	32.3033	32.3037
0.9999	34.9839	34.1637	34.9737	34.9839

B75 is the approximation to the value of an American call on a dividend-paying stock suggested by Black (1975). This is basically the escrowed dividend method, where the stock price in the

TABLE 9-5

American Calls with Dividend of 50

($D = 50, S = 100, T = 1, r = 6\%, \sigma = 30\%$)

	B75	RGW	Num	HHL(9.8)
t		$X = 100$		
0.0001	0.1282	0.1437	0.1273	0.1922
0.5000	9.8827	5.8639	9.8745	9.8828
0.9999	14.7162	9.3137	14.7053	14.7162
		$X = 130$		
0.0001	0.0094	0.0094	0.0092	0.0094
0.5000	1.6439	0.1375	0.5112	1.6492
0.9999	4.9189	1.1029	4.9071	4.9189
		$X = 70$		
0.0001	30.0004	30.0004	30.0000	30.0004
0.5000	32.3034	32.0762	32.6600	32.3034
0.9999	34.9839	34.1637	34.9737	34.9839

BSM formula is replaced with the stock price minus the present value of the dividend. To take into account the possibility of early exercise, one should also compute an option value just before the dividend payment, without subtracting the dividend. The value of the option is considered to be the maximum of these values.

RGW is the model of Roll (1977), Geske (1979a), and Whaley (1981). It is considered a closed-form solution for American call options on dividend-paying stocks. As we already know, the model is seriously flawed.

HHL is the exact solution in (9.8), again using the liquidator policy.

Table 9-3 shows that the RGW model works reasonably well when the dividend is in the very beginning of the option's lifetime. The RGW model exhibits the same problems as the simpler M73 or escrowed dividend method used for European options. The pricing error is particularly large when the dividend occurs at the end of the option's lifetime. The B75 approximation also significantly misprices options.

For a very high dividend, as in Table 9-5, the mispricing in the RGW formula is even more clear; the values are significantly off compared with both non-recombining tree (Num) and the exact HHL sclution (9.8). The simple B75 approximation is remarkably accurate. The intuition behind this is that a very high dividend makes it very likely to be optimal to exercise just before the dividend date—a situation where the B75 approximation for good reasons should be accurate.

Multiple Dividend Approximation

It is necessary to evaluate an n-fold integral in the HHL model when there are multiple dividends. It is therefore useful to have a fast, accurate approximation. We next show how to approximate the option value in the case of a call option on a stock whose cum-dividend price follows a GBM, using the liquidator dividend policy.

First, let's write the exact answer on date t with a sequence of n dividends prior to T as $C_n(S, X, t, T)$, where X is the strike and T is the expiration date. Then, the first iteration of (9.7) in an exact treatment becomes

$$C_1(S, X, t_{n-1}, T) = e^{-r(t_n - t_{n-1})} \int_{D_n}^{\infty} C_{\text{BSM}}(S_1 - D_n, X, t_n, T)\phi(S, S_1, t_n - t_{n-1})dS_1, \tag{9.11}$$

where $C_{\text{BSM}}(\cdot)$ is the BSM model. This integral is quick to evaluate, just as in the single dividend cases tabulated above. The second iteration becomes

$$C_2(S, X, t_{n-2}, T) = e^{-r(t_{n-1} - t_{n-2})} \int_{D_{n-1}}^{\infty} C_1(S_1 - D_{n-1}, X, t_{n-1}, T)\phi(S, S_1, t_{n-1} - t_{n-2})dS_1 \tag{9.12}$$

Notice that we now integrate not over the BSM model, but rather the option price derived in the first iteration (9.11). Evaluation of (9.12) therefore involves a double integral. We know, however, that $C_1(\cdot)$ will look like an option solution and hence will have many of the characteristics of the BSM formula. If we can effectively parameterize $C_1(\cdot)$ with a BSM formula, then it will be quick to evaluate (9.12).

Some key characteristics of $C_1(S, X, t_{n-1}, T)$ are as follows. First, it vanishes as $S \to 0$. Second, because (standard) put-call parity becomes asymptotically exact for large S,

$$C_1(S, X, t_{n-1}, T) \approx S - e^{-r(T-t_{n-1})}X - e^{-r(t_n - t_{n-1})}D_n$$

This suggests the BSM parameterization:

$$C_1(S, X, t_{n-1}, T) \approx C_{\text{BSM}}(S, X_{\text{adj}}, t_{n-1}, T), \tag{9.13}$$

where $X_{\text{adj}} = X + D_n e^{-r(t_n - T)}$. The strike adjustment ensures correct large-S behavior.

A little experimentation will show that the approximating BSM formula just suggested is inaccurate for S near the money. Still, we have another degree of freedom in our ability to adjust the volatility in the right-hand side of (9.13). By choosing σ_{adj} so that $C_1(S_0, X, t_{n-1}, T) \equiv C_{\text{BSM}}(S_0, X_{\text{adj}}, \sigma_{\text{adj}}, t_{n-1}, T)$, where S_0 is the original stock price of the problem, we obtain an accurate approximation

$$C_1(S, X, t_{n-1}, T) \approx C_{\text{BSM}}(S, X_{\text{adj}}, \sigma_{\text{adj}}, t_{n-1}, T)$$

that often differs by less than a penny over the full range of S on $(0, \infty)$.

This same scheme is then used at successive iterations of the exact integration. That is, the "previous" iteration will always be fast because it uses the BSM formula. Then after you get the answer, you approximate that answer by a BSM formula parameterization. In that parameterization, you choose an adjusted strike price and an adjusted volatility to fit the large-S behavior and the S_0 value. This enables you to move on to the next iteration.

Table 9-6 reports call option values when there is a dividend payment of 4 in the middle of each year. The first column shows the years to expiration for the contracts we consider. The models Vol2, Vol3, BV, and Num are identical to the ones described earlier. HHL is the closed-form solution from Section 2 evaluated by numerical quadrature. This approach is computer-intensive. The table therefore reports values of options with this method with up to three dividend payments. An efficient implementation in, for instance, C^{++} will naturally make this approach viable for any practical number of dividend payments. Non-recombining trees are even more computer-intensive, especially

TABLE 9-6

European Calls with Multiple Dividends of 4

($S = 100$, $X = 100$, $r = 6\%$, $\sigma = 25\%$, $D = 4$)

T	Num	Vol2	Vol3	BV	HHL	Appr	Adjusted strike	Adjusted volatility
1	10.6615	10.6585	10.6530	10.6596	10.6606	10.6606	104.122	0.2467
2	15.2024	15.1780	15.1673	15.1992	15.1989	15.1996	108.499	0.2421
3	18.5798	18.5348	18.5241	18.5981	18.5984	18.5998	113.146	0.2375
4	–	21.2297	21.2304	21.3592	–	21.3644	118.081	0.2328
5	–	23.4666	23.4941	23.6868	–	23.6978	123.320	0.2282
6	–	23.3556	25.4279	25.6907	–	25.7100	128.884	0.2237
7	–	26.9661	27.1023	27.4395	–	27.4695	–	–

for multiple dividends. They also entail problems with propagation of errors when the number of time steps is increased, and the table therefore reports option values for only up to three dividends (three years to maturity), with 500 time steps for $T = 1, 2$ and 1000 time steps for $T = 3$. The column Appr is the approximation just described. The two rightmost columns report the adjusted strike and voiatility used in this approximation method.

The approximation just suggested (Appr) is clearly very accurate when compared to the exact solution (HHL). The non-recombining binomial implementation (Num) of the spot process also yields results very close to HHL. Vol2 and Vol3 seem to give rise to significant mispricing with multiple dividends. The BV approximation seems somewhat more accurate. However, as we already know, it significantly misprices options when the dividend is very high. From a trader's perspective, the HHL-based models seem to be a clear choice—at least if you care about having a robust and accurate model that will work in "any" situation. Remember also that the HHL method is valid for *any* price process, including stochastic volatility, jumps, and other factors that can have a significant impact on pricing and hedging.

Exotic and Real Options on Dividend-Paying Stocks

There are a wide variety of exotic options that trade in the OTC equity market, and many are embedded in warrants and other complex equity derivatives (see Chapters 4 and 5). The HHL approach to options pricing in the presence of discrete dividends also holds in these cases. Many exotic options, in particular barrier options, are known to be very sensitive to stochastic volatility. Luckily, the model described above also holds for stochastic volatility, jumps, volatility term structure, as well as other factors that can be of vital importance when pricing exotic options. The HHL model should also be relevant to real options pricing, when the underlying asset offers known discrete payouts (of generic nature) during the lifetime of the real option.

9.6 OPTIONS ON STOCKS WITH DISCRETE DIVIDEND YIELD

If the stock price of a company increases the company is typically doing well and one can expect the dividend payout in terms of cash per stock to increase. On the other hand, if the stock price falls significantly, this typically indicates that the company is not doing well and that the company may reduce its dividend payout. Some option traders prefer taking this form of "discrete dividend yield" explicitly into account in the model, for long-lived equity options. If we assume the dividend payout is a fixed percentage of the stock price, we implicitly take this effect into account. I next present how to modify the BSM formula to take this into account, and also present a closed-form solution for American calls. Also presented is a recombining tree model that can be used to value a large number of options with discrete dividend yield.

9.6.1 European with Discrete Dividend Yield

To value standard European call or put options on a stock paying a discrete dividend yield, all we need to do is replace the stock price S with $S(1-\delta)$ in the Black and Scholes (1973) formula, where δ is the discrete dividend yield. In the case of multiple dividends, replace S with $S = S(1-\delta)^i$, where i is the number of dividends, or in the most general case with varying discrete dividend yield, we have $S = S(1-\delta_1)(1-\delta_2)\cdots(1-\delta_n)$.

9.6.2 Closed-Form American Call

Villiger (2005) suggests a closed-form solution for an American call on assets paying a discrete dividend yield δ. The solution is based on some of the same ideas as the Roll-Geske-Whaley formula, but in this case, due to discrete dividend yield versus absolute cash dividend, it does not contain the serious flaws of the Roll-Geske-Whaley formula.

$$C = SN(b_1) + (1-\delta)SM\left(a_1, -b_1; -\sqrt{\frac{t}{T}}\right)$$

$$- Xe^{-rT}M\left(a_2, -b_2; -\sqrt{\frac{t}{T}}\right) - Xe^{-rt}N(b_2), \qquad (9.14)$$

TABLE 9-7

Examples of American Calls with Discrete Dividend Yield Closed Form

($S = 100, X = 102, T = 0.5, r = 0.1$)

	$\sigma = 0.15$			$\sigma = 0.3$		
δ	$t = 0.1$	$t = 0.25$	$t = 0.4$	$t = 0.1$	$t = 0.25$	$t = 0.4$
1%	5.2072	5.2072	5.2397	9.3036	9.3036	9.3578
2%	4.6298	4.6300	5.0516	8.7333	8.7403	9.1232
5%	3.1468	3.7178	4.8766	7.1533	7.6473	8.7883
10%	1.9218	3.3357	4.8320	5.2984	6.8202	8.5928
15%	1.5620	3.2615	4.8297	4.3020	6.4667	8.5389
20%	1.4695	3.2509	4.8297	3.7852	6.3140	8.5266

where t is the time to dividend payment and T is the time to maturity and

$$a_1 = \frac{\ln[(1-\delta)S/X] + (r + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$a_2 = a_1 - \sigma\sqrt{T}$$

$$b_1 = \frac{\ln(S/I) + (r + \sigma^2/2)t}{\sigma\sqrt{t}}$$

$$b_2 = b_1 - \sigma\sqrt{t},$$

where $N(x)$ is the cumulative normal distribution function, and $M(a, b; \rho)$ is the cumulative bivariate normal distribution function with upper integral limits a and b, and correlation coefficient ρ, as described in Chapter 13. I is the critical ex-dividend stock price that solves

$$c(I(1-\delta), X, T-t) = I - X,$$

where $c(I(1-\delta), X, T-t)$ is the value of a European call with stock price $I(1-\delta)$ and time to maturity $T-t$.

Table 9-7 gives values of American call options on a stock paying discrete dividend yield, for different choices of dividend yield δ, time to dividend payment t, and volatility σ.

Computer algorithm

Here I present the VBA code for an American call on a stock paying a single discrete dividend yield. The method can be extended to a multiple dividends.

```
Function DiscreteDividenYieldAnalytic(EurAmeFlag As String, S As Double, _
X As Double, t1 As Double, T2 As Double, _
```

```
r As Double, Dy As Double, v As Double) As Double
    't1 time to dividend payout
    'T2 time to option expiration

   Dim Sx As Double, i As Double
   Dim a1 As Double, a2 As Double, b1 As Double, b2 As Double
   Dim HighS As Double, LowS As Double, epsilon As Double
   Dim ci As Double, infinity As Double

   infinity = 1000000000
   epsilon = 1e-08
   Sx = S * (1 - Dy)
   If EurAmeFlag = "e" Or S * (1 - Dy) <= X * (1 - Exp(-r * (T2 - t1))) Then
   '// Not optimal to exercise
       DiscreteDividenYieldAnalytic = GBlackScholes("c", Sx, X, T2, r, r, v)
       Exit Function
   End If
   ci = GBlackScholes("c", S, X, T2 - t1, r, r, v)
   HighS = S
   While (ci - HighS + X) > 0 And HighS < infinity
       HighS = HighS * 2
       ci = GBlackScholes("c", HighS * (1 - Dy), X, T2 - t1, r, r, v)
   Wend
   If HighS > infinity Then
       DiscreteDividenYieldAnalytic = GBlackScholes("c", Sx, X, T2, r, r, v)
       Exit Function
   End If

   LowS = 0
   i = HighS * 0.5
   ci = GBlackScholes("c", i * (1 - Dy), X, T2 - t1, r, r, v)

   '// Search algorithm to find the critical stock price I
   While Abs(ci - i + X) > epsilon And HighS - LowS > epsilon
       If (ci - i + X) < 0 Then
           HighS = i
       Else
           LowS = i
       End If
       i = (HighS + LowS) / 2
       ci = GBlackScholes("c", i * (1 - Dy), X, T2 - t1, r, r, v)
   Wend

   a1 = (Log(Sx / X) + (r + v^2 / 2) * T2) / (v * Sqr(T2))
   a2 = a1 - v * Sqr(T2)
   b1 = (Log(S / i) + (r + v^2 / 2) * t1) / (v * Sqr(t1))
   b2 = b1 - v * Sqr(t1)

   DiscreteDividenYieldAnalytic = Sx * CBND(a1, -b1, -Sqr(t1 / T2)) _
   - X * Exp(-r * T2) * CBND(a2, -b2, -Sqr(t1 / T2)) _
   + S * CND(b1) - X * Exp(-r * t1) * CND(b2)

 End Function
```

Example

To price an American call option with stock price 100, strike 102, time to maturity six months, risk-free rate equal to 10%, volatility

15%, time to dividend payment, three months, and proportional dividend yield of 5%: *DiscreteDividenYieldAnalytic*("*a*", 100, 102, 0.25, 0.5, 0.1, 0.05, 0.15). This will return an American call option value of 3.7178.

9.6.3 Recombining Tree Model

A binomial tree can be used to price options on a stock that at certain points in time pays a known dividend yield. Before the stock goes ex-dividend, the stock price at each node is set equal to

$$Su^i d^{j-i}, \qquad i = 0, 1, \ldots, j$$

After the stock goes ex-dividend, the stock price corresponds to

$$S(1 - \delta_j)u^i d^{j-i}, \qquad i = 0, 1, \ldots, j,$$

where δ_j is the total dividend yield from all ex-dividend dates between time zero and the relevant time step in the binomial tree. The binomial tree will still be recombining.

If we choose to use the Cox, Ross, and Rubinstein (1979) parameters, we have

$$u = e^{\sigma\sqrt{\Delta t}}, \qquad d = e^{-\sigma\sqrt{\Delta t}},$$

where $\Delta t = T/n$ is the size of each time step and n is the number of time steps. The probability of the stock price increasing at the next time step is

$$p = \frac{e^{b\Delta t} - d}{u - d}$$

The probability of going down must be $1 - p$, since the probability of going either up or down equals unity.

Table 9-8 shows values from the discrete dividend binomial tree using 500 time steps, with the same input parameters as in Table 9-7.

Computer algorithm

The code returns a column array of option value, delta, gamma, and theta, taking into account discrete dividend yields. The years to maturity of the dividends is put into "*DividendTimes*". Similarly, the corresponding dividend yields are going into "*Dividends*". The tree is fully recombining and very efficient.

```
Function DiscreteDividendYield(AmeEurFlag As String, CallPutFlag As String, _
  S As Double, X As Double, T As Double,  r As Double, v As Double, _
  n As Integer, DividendTimes As Object, Dividends As Object) As Variant
```

TABLE 9-8

Examples of American Option Values from Discrete Dividend Yield Binomial Tree.

($S = 100, X = 102, T = 0.5, r = 0.1, n = 500$)

	$\sigma = 0.15$			$\sigma = 0.3$		
δ	$t = 0.1$	$t = 0.25$	$t = 0.4$	$t = 0.1$	$t = 0.25$	$t = 0.4$
			Call options			
1%	5.2065	5.2065	5.2391	9.3072	9.3072	9.3615
2%	4.6298	4.6299	5.0516	8.7320	8.7389	9.1231
5%	3.1474	3.7190	4.8772	7.1546	7.6493	8.7899
10%	1.9192	3.3363	4.8311	5.2965	6.8194	8.5965
15%	1.5647	3.2614	4.8283	4.2979	6.4655	8.5387
20%	1.4693	3.2494	4.8283	3.7785	6.3180	8.5238
			Put options			
1%	4.1048	3.8987	3.7610	8.0053	7.9105	7.7738
2%	4.6827	4.3741	4.0454	8.4898	8.3658	8.1149
5%	7.0000	7.0000	7.0000	10.0936	9.9051	9.5369
10%	12.0000	12.0000	12.0000	13.2590	12.9060	12.3760
15%	17.0000	17.0000	17.0000	17.0421	17.0000	17.0000
20%	22.0000	22.0000	22.0000	22.0000	22.0000	22.0000

```
Dim ReturnValue() As Double
Dim StepsDividend() As Double
Dim St() As Double
Dim OptionValue() As Double        'Option Value at each node
Dim i As Integer, j As Integer, m As Integer, z As Integer
Dim nDividends As Integer
Dim Df As Double, dt As Double, p As Double
Dim u As Double, uu As Double, d As Double, SumDividends As Double

nDividends = Application.Count(DividendTimes)

If nDividends = 0 Then
    DiscreteDividendYield = CRRBinomial(AmeEurFlag, CallPutFlag, _
      S, X, T, r, r, v, n)
    Exit Function
End If

ReDim ReturnValue(0 To 3)
ReDim StepsDividend(0 To nDividends)
ReDim St(0 To n + 2)
ReDim OptionValue(0 To n + 2)

dt = T / n  '//Size of time step
Df = Exp(-r * (T / n))  '// Discount factor
u = Exp(v * Sqr(T / n))
 d = 1 / u
uu = u ^ 2
p = (Exp(r * dt) - d) / (u - d)  ' // Up probability

z = 1 '// call
If CallPutFlag = "p" Then
    z = -1 '// put
```

```
    End If

    SumDividends = 1

    For i = 0 To nDividends - 1
        StepsDividend(i) = Int(DividendTimes(i + 1) / T * n)
        SumDividends = SumDividends * (1 - Dividends(i + 1))
    Next

    For i = 0 To n '// Option value at expiry
        St(i) = S * u^i * d^(n - i) * SumDividends
        OptionValue(i) = Max(z * (St(i) - X), 0)
    Next

    For j = n - 1 To 0 Step -1
        For m = 0 To nDividends
            If j = StepsDividend(m) Then
                For i = 0 To j
                    St(i) = St(i) / (1 - Dividends(m + 1))
                Next i
            End If
        Next m
        For i = 0 To j
            St(i) = d * St(i + 1)
            '//European value:
            OptionValue(i) = (p * OptionValue(i + 1) _
              + (1 - p) * OptionValue(i)) * Df
           If AmeEurFlag = "a" Then '// American value
                OptionValue(i) = Max(OptionValue(i), z * (St(i) - X))
           End If
        Next i

          If j = 2 Then
          '// Gamma
                ReturnValue(2) = ((OptionValue(2) - OptionValue(1)) _
                / (S * u^2 - S) - (OptionValue(1) - OptionValue(0)) _
                / (S - S * d^2)) / (0.5 * (S * u^2 - S * d^2))
          '// Part of theta
               ReturnValue(3) = OptionValue(1)
            End If

         If j = 1 Then
         '// Delta
               ReturnValue(1) = (OptionValue(1) _
                 - OptionValue(0)) / (S * u - S * d)
            End If

    Next j

    ReturnValue(0) = OptionValue(0)
    ' // One day theta:
    ReturnValue(3) = (ReturnValue(3) - OptionValue(0)) / (2 * dt) / 365
    DiscreteDividendYield = Application.Transpose(ReturnValue())

End Function
```

CHAPTER

10

COMMODITY AND ENERGY OPTIONS

A practical option formula should be as simple as possible, but not simpler than that.

The Author

Most commodity and energy options can be valued with the formulas already described in this book. The most popular commodity option model is actually the Black-76 model described in Chapter 1. This is the benchmark model used for most European options on commodity and energy futures. Also, many of the numerical methods described in earlier chapters can be applied to commodity and energy derivatives. For example, in the case of American futures options, the binomial or trinomial model is popular, and Monte Carlo methods can easily be used to take mean reversion into account. As in most other markets, taking jumps and stochastic volatility into account can be of great importance. Valuing commodity and energy options is thus not very different from valuing options in other markets. The difference is more in what input to feed into the model. This chapter briefly presents some of the adjustments that are particular to energy and commodity valuation.

10.1 ENERGY SWAPS/FORWARDS

Oil and electricity swaps are actively traded in the energy markets. To find the fair value of an electricity forward or swap, we need to know the cost of storing (hydropower can to some degree be stored in water reservoirs) and producing electricity. These costs can vary among different operators in the market and the topic is outside the scope of this book. Given the presence of traded contracts with quoted market prices—for instance, a swap (forward)—we can come up with a way to value the swap relative to other swaps. For example, a strip of quarterly power swaps covering the whole year should have the same value as an annual contract. Otherwise, there will be an arbitrage opportunity. Oil swaps are similar and typically have a daily fixing against NYMEX.

The Nordic Electricity Exchange (Nord Pool) lists actively traded electricity swaps. The electricity swaps traded in the Nordic power market are known as forwards but are from a valuation perspective power swaps, a strip of one-day electricity forwards. To compare the value of different swaps, we need to discount the cash flows. The swap/forward price is not the value of the swap contract, but only the contract price. To compare different power swaps with each other, we need to find the value (Haug, 2005a):

$$F_{ValueToday} = \frac{e^{-r_b T_b}}{n} \sum_{i=1}^{n} \frac{F}{(1 + r_{j,i}/j)^i}, \tag{10.1}$$

where $F_{ValueToday}$ is the swap value today and

- F is the forward/swap price in the market. In this case, “price” should not be confused with “value”!
- j is the number of compoundings per year (number of settlements in a one-year forward contract). We assume here they are evenly spread out. In practice, there are no payments during weekends, so every fifth payment does not have the same time interval as the rest of the payments. However, the effect of taking this into account is not of economic significance, at least for monthly or longer contracts.
- n is the number of settlements in the delivery period for the particular forward contracts. Nord Pool uses daily settlement, so this will typically be the number of trading days in the forward period.
- $r_{j,i}$ is a risk-free interest swap rate starting at the beginning of the delivery period and ending at the i period. Further, it has j compoundings per year.
- T_b is the time to the beginning of the forward delivery period.
- r_b is a risk-free continuously compounded zero coupon rate with T_b years to maturity.

In the case where we assume a constant interest rate in the delivery period, we can simplify the swap value formula to

$$F_{ValueToday} = Fe^{-r_b T_b} \frac{\left(1 - \frac{1}{(1+r_j/j)^n}\right)}{r_j} \frac{j}{n}, \tag{10.2}$$

where r_j now is the forward start swap rate, starting at the beginning of the delivery period and ending at the end of the delivery period, with j compoundings per year set equal to the number of fixings per year.

Example
Consider a quarterly electricity forward that trades at a price of 35 EUR/MwH (EUR per mega watt hour), the delivery period is 2160 hours, or 90 days. It is six months to the start of the delivery period. Assume the forward start swap rate, starting six months from now and ending six months plus 90 days from now, is 5% converted to the basis of daily compounding. The six-month continuous zero coupon rate is 4%. What is the present value of the power contract when using 365 days per year? $F = 35$, $r_j = 0.05$, $j = 365$, $n = 90$, $r_b = 0.04$, and $T_b = 0.5$.

$$F_{ValueToday} = 35e^{-0.04\times 0.5} \times \frac{\left(1 - \frac{1}{(1+0.05/365)^{90}}\right)}{0.05} \times \frac{365}{90} = 34.0940$$

The present value of the power forward/swap is thus 34.0940 EUR per MwH. The total value of one contract is found by multiplying the number of hours in the contract period by the value per MwH $2160 \times 34.0940 = 73{,}643.08$ EUR.

Approximation
Formula (10.2) can be approximated by

$$F_{ValueToday} \approx Fe^{-r_b T_b} e^{-r_d (T_m - T_b)},$$

where r_d is the forward starting continuously compounded zero coupon rate for the delivery period, multiplied by the time from the start of the delivery period T_b to the middle of the delivery period T_m. This can be simplified further by

$$F_{ValueToday} \approx Fe^{-r_e T_m}, \tag{10.3}$$

where r_e is a continuously compounded zero coupon rate from now to the end of the delivery period. This approximation is reasonably accurate as long as we use consistent rates.

Example
Consider the same input as in the last example. To make the examples equivalent, we have to find a rate r_e that is consistent with $r_b = 0.04$ and $r_j = 0.05$. To find r_e from the example above, we first need to convert r_j to a continuously compounded rate

$r_{cj} = 365 \ln(1 + 0.05/365) = 0.04999658$. We know that

$$e^{-r_e T_m} = e^{-r_b T_b} e^{-r_{cj}(T_m - T_b)}$$

$$r_e = \frac{r_b T_b + r_{cj}(T_m - T_b)}{T_b + (T_m - T_b)}$$

$$r_e = \frac{0.04 \times 0.5 + 0.049997 \times 90/365/2}{0.5 + 90/365/2} = 0.04198,$$

and we can now approximate the value of the forward price

$$F_{ValueToday} \approx 35e^{-0.04198 \times (0.5 + 90/365/2)} = 34.0961$$

The approximate value is thus not very different from 34.0940 calculated by the more accurate formula.

10.2 ENERGY OPTIONS

10.2.1 Options on Forwards, Black-76F

Traders in commodity markets often use the Black-76 model to value options on commodity futures. When it comes to commodity options on forwards, the Black-76 formula holds only for the case when the forward contract expires at the same time as the option contract T. In the case where there is delivery of a forward contract that has a different expiration date, one has only locked in the payoff from the option but will receive the intrinsic value first at the forward's expiration. The Black-76 formula has to be adjusted for this effect. Thus, if you are long an in-the-money call option and you exercise, you will receive a forward contract that expires at time T_f ($T_f > T$), with a forward delivery price set equal to the strike of the option, X. To lock in the intrinsic value, you must sell a forward in the market at market price. You will, however, receive the money from this transaction first when the forward contract expires at time T_f. We get a modified Black-76 formula that we will name Black-76F:

$$c = e^{-rT_f}[FN(d_1) - XN(d_2)] \tag{10.4}$$

$$p = e^{-rT_f}[XN(-d_2) - FN(-d_1)], \tag{10.5}$$

where

$$d_1 = \frac{\ln(F/X) + (\sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = \frac{\ln(F/X) - (\sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T}$$

Example

Consider a European option on the Brent Blend forward that expires in 12 months, with 9 months to expiration. The forward price is USD 19, the strike price is USD 19, the risk-free interest rate is 10% per year, and the volatility is 28% per year. $F = 19, X = 19, T = 0.75, T_f = 1, r = 0.1$, and $\sigma = 0.28$. Thus:

$$d_1 = \frac{\ln(19/19) + (0.28^2/2)0.75}{0.28\sqrt{0.75}} = 0.1212$$

$$d_2 = d_1 - 0.28\sqrt{0.75} = -0.1212$$

$$N(d_1) = N(0.1212) = 0.5483 \qquad N(d_2) = N(-0.1212) = 0.4517$$

$$N(-d_1) = N(-0.1212) = 0.4517 \qquad N(-d_2) = N(0.1212) = 0.5483$$

$$c = e^{-0.1\times 1}[19N(d_1) - 19N(d_2)] = 1.6591$$

$$p = e^{-0.1\times 1}[19N(-d_2) - 19N(-d_1)] = 1.6591$$

10.2.2 Energy Swaptions

European options on energy swaps, also called energy swaptions, are options that at maturity give a delivery of an energy swap at the strike price (but not necessarily physical delivery of any energy). The swap can have either financial or physical settlement. For example, the most popular options trading at the Nordic Power Exchange, Nord Pool, are in reality energy swaptions. If a call swaption is in-the-money at maturity, the option has delivery of a swap. The payout[1] from the option is thus not received immediately at expiration, but rather during the delivery period of the underlying swap (forward). For example, at Nord Pool there is financial daily settlement in the delivery period of the swap/forward against the daily settlement (auction price) of

[1]Typically named forward by the market participants.

the underlying physical market. The energy call swaption formula is (see Haug, 2005a)

$$c = \frac{\left(1 - \frac{1}{(1+r_j/j)^n}\right)}{r_j} \frac{j}{n} e^{-r_p(T_b - T)} \times \text{Black-76}$$

$$= \frac{\left(1 - \frac{1}{(1+r_j/j)^n}\right)}{r_j} \frac{j}{n} e^{-r_p(T_b - T)} e^{-rT} [FN(d_1) - XN(d_2)]$$

$$= \frac{\left(1 - \frac{1}{(1+r_j/j)^n}\right)}{r_j} \frac{j}{n} e^{-r_b T_b} [FN(d_1) - XN(d_2)], \tag{10.6}$$

where r_p is the risk-free rate from the option's expiration to the beginning of the delivery period. T_b is the time from now to the beginning of the delivery period. Moreover:

$$d_1 = \frac{\ln(S/X) + \sigma^2 T/2}{\sigma\sqrt{T}}, \qquad d_2 = d_1 - \sigma\sqrt{T}$$

For a put we similarly have

$$p = \frac{\left(1 - \frac{1}{(1+r_j/j)^n}\right)}{r_j} \frac{j}{n} e^{-r_b T_b} [XN(-d_2) - FN(-d_1)] \tag{10.7}$$

Good approximations for calculating the call and the put swaption values are

$$c \approx e^{-r_e T_m} [FN(d_1) - XN(d_2)] \tag{10.8}$$

$$p \approx e^{-r_e T_m} [XN(-d_2) - FN(-d_1)], \tag{10.9}$$

where r_e is the risk-free rate from now to the end of the delivery period and

$$d_1 = \frac{\ln(S/X) + \sigma^2 T/2}{\sigma\sqrt{T}} \qquad d_2 = d_1 - \sigma\sqrt{T},$$

and T_m is the time in years from now to the middle of the delivery period.

As there is a lot of notation here, following are summaries of the notation used for this option formula:

- F is the forward/swap price observed in the market. In this case price should not be confused with value!
- j is the number of compoundings per year (number of settlements in a one-year forward contract). Here we assume evenly spread

out. In practice, no payments are made during the weekend, so every fifth payment is not even. However, the effect of taking this into account is not of practical importance, at least if we have to deal with monthly or longer contracts.

n is the number of settlements in the delivery period for the particular forward contracts. In Nord Pool there is daily settlement, so this will be the number of trading days in the forward period.

r_j is a swap rate starting at the beginning of the delivery period and ending at the end of the delivery period with j compoundings per year, equal to the number of fixings in the delivery period.

T_b is the time to the beginning of the forward delivery period.

r_b is a risk-free continuous compounding zero coupon rate with T_b years to maturity.

r_e is a risk-free continuous compounding zero coupon rate with time to maturity equal to from now to the end of the delivery period.

r_p is a risk-free continuous compounding zero coupon rate with forward start at the option maturity T and ending at the beginning of the delivery period T_b.

Example

Consider a call on a quarterly electricity swap, with six months to maturity. The start of the delivery period is 17 days after the option expires, and the delivery period is 2208 hours, or 92 days. The swap/forward trades at 33 EUR/MwH, and the strike is 35 EUR/MwH. The number of fixings in the delivery period is 92. The risk-free rate from now until the beginning of the delivery period is 5%. The daily compounding swap rate starting at the beginning of the delivery period and ending at the end of the delivery period is 5%. The volatility of the swap is 18%. What is the option value? $T = 0.5$, $T_b = 0.5 + 17/365 = 0.5466$, $r_b = 0.05$, $r_j = 0.05$, $j = 365$, $n = 92$, and $\sigma = 0.18$ yields

$$d_1 = \frac{\ln(33/35) + 0.5 \times 0.18^2/2}{0.18\sqrt{0.5}} = -0.3987$$

$$d_2 = -0.3987 - 0.18\sqrt{0.5} = -0.5259$$

$$N(d_1) = N(-0.3987) = 0.3451 \qquad N(d_2) = N(-0.5259) = 0.2995$$

$$c = \frac{\left(1 - \frac{1}{(1+0.05/365)^{92}}\right)}{0.05} \frac{365}{92} e^{-0.05\times 0.5466}[33N(d_1) - 35N(d_2)] = 0.8761$$

To find the value of an option on one swap/forward contract, we need to multiply by the number of delivery hours. This yields a price of $2208 \times 0.8761 = 1{,}934.37$ EUR. Alternatively, we could have found the option value using the approximation (10.8), using time from now to the middle of delivery period $T_m = 0.5 + 17/365 + 92/2/365 = 0.6726$ and assuming the rate from now to the end of the delivery period is $r_e \approx 0.05$:

$$c \approx e^{-0.05\times 0.6260}[33N(d_1) - 35N(d_2)] = 0.8761$$

At four-decimals accuracy, the approximation evidently gives the same result as the more accurate formula.

Put-Call Parity

For a standard put, also called a receiver swaption, or call option, also called a payer swaption, the put-call parity is

$$p = c + (X - F)\frac{\left(1 - \frac{1}{(1+r_j/j)^n}\right)}{r_j}\frac{j}{n}e^{-r_b T_b} \tag{10.10}$$

and, of course,

$$c = p + (F - X)\frac{\left(1 - \frac{1}{(1+r_j/j)^n}\right)}{r_j}\frac{j}{n}e^{-r_b T_b} \tag{10.11}$$

The put-call parity can be used to construct synthetic puts or calls as well as synthetic swaps/forwards from traded puts and calls. The synthetic/implied forward price from a put and a call is given by

$$F = \frac{c - p}{\frac{\left(1 - \frac{1}{(1+r_j/j)^n}\right)}{r_j}\frac{j}{n}e^{-r_b T_b}} + X$$

Energy Swaption Greeks

Delta:

$$\Delta_{call} = \frac{\left(1 - \frac{1}{(1+r_j/j)^n}\right)}{r_j}\frac{j}{n}e^{-r_b T_b}N(d_1) \tag{10.12}$$

$$\Delta_{put} = -\frac{\left(1 - \frac{1}{(1+r_j/j)^n}\right)}{r_j}\frac{j}{n}e^{-r_b T_b}N(-d_1) \tag{10.13}$$

Vega:

The vega is the swaption's sensitivity with respect to change in volatility.

$$\text{Vega}_{call,put} = \frac{\left(1 - \frac{1}{(1+r_j/j)^n}\right)}{r_j}\frac{j}{n}e^{-r_b T_b}Fn(d_1)\sqrt{T} \tag{10.14}$$

It is necessary to divide by 100 to express vega as the change in the option value for a 1% point change in volatility.

Gamma:
Gamma for swaptions:

$$\Gamma_{call,put} = \frac{\frac{\left(1-\frac{1}{(1+r_j/j)^n}\right)}{r_j}\frac{j}{n}e^{-r_b T_b}n(d_1)}{F\sigma\sqrt{T}} \tag{10.15}$$

Rho:

$$\rho_{call} = \left(\frac{(1+r_j/j)^{-n-1}}{r_j} - \frac{\left(1-\frac{1}{(1+r_j/j)^n}\right)}{r_j^2}\frac{j}{n} - T\frac{\left(1-\frac{1}{(1+r_j/j)^n}\right)}{r_j}\frac{j}{n}\right) \times e^{-r_b T_b}[FN(d_1) - XN(d_2)] \tag{10.16}$$

$$\rho_{put} = \left(\frac{(1+r_j/j)^{-n-1}}{r_j} - \frac{\left(1-\frac{1}{(1+r_j/j)^n}\right)}{r_j^2}\frac{j}{n} - T\frac{\left(1-\frac{1}{(1+r_j/j)^n}\right)}{r_j}\frac{j}{n}\right) \times e^{-r_b T_b}[XN(-d_2) - FN(-d_1)] \tag{10.17}$$

10.2.3 Hybrid Payoff Energy Swaptions

In certain markets there are options that pay off cash at expiration and also give physical delivery of the underlying swap at close (or alternatively its settlement price) on option expiration. The option thus has payoff similar to an option on a futures contract, but in addition, delivery of a swap equal to the settlement (fixing) price at expiration. This "hybrid" used to be the standard for electricity options trading at Nord Pool until they changed delivery to conform to the standard energy swaption style. This option may seem difficult to value at first. Assuming geometric Brownian motion of the swap/forward price, the value is simply the Black-76 value, however. Still, since the underlying asset is a swap and not a futures contract, the delta hedge is not the Black-76 delta but has to be adjusted for the fact that the underlying swap has a payoff over the delivery period, after the option expires. Letting $\Delta_{B76} = e^{-rT}N(d_1)$ be the Black-76 delta, the delta for hedging with electricity forwards/swaps is given by Haug (2005):

$$\Delta_{B76}F = \Delta_H F_{VaT}$$

$$\Delta_{B76}F = \Delta_H F e^{-r_p(T_b-T)} \frac{\left(1 - \frac{1}{(1+r_j/j)^n}\right)}{r_j} \frac{j}{n}$$

$$\Delta_H = N(d_1)e^{-rT} e^{r_p(T_b-T)} \frac{nr_j}{j\left(1 - \frac{1}{(1+r_j/j)^n}\right)}$$

$$\Delta_H \approx N(d_1)e^{-rT} e^{r(T_b-T)} \frac{nr_j}{j\left(1 - \frac{1}{(1+r_j/j)^n}\right)},$$

where F_{VaT} is the swap value at option expiration T, r is the continuous risk-free zero coupon rate until option expiration, and r_p is the continuous compounding zero coupon rate from option expiration until beginning of the delivery period of the swap.

$$d_1 = \frac{\ln(F/X) + T\sigma^2/2}{\sigma\sqrt{T}}$$

and for a put

$$\Delta_{put} = N(-d_1)e^{-rT} e^{r_p(T_b-T)} \frac{nr_j}{j\left(1 - \frac{1}{(1+r_j/j)^n}\right)} \tag{10.18}$$

10.3 THE MILTERSEN-SCHWARTZ MODEL

Miltersen and Schwartz (1998) developed an advanced model for pricing options on commodity futures. The model is a three-factor model with stochastic futures price, a term structure of convenience yields,[2] and interest rates.[3] The model assumes commodity prices are lognormally distributed and that continuously compounded forward interest rates and future convenience yields are normally distributed (aka Gaussian).

Investigations using this option pricing model show that the time lag between the expiration on the option and the underlying futures

[2]The convenience yield can be seen as the benefit or cost that accrues to the owner of the physical commodity but not to the owner of a contract for delivery in the future. It causes the expected price change of the commodity to be different from the expected return from investing in it.

[3]The Miltersen and Schwartz (1998) model can be seen as a generalization of previous work by Merton (1973), Gibson and Schwartz (1990), Amin and Jarrow (1992), Reismann (1992), Cortazar and Schwartz (1994), Amin, Ng, and Pirrong (1995), and Schwartz (1997).

will have a significant effect on the option value. Even with three stochastic variables, Miltersen and Schwartz manage to derive a closed-form solution similar to a BSM-type formula. The model can be used to price European options on commodity futures.[4]

$$c = P_t[F_T e^{-\sigma_{xz}} N(d_1) - XN(d_2)], \tag{10.19}$$

where t is the time to maturity of the option, F_T is a futures price with time to expiration T ($T \geq t$), and P_t is a zero-coupon bond that expires on the option's maturity.

$$d_1 = \frac{\ln(F_T/X) - \sigma_{xz} + \sigma_z^2/2}{\sigma_z}, \qquad d_2 = d_1 - \sigma_z,$$

and the variances and covariance can be calculated as[5]

$$\sigma_z^2 = \int_0^t \left\| \sigma_S(u) + \int_u^T [\sigma_f(u,s) - \sigma_\epsilon(u,s)]ds \right\|^2 du = \int_0^t \|\sigma_{F_T}(u)\|^2 du$$

$$\begin{aligned}\sigma_{xz} &= \int_0^t \left[\int_u^t \sigma_f(u,s)ds\right] \cdot \left\{\sigma_S(u) + \int_u^T [\sigma_f(u,s) - \sigma_\epsilon(u,s)]ds\right\} du \\ &= -\int_0^t \sigma_{P_t}(u) \cdot \sigma_{F_T}(u)du,\end{aligned}$$

where

$$\sigma_{P_T}(t) = -\int_t^T \sigma_f(t,s)ds$$

$$\sigma_{F_T}(t) = \sigma_S(t) + \int_t^T [\sigma_f(t,s) - \sigma_\epsilon(t,s)]ds$$

This is an extremely flexible model where the variances and covariances admits several specifications. One possibility is to assume a three-factor Gaussian model, with three deterministic σ processes defined as

$$\sigma_S(t) = \sigma_S \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

[4]In the same paper, Miltersen and Schwartz (1998) also give a separate formula for options on forwards, as well as on developing a relationship between the forward and futures contract.

[5]"$\cdot$" denotes the standard Euclidean inner product of R^d, and the corresponding norm is defined as $\|x\|^2 = x \cdot x$ for any $x \in R^d$.

$$\sigma_\epsilon(t,s) = \sigma_\epsilon e^{-\kappa_\epsilon(s-t)} \begin{pmatrix} \rho_{S\epsilon} \\ \sqrt{1-\rho_{S\epsilon}^2} \\ 0 \end{pmatrix}$$

$$\sigma_f(t,s) = \sigma_f e^{-\kappa_f(s-t)} \begin{pmatrix} \rho_{Sf} \\ \frac{\rho_{\epsilon f}-\rho_{S\epsilon}\rho_{Sf}}{\sqrt{1-\rho_{S\epsilon}^2}} \\ \sqrt{1-\rho_{Sf}^2 - \frac{(\rho_{\epsilon f}-\rho_{S\epsilon}\rho_{Sf})^2}{\sqrt{1-\rho_{S\epsilon}^2}}} \end{pmatrix}$$

This leads to the following solution of σ_z and σ_{xz}:

$$\begin{aligned}
\sigma_z^2 = \sigma_S^2 t + 2\sigma_S &\left\{\sigma_f \frac{\rho_{Sf}}{\kappa_f}\left[t - \frac{1}{\kappa_f}e^{-\kappa_f T}(e^{\kappa_f t}-1)\right]\right. \\
&\left. - \sigma_\epsilon \frac{\rho_{S\epsilon}}{\kappa_\epsilon}\left[t - \frac{1}{\kappa_\epsilon}e^{-\kappa_\epsilon T}(e^{\kappa_\epsilon t}-1)\right]\right\} \\
&+ \frac{\sigma_\epsilon^2}{\kappa_\epsilon^2}\left[t + \frac{1}{2\kappa_\epsilon}e^{-2\kappa_\epsilon T}(e^{2\kappa_\epsilon t}-1) - \frac{2}{\kappa_\epsilon}e^{-\kappa_\epsilon T}(e^{\kappa_\epsilon t}-1)\right] \\
&+ \frac{\sigma_f^2}{\kappa_f^2}\left[t + \frac{1}{2\kappa_f}e^{-2\kappa_f T}(e^{2\kappa_f t}-1) - \frac{2}{\kappa_f}e^{-\kappa_f T}(e^{\kappa_f t}-1)\right] \\
&- 2\sigma_\epsilon\sigma_f\rho_{\epsilon f}\frac{1}{\kappa_\epsilon}\frac{1}{\kappa_f}\left[t - \frac{1}{\kappa_\epsilon}e^{-\kappa_\epsilon T}(e^{\kappa_\epsilon t}-1) - \frac{1}{\kappa_f}e^{-\kappa_f T}(e^{\kappa_f t}-1)\right. \\
&\left. + \frac{1}{(\kappa_\epsilon+\kappa_f)}e^{-(\kappa_\epsilon+\kappa_f)T}\left(e^{(\kappa_\epsilon+\kappa_f)t}-1\right)\right] \\
\sigma_{xz} = \frac{\sigma_f}{\kappa_f}&\left\{\sigma_S\rho_{Sf}\left[t - \frac{1}{\kappa_f}(1-e^{-\kappa_f t})\right] + \frac{\sigma_f}{\kappa_f}\left[t - \frac{1}{\kappa_f}e^{-\kappa_f T}(e^{\kappa_f t}-1)\right.\right. \\
&\left. - \frac{1}{\kappa_f}(1-e^{-\kappa_f t}) + \frac{1}{2\kappa_f}e^{-\kappa_f T}(e^{\kappa_f t}-e^{-\kappa_f t})\right] \\
&- \sigma_\epsilon\frac{\rho_{\epsilon f}}{\kappa_\epsilon}\left[t - \frac{1}{\kappa_\epsilon}e^{-\kappa_\epsilon T}(e^{\kappa_\epsilon t}-1) - \frac{1}{\kappa_f}(1-e^{-\kappa_f t})\right. \\
&\left.\left. + \frac{1}{(\kappa_\epsilon+\kappa_f)}e^{-\kappa_\epsilon T}(e^{\kappa_\epsilon t}-e^{-\kappa_f t})\right]\right\},
\end{aligned}$$

TABLE 10-1

Comparison of European Futures Option Prices Using the Miltersen and Schwartz Commodity Option Model and the Black-76 and Black-76F Modified Black-Scholes Model

In this table, we follow the numerical examples used by Miltersen and Schwartz (1998) and are using the parameter estimates for the COMEX High Grade Copper Futures data presented in the Schwartz (1997) paper. In the Black-76 and Black-76F model, we set $\sigma = \sigma_S$.

$(F_T = 95, \sigma_S = 0.266, \sigma_\epsilon = 0.249, \sigma_f = 0.0096, \rho_{Sf} = 0.0964,$
$\rho_{S\epsilon} = 0.805, \rho_{\epsilon f} = 0.1243, \kappa_\epsilon = 1.045, \kappa_f = 0.2, P_t = e^{-0.05t})$

	X	Black-76	Black-76F	Miltersen–Schwartz
$t = 0.25$ $T = 0.25$	80	15.3430	15.3430	15.1918
	95	4.9744	4.9744	4.5669
	110	0.9159	0.9159	0.6896
$t = 0.25$ $T = 0.3$	80	15.3430	15.3047	15.1424
	95	4.9744	4.9619	4.4170
	110	0.9159	0.9136	0.6131
$t = 0.25$ $T = 0.5$	80	15.3430	15.1524	15.0049
	95	4.9744	4.9126	3.9251
	110	0.9159	0.9045	0.3903
$t = 0.5$ $T = 1$	80	16.1917	15.7919	15.0787
	95	6.9423	6.7709	4.7245
	110	2.3311	2.2735	0.7972

where

σ_S = Volatility of the spot commodity price.
σ_ϵ = Volatility of futures convenience yield.
σ_f = Volatility of the forward interest rate.
κ_f = Speed of mean reversion of the forward interest rate.
κ_ϵ = Speed of mean reversion of the convenience yield.
ρ_{Sf} = Correlation between the spot commodity price and the forward interest rate.
$\rho_{S\epsilon}$ = Correlation between the spot commodity price and the convenience yield.
$\rho_{\epsilon f}$ = Correlation between the forward interest rate and the convenience yield.

Using these expressions of σ_z and σ_{xz} in the Miltersen and Schwartz model, we can easily calculate values for European call and put options on commodity futures with stochastic forward interest rates and convenience yields.[6] Table 10-1 compares the Black-76 as well as

[6]This can be seen as a special case of the Miltersen and Schwartz option pricing model where the underlying futures follows a stochastic process as described in the paper by Schwartz (1997).

the Black-76F model with the Miltersen and Schwartz model. The price difference between the two types of models will be strongly dependent on the input parameters. Black-76F is simply the Black-76 formula adjusted for the fact that the payoff from an option on forwards typically arrives when the forward expires.

10.4 MEAN REVERSION MODEL

It is possible to derive a closed-form BSM type option formula when we assume that $\ln(S)$ follows a mean reversion process:

$$dS_t/S_t = \kappa(\theta - \ln(S))dt + \sigma dZ,$$

where θ is the mean reversion level and κ is the speed of mean reversion. Given then spot price volatility, σ_S, we can now calculate what the volatility of the forward must be

$$\sigma_F = \sigma_S\sqrt{\frac{1 - e^{-2\kappa T}}{2\kappa}} \tag{10.20}$$

The value of the option can now be found by the Black-76 formula by simply using the volatility from (10.20). Thus:

$$c = e^{-rT}[FN(d_1) - XN(d_2)] \tag{10.21}$$

$$p = e^{-rT}[XN(-d_2) - FN(-d_1)], \tag{10.22}$$

where

$$d_1 = \frac{\ln(F/X) + (\sigma_F^2/2)T}{\sigma_F\sqrt{T}}$$

$$d_2 = d_1 - \sigma_F\sqrt{T}$$

The mean reversion is already taken into account for in the forward price. For this reason, the mean reversion adjusted drift does not enter the option formula directly. Few traders would normally use this model. The reason is that it in general does not fit the implied volatility term structure observed in the market, at least as long as one insists on a constant speed of mean reversion. The model still gives some intuition on how mean reversion affects volatility. For more information on energy option valuation under mean reversion, see Pilipović (1997) and Eydeland and Wolyniec (2003).

10.5 SEASONALITY

Many commodities are subject to seasonal effects. Pilipović (1997) suggests a way to take seasonality into account through the stochastic process

$$S_t = S_t^{\text{Und}} + \text{Seasonal effect}, \tag{10.23}$$

where S_t^{Und} is the asset price process without adjustment for seasonality. The underlying stochastic process without seasonality dS_t^{Und} could take different forms—for example, a mean reversion model or simply a geometric Brownian motion. The seasonal adjustment will still take the same form:

$$dS_t = dS_t^{\text{Und}} - [2\pi\beta_A \sin(2\pi(t - t_A)) + 4\pi\beta_{SA}\sin(4\pi(t - t_{SA}))]dt,$$

where β_A and β_{SA} are the annual and semiannual seasonality parameters, respectively. t_A and t_{SA} are the timing for the centering of the annual and semiannual seasonality (seasonal peaks). The seasonality is thus assumed to be deterministic. The seasonal parameters, β_A, β_{SA}, t_A, and t_{SA}, can be specified by the trader (subjective beliefs) or calibrated to historical or current market data. The underlying price process without seasonality could for example be on the form

$$dS_t = S_t + \kappa(\theta - S_t)dt + \sigma S_t^{\beta} dZ,$$

where κ is the speed of mean reversion, θ is the mean reversion level, σ is the volatility, and β is a parameter that is equal to 1 for the "log-normal equivalent" volatility, and to 0 for a normally distributed asset price, or it could simply be calibrated to the market. A simple example of how to implement such a model with Monte Carlo simulation is given in an Excel spreadsheet on the accompanying CD.

It is, however, worthwhile to mention that many academics seem to overemphasize the importance of taking seasonality into account when, for example, modeling energy derivatives. Even if the spot price is normally seasonal, this is typically already reflected in the futures or forward prices. When modeling an option directly on a futures or forward, one therefore typically does not need to take into account seasonality in the model. Modeling options on the spot will, on the other hand, typically require a model that takes seasonality into account.

CHAPTER

11

INTEREST RATE DERIVATIVES

Go down deep enough into anything and you will find options. Then to value them you have to go even deeper.

Option Trader

In this chapter we consider how to value interest rate options. The description is limited mostly to hands-on models that are frequently used to price most standard fixed-income options. The academic literature on the topic has shown tremendous growth, but many of the suggested models are still in very limited use by market participants—perhaps for good reasons.

11.1 FRAs AND MONEY MARKET INSTRUMENTS

We first look at some useful formulas for interest rate forward rate agreements, better known as FRAs. As FRAs are the underlying building blocks in many interest rate options, it is worthwhile to understand the most basic formulas and arbitrage relationships for this type of contract. FRAs are, moreover, actively traded in the interbank market.

11.1.1 FRAs From Cash Deposits

FRAs with maturity less than one year are normally calculated directly from deposit rates, using the formula

$$\left(1 + r_1\frac{\tau_1}{Basis}\right)\left(1 + FRA\frac{\tau_2 - \tau_1}{Basis}\right) = \left(1 + r_2\frac{\tau_2}{Basis}\right)$$

$$FRA = \left[\frac{\left(1 + r_2\frac{\tau_2}{Basis}\right)}{\left(1 + r_1\frac{\tau_1}{Basis}\right)} - 1\right]\frac{Basis}{\tau_2 - \tau_1}, \tag{11.1}$$

where τ_1 and τ_2 are the number of days in the deposit period with corresponding cash rates r_1 and r_2, and *Basis* is the day count basis used in the specific market (for example, 360 or 365).

Example
Consider a three-month (91 days) LIBOR rate trading at 6%, and a six-month (183 days) LIBOR rate trading at 7%. The interest rate convention is $Act/360$. What is the theoretical FRA rate starting 91 days from today and ending 183 days from today? With $r_1 = 0.06$, $r_2 = 0.07$, $\tau_1 = 91$, $\tau_2 = 183$, and $Basis = 360$ we get

$$FRA = \left[\frac{\left(1 + 0.07\frac{183}{360}\right)}{\left(1 + 0.06\frac{91}{360}\right)} - 1\right]\frac{360}{183 - 91} = 0.0787 = 7.87\%$$

11.1.2 The Relationship between FRAs and Currency Forwards

We will now derive a relationship that can be used to compare FRAs (money market forward rate agreements) in different countries and/or produce synthetic FRAs. If you compare FRAs in one country with implied money market futures rates in another country, remember to adjust for lack of convexity in the money market futures yield. The convexity adjustment in money market futures is, however, often negligible for futures with maturity within a couple of years, and can typically be ignored. For longer dated contracts, the convexity adjustment can be of great importance.

$$1 + FRA\frac{\tau_2 - \tau_1}{Basis} = \left(\frac{Spot + \frac{LongSwap}{10000}}{Spot + \frac{ShortSwap}{10000}}\right)\left(1 + FRA_f\frac{\tau_2 - \tau_1}{Basis_f}\right)$$

$$FRA = \left[\left(\frac{Spot + \frac{LongFXSwap}{10000}}{Spot + \frac{ShortFXSwap}{10000}}\right)\left(1 + FRA_f\frac{\tau_2 - \tau_1}{Basis_f}\right) - 1\right]\frac{Basis}{\tau_2 - \tau_1}, \tag{11.2}$$

where $Spot$ is the currency spot rate and $Swap$ is the swap points in a currency forward. FRA_f is the FRA in the foreign market, and FRA is the synthetic domestic forward price.

Example
Assume a foreign FRA trading at 6% on a deposit starting 90 days from now and expiring 180 days from now with a day count basis of $Act/365$. Further, the spot currency rate is 7, the currency swap for the same period is bid 400 offered 600, and the day count basis in the domestic FRA market is $Act/360$.

$FRA_f = 0.06$, $Spot = 7$, $ShortFXSwap = 600$, $LongFXSwap = 400$, $\tau_1 = 90$, $\tau_2 = 180$, $Basis = 360$, $Basis_f = 365$ yields

$$FRA = \left[\left(\frac{7 + \frac{600}{10000}}{7 + \frac{400}{10000}}\right)\left(1 + 0.06\frac{180 - 90}{365}\right) - 1\right]\frac{360}{180 - 90} = 0.0707$$

11.1.3 Convexity Adjustment Money Market Futures

Money market futures—for example, Eurodollar futures—have a linear payoff for every basis point the interest rate changes, while FRAs and interest rate swaps have a convex payoff. That is, the Eurodollar (or similar money market futures) have no convexity. For this reason, a FRA with the same maturity as a Eurodollar futures must be more valuable than the futures. To avoid arbitrage opportunities, the Eurodollar implied rate, $100 - F$, must be higher than the FRA (forward) rate. The difference between the FRA rate and the future rate is called the convexity bias. Often, Eurodollar futures are much more liquid than the FRAs, and we can use the Eurodollar future price and a theoretical convexity bias to find the implied theoretical FRA rate (F_{Adj}). A strip of Eurodollar futures is also often used to hedge interest rate swaps. Also, in this case to calculate the implied swap rate from the futures, we first need to do the appropriate convexity adjustments.

Kirikos and Novak (1997) has published a closed-form money market convexity adjustment that has been quite popular among practitioners. The adjustment is based on the interest rate in the risk-neutral world and follows the Hull and White (1990a) model. (With no mean reversion, this gives convexity adjustment under the Ho and Lee (1986) model.)

$$F_{Adj} = F + \text{ConvexityBias}, \qquad (11.3)$$

where

$$\text{ConvexityBias} = (1 - e^{-Z})\left(100 - F + 100 \times \frac{Basis}{\tau}\right)$$

and

$$Z = \sigma^2\left(\frac{1 - e^{-2\kappa t_1}}{2\kappa}\right)\left(\frac{1 - e^{-\kappa(T_2 - t_1)}}{\kappa}\right)^2 + \frac{\sigma^2}{2\kappa^3}(1 - e^{-\kappa(T_2 - t_1)})(1 - e^{-\kappa t_1})$$

In the special case when the speed of mean reversion is zero, $\kappa = 0$, which is equivalent to the Ho-Lee model,[1] Z is set equal to

$$Z = \sigma^2 t_1 (T_2 - t_1)^2 + \frac{\sigma^2}{2} t_1^2 (T_2 - t_1)$$

F = Market price of the futures contract.
F_{Adj} = Theoretical futures price with convexity adjustment.
κ = Mean reversion speed.
t_1 = Time to maturity of the futures contract.
T_2 = Time to maturity of the underlying money (LIBOR) deposit.
σ = Volatility of underlying futures rate.
τ = The number of days in the underlying deposit period.
Basis = The day basis or number of days per year used in the market.

For more on convexity bias in money market futures, see also Flesaker (1993), Burghardt and Hoskins (1994), Burghardt and Panos (2001), Burghardt and Liu (2002), and Pieterbarg and Renedo (2004).

Example
A Eurodollar futures trades at 95.50 expiring six years from now, and the underlying three-month money deposit expires 90 days later. The money deposit is on a 360-day count basis (money market basis). The expected volatility is 150 basis points per year; the mean reversion speed is 0.04. $F = 96.50$, $t_1 = 6$, $T_2 = 6 + 90/365 = 6.2466$, $\sigma = 0.015$, $\kappa = 0.04$, and Basis $= 360$.

The unadjusted implied futures yield is $100 - 95.50 = 4.50$—that is, 4.5%. What is the implied convexity adjusted yield?

$$Z = 0.015^2 \left(\frac{1 - e^{-2\times 0.04\times 6}}{2 \times 0.04} \right) \left(\frac{1 - e^{-0.04(6.2466-6)}}{0.04} \right)^2$$

$$+ \frac{0.015^2}{2 \times 0.04^3} (1 - e^{-0.04(6.2466-6)})(1 - e^{-0.04\times 6}) = 0.00085$$

$$\text{ConvexityBias} = (1 - e^{-0.00085}) \left(100 - 95.5 + 100 \times \frac{360}{90} \right) = 0.3437$$

$$F_{Adj} = 95.5 + 0.3437 = 95.8437$$

[1]Using a slightly different notation (all with continuous compounding rates), Hull (2005) gives with background in the Ho-Lee model the following formula for the convexity adjustment: $0.5\sigma^2 t_1 T_2$.

This gives us an implied forward rate of $100 - 95.8437 = 4.1563$—that is, 4.1563%, which is 34 basis points lower than the unadjusted futures rate due to the convexity adjustment.

11.2 SIMPLE BOND MATHEMATICS

11.2.1 Dirty and Clean Bond Price

In the bond market people are speaking about clean and dirty bond prices. A dirty bond price is the price quoted in the market plus the accrued interest rates. In other words, the dirty price is the price you have to pay for the bond. Bond prices are typically quoted as clean prices, though—the bond price without accrued interest rate. When you are pricing, for instance, bond options, it is important to know if the strike price is against the clean or dirty bond price.

11.2.2 Current Yield

Although not very useful in derivatives calculations, several bond investor still use the concept of "current yield":

$$\text{Current yield} = \frac{\text{Annual coupon income in \$}}{\text{Dirty bond price}}$$

11.2.3 Modified Duration and BPV

$$\text{ModifiedDuration} = \frac{\text{Duration}}{1 + \text{Yield}} \tag{11.4}$$

A bonds basis point value (BPV) is how much a bond changes in value with a 1-basis-point change in yield. BPV per million notional can be found by

$$BPV = -\text{ModifiedDuration} \times \text{DirtyPrice} \tag{11.5}$$

We can similarly find the BPV of a swap by first calculating the duration of the swap:

$$\text{SwapDuration} = \text{FixedSideDuration} - \text{FloatSideDuration} \tag{11.6}$$

Example
Consider a bond with duration 6.43, yield 5.28%, clean price 103.02, and accrued interest 1.48. What is the BPV of the bond?

$$BPV = -\frac{6.43}{1 + 0.0528}(103.02 + 1.48) = -638.24$$

Thus, for every 1-basis-point increase in yield, the bond will decrease 638.24 dollars for every 1 million in notional.

11.2.4 Bond Price and Yield Relationship

The bond price continuous compounded yield relationship is

$$P = \sum_{i=1}^{n} C_i e^{-yT_i},$$

where

$$\begin{aligned} P &= \text{Bond price} \\ y &= \text{Bond yield continuous compounding} \\ C_i &= \text{Cash flow (coupon) at time } T_i \end{aligned}$$

The bond's sensitivity to a small change in yield is

$$\frac{\partial P}{\partial y} = \sum_{i=1}^{n} -C_i T_i e^{-yT_i}$$

The convexity of a bond is given by

$$\frac{\partial^2 P}{\partial y^2} = \sum_{i=1}^{n} C_i T_i^2 e^{-yT_i}$$

11.2.5 Price and Yield Relationship for a Bond

It is more common to quote the bond yield as annual compounding or with the same compounding as the number of coupons per year. Assuming annual compounding yield, the dirty price of a bond (clean price $P +$ accrued interest rates) at any time is given by

$$\begin{aligned} P + C\frac{\tau_1}{Basis} &= \frac{C}{m}\left[\frac{1}{(1+y)^{\frac{\tau_2}{Basis}}} + \frac{1}{(1+y)^{\frac{\tau_2}{Basis}+\frac{1}{m}}} + \cdots + \frac{1}{(1+y)^{\frac{\tau_2}{Basis}+\frac{N-1}{m}}}\right] \\ &\quad + \frac{F}{(1+y)^{\frac{\tau_2}{Basis}+\frac{N-1}{m}}} \\ &= (1+y)^{\frac{1}{m}-\frac{\tau_2}{Basis}}\left[\frac{C}{m} \times \frac{(1+y)^{-\frac{N}{m}} - 1}{1-(1+y)^{\frac{1}{m}}} + \frac{F}{(1+y)^{\frac{N}{m}}}\right], \end{aligned} \tag{11.7}$$

where N is the number of coupons left to maturity, m is the number of coupons per year, y is the yield to maturity, τ_1 is the number of days since the last coupon payment, τ_2 is the number of days to the next coupon payment, F is the face value of the bond, and *Basis* is the day basis (i.e., 365 or 360).

11.2.6 From Bond Price to Yield

The Newton-Raphson algorithm is useful to find the yield given the bond price:

$$y_{n+1} = y_n - \frac{P(y_n) - P_m}{\frac{\partial P}{\partial y_n}},$$

until $| P_m - P(y_{n+1}) | \leq \epsilon$, at which point y_{n+1} is the implied yield. ϵ is the desired degree of accuracy. P_m is the market price of the bond, and $\frac{\partial P}{\partial y_n}$ is the delta of the bond (the sensitivity of the bond value for a small change in the yield).

$$\frac{\partial P}{\partial y} = a^{\frac{1}{m} - \frac{d_2}{Basis}} \left[\frac{\frac{1}{m} - \frac{d_2}{Basis}}{a} \left(\frac{C}{m} \frac{a^{-\frac{N}{m}} - 1}{1 - a^{\frac{1}{m}}} + F a^{-\frac{N}{m}} \right) + \frac{C}{m^2} \left(\frac{a^{-\frac{N}{m}}[1 + (N+1)(a^{\frac{1}{m}} - 1)] - a^{\frac{1}{m}}}{a(1 - a^{\frac{1}{m}})^2} \right) - F \frac{N}{m} a^{-\frac{N}{m} - 1} \right],$$

where $a = 1 + y$. Alternatively, assuming yield compounding equal to the number of coupons per year gives the dirty price of a bond (price + accrued interest rates) as a function of the yield:

$$P + C\frac{\tau_1}{Basis} = \left(1 + \frac{y}{m}\right)^{1 - \frac{\tau_2}{Basis/m}} \left[\frac{C}{m} \times \frac{(1 + \frac{y}{m})^{-N} - 1}{1 - (1 + \frac{y}{m})} + \frac{F}{(1 + \frac{y}{m})^N} \right] \quad (11.8)$$

11.3 PRICING INTEREST RATE OPTIONS USING BLACK-76

The Black-76 model is probably the most widely used model to price interest rate options. The model was originally developed to price options on forwards and assumes that the underlying asset is lognormally distributed. When used to price a cap, for example, the underlying forward rates of the cap are thus assumed to be lognormal. Similarly, when used to price a swaption (an option on a swap), the underlying swap rate is assumed to be lognormal. This can be justified when pricing these types of options independently (Jamshidian, 1996; Miltersen, Sandmann, and Sondermann, 1997). Still, using the model to price both a cap and a swaption is theoretically inconsistent. The cap forward rate and the swap rate cannot both be lognormal. Still, the overwhelming popularity of this model for pricing both caps and swaptions suggests that any problems due to this inconsistency are economically insignificant.[2]

---

[2]Traders typically take inconsistencies into account by adjusting the volatility. The adjustment is based on experience from the particular market in which they operate.

The problem of inconsistent pricing is true also for bonds and swaps: They cannot both have lognormal rates. For instance, if the bond price is assumed to be lognormal, the continuously compounded swap rate must be normally distributed. Using the same model (Black-76) for pricing swaptions and bond options is then inconsistent. The next section is a slight detour on options on money market futures, before we come back to swaptions and bond options.

11.3.1 Options on Money Market Futures

Several exchanges list actively traded options on money market futures. Examples of these are options on Eurodollar futures traded on the Chicago Mercantile Exchange (CME) and short Sterling futures traded on the London International Futures Exchange (LIFFE). The price on a money market futures is typically quoted as 100 minus the yield. For instance, a price of 94.56 would imply a money market futures implied rate of $100 - 94.56 = 5.44$. Assuming that we cannot have negative real interest rates, the price on the money market futures has an upper bound of 100. The price on the money market futures is clearly not lognormally distributed. It is, on the other hand, quite reasonable to model the implied money market futures yield as being lognormally distributed. Based on this assumption, most traders use the Black-76 formula directly on the implied yield. If one uses this approach, one must remember that a call on the money market futures must be priced as a put on the yield and vice versa:

$$p = e^{-rT}[yN(d_1) - XN(d_2)] \tag{11.9}$$

$$c = e^{-rT}[XN(-d_2) - yN(-d_1)], \tag{11.10}$$

where y is the implied money market futures yield; $y = 100 - F$, and

$$d_1 = \frac{\ln(y/X) + (\sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

Example

Consider a European put option with three months to expiration on a Eurodollar futures contract. The Eurodollar futures currently trades at 94.52, the volatility of the implied money market yield is 23%, the strike of the put is 94.25, and the risk-free rate used for discounting is 5%. $y = 100 - 94.52 = 5.48$, $X = 5.75$, $T = 0.25$, $r = 0.05$, and $\sigma = 0.23$.

Thus:

$$d_1 = \frac{\ln(5.48/5.75) + (0.23^2/2)0.25}{0.23\sqrt{0.25}} = -0.3607$$

$$d_2 = d_1 - 0.23\sqrt{0.25} = -0.4757$$

$$N(d_1) = N(-0.3607) = 0.3592 \qquad N(d_2) = N(-0.4757) = 0.3171$$

$$p = e^{-0.05\times 0.25}[5.48N(d_1) - 5.75N(d_2)] = 0.1428$$

The value is thus approximately 14 basis points, and in a three-month Eurodollar futures contract, one basis point is worth \$25. In other words, the option is approximately worth \$350 (14 × 25) per contract we have the right to sell.

11.3.2 Price and Yield Volatility in Money Market Futures

The yield y of money market futures F (i.e., Eurodollar futures) is

$$y = 100 - F$$

The relation between price volatility and yield volatility is

$$\sigma_y = \sigma_F \frac{F}{100 - F} \tag{11.11}$$

$$\sigma_F = \sigma_y \frac{100 - F}{F} \tag{11.12}$$

Example

Consider a Eurodollar futures with price 94.53 and price volatility 1.5%. What is the equivalent yield volatility? $F = 94.53$, $\sigma_F = 0.015$, and $y = 5.47\%$.

$$\sigma_y = 0.015 \frac{94.53}{100 - 94.53} = 25.92\%$$

11.3.3 Caps and Floors

An interest rate cap consists of a series of individual European call options, called caplets. Each caplet can be priced by a modified version of the Black-76 formula. This is accomplished by using the implied forward rate, F, at each caplet maturity as the underlying asset. The price of the cap is the sum of the price of the caplets that make up the cap. Similarly, the value of a floor is the sum of the sequence of individual put options, often called floorlets, that make up the floor.

$$Cap = \sum_{i=1}^{n} Caplet_i \qquad Floor = \sum_{i=1}^{n} Floorlet_i,$$

where

$$\text{Caplet value} = \frac{\text{Notional} \times \frac{\tau}{Basis}}{\left(1 + F\frac{\tau}{Basis}\right)} \times \text{Black-76 call value}$$

$$= \frac{\text{Notional} \times \frac{\tau}{Basis}}{\left(1 + F\frac{\tau}{Basis}\right)} e^{-rT}[FN(d_1) - XN(d_2)] \tag{11.13}$$

τ is the number of days in the forward rate period. $Basis$ is the day basis or number of days per year used in the market (i.e., 360 or 365).

$$\text{Floorlet value} = \frac{\text{Notional} \times \frac{\tau}{Basis}}{\left(1 + F\frac{\tau}{Basis}\right)} e^{-rT}[XN(-d_2) - FN(-d_1)], \tag{11.14}$$

where

$$d_1 = \frac{\ln(F/X) + (\sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = \frac{\ln(F/X) - (\sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T}$$

Miltersen, Sandmann, and Sondermann (1997) recently developed a closed-form lognormal yield-based model for the pricing of caps and floors. Their model is an extension of the modified Black-76 caps and floor model, which allows for stochastic discount rates and time-dependent volatility.

Example

What is the value of a caplet on a 182-day forward rate, with six months to expiration and a notional principle of 100 million? The six-month forward rate is 8% ($\frac{Act}{360}$ basis), the strike is 8%, the risk-free interest rate is 7%, and the volatility of the forward rate is 28% per year. $Basis = 360$, $\tau = 182$, $F = 0.08$, $X = 0.08$, $T = 0.5$, $r = 0.07$, and $\sigma = 0.28$, which yields

$$d_1 = \frac{\ln(0.08/0.08) + (0.28^2/2)0.5}{0.28\sqrt{0.5}} = 0.0990 \qquad d_2 = d_1 - 0.28\sqrt{0.5} = -0.0990$$

$$N(d_1) = N(0.0990) = 0.5394 \qquad N(d_2) = N(-0.0990) = 0.4606$$

$$\text{Caplet value} = \frac{100{,}000{,}000 \times \frac{182}{360}}{\left(1 + 0.08\frac{182}{360}\right)} e^{-0.07\times 0.5}[0.08N(d_1) - 0.08N(d_2)] = 295.995$$

11.3.4 Swaptions

It is usual to distinguish between the following:

Payer Swaption The right but not the obligation to pay the fixed rate and receive the floating rate in the underlying swap.

Receiver Swaption The right but not the obligation to receive the fixed rate and pay the floating rate in the underlying swap.

European swaptions are normally priced by using the forward swap rate as input in the Black-76 option pricing model.[3] The Black-76 value is multiplied by a factor adjusting for the tenor of the swaption, as shown by Smith (1991). This is the practitioner's benchmark swaption model. The model is arbitrage-free under the assumption of a lognormal swap rate (Jamshidian, 1996).

$$c = \left[\frac{1 - \frac{1}{\left(1+\frac{F}{m}\right)^{t_1 \times m}}}{F}\right] e^{-rT}[FN(d_1) - XN(d_2)] \tag{11.15}$$

$$p = \left[\frac{1 - \frac{1}{\left(1+\frac{F}{m}\right)^{t_1 \times m}}}{F}\right] e^{-rT}[XN(-d_2) - FN(-d_1)], \tag{11.16}$$

where

$$d_1 = \frac{\ln(F/X) + (\sigma^2/2)T}{\sigma\sqrt{T}}, \quad d_2 = d_1 - \sigma\sqrt{T},$$

where c indicates a payer swaption, and p indicated a receiver swaption.

t_1 = Tenor of swap in years.
F = Forward rate of underlying swap.
X = Strike rate of swaption.
r = Risk-free interest rate.
T = Time to expiration in years.
σ = Volatility of the forward-starting swap rate.
m = Compoundings per year in swap rate.

Example

Consider a two-year payer swaption on a four-year swap with semiannual compounding. The forward swap rate of 7% starts two years from now and ends six years from now. The strike is 7.5%, the risk-free interest rate is 6%, and the volatility of the forward starting swap rate is 20% per year. $t_1 = 4, m = 2, F = 0.07, X = 0.075, T = 2, r = 0.06,$

[3] More than 90% of the swaptions' market consists of European swaptions. To price American swaptions, look at section 11.4 for yield-based models.

and $\sigma = 0.2$.

$$d_1 = \frac{\ln(0.07/0.075) + (0.2^2/2)2}{0.2\sqrt{2}} = -0.1025 \qquad d_2 = d_1 - 0.2\sqrt{2} = -0.3853$$

$$N(d_1) = N(-0.1025) = 0.4592 \qquad N(d_2) = N(-0.3853) = 0.3500$$

$$c = e^{-0.06\times 2}[0.07N(d_1) - 0.075N(d_2)] = 0.5227\%$$

With a semiannual forward swap rate, the up-front value of the payer swaption in percent of the notional is

$$c \times \left[\frac{1 - \frac{1}{\left(1+\frac{0.07}{2}\right)^{4\times 2}}}{0.07}\right] = 1.7964\%$$

11.3.5 Swaption Volatilities from Caps or FRA Volatilities

When you are trading caps, floors, and swaptions, it is of interest to be able to compare cap and floor volatilities with swaption volatilities. The relationship between swap rates, Y, and forward rates, F_i is

$$Y = \frac{\sum_{i=1}^{n} N_i F_i \tau_i P(0, t_{i+1})}{\sum_{i=1}^{n} N_i \tau_i P(0, t_{i+1})}$$

This shows that the swap rate is a weighted average of the forward rates. This becomes more clear by introducing (see Rebonato, 1996)

$$w_i = \frac{N_i \tau_i P(0, t_{i+1})}{\sum_{j=1}^{n} N_i \tau_i P(0, t_{i+1})}$$

In terms of the swap rate

$$Y = \sum_{i=1}^{n} F_i \frac{N_i \tau_i P(0, t_{i+1})}{\sum_{j=1}^{n} N_i \tau_i P(0, t_{i+1})}$$

By assuming the weights w_i to be approximately constant with rate movements, we can approximate the swap volatility by

$$\sigma_Y \approx \sqrt{\sum_{i=1}^{n} w_i^2 \sigma_{F_i}^2 + 2\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} w_i w_j \rho_{i,j} \sigma_{F_i} \sigma_{F_j}} \tag{11.17}$$

This formula can be useful for comparing swaption volatilities with caps and floors volatilities and vice versa. Alternatively, the relationship can be used to calculate implied correlations, taking the swaption and cap volatilities as given. For a more in-depth analysis of the relationship between caps and swaptions, see Longstaff, Santa-Clare, and Schwartz (2000).

Computer algorithm

The *SwaptionVol(·)* function returns the swap volatility given the one-year FRA volatilities and their correlations. The function is limited to calculate the volatility of swaps with full years to maturity, from annual forward rates (FRAs). It can be easily extended to cover swaps, for example, starting three months from now and based on, for example, three-month FRAs. I leave this extension up to the reader. See the accompanying spreadsheet on the CD to attain a better understanding of how to use the function.

```
Function SwaptionVol(SwapStart As Integer, SwapTenor As Integer, _
  Zeros As Variant, Vols As Variant, _
  CorrelationMatrix As Variant) As Double

        Dim Weight() As Double
        no_weights = Application.Count(Vols)
        ReDim Weight(no_weights) As Double

        For i = SwapStart + 1 To SwapStart + SwapTenor
            Weight(i) = 1 / (1 + Zeros(i))^i
            SumDiscountFactors = SumDiscountFactors + Weight(i)
        Next

        For i = SwapStart + 1 To SwapStart + SwapTenor
            Weight(i) = Weight(i) / SumDiscountFactors
        Next

        For i = SwapStart + 1 To SwapStart + SwapTenor
                Sum = Sum + Weight(i)^2 * Vols(i)^2
            For j = i + 1 To SwapStart + SwapTenor
                Sum = Sum + 2 * Weight(i) * Weight(j) _
        * Vols(i) * Vols(j) * CorrelationMatrix(i, j)
            Next
        Next

        SwaptionVol = Sqr(Sum)

End Function
```

11.3.6 Swaptions with Stochastic Volatility

For more sophisticated valuation of swaptions with a two-factor stochastic volatility model, one interesting approach is to combine the method above with the SABR model described in Chapter 6.

11.3.7 Convexity Adjustments

A standard bond or interest rate swap exhibits a convex price-yield relationship. To price options with the Black-76 model when the underlying asset is a derivative security, with a payoff function linear in the bond or swap yield, the yield should be adjusted for the lack of convexity.

Examples of derivatives where the payoff is a linear function of the bond or swap yield are constant maturity swaps (CMS) and constant maturity treasury swaps (CMT). The closed-form formula published by Brotherton-Ratcliffe and Iben (1993) assumes that the forward yield is lognormally distributed.[4]

$$\text{Convexity adjustment} = -\frac{1}{2}\frac{\frac{\partial^2 P}{\partial y_F^2}}{\frac{\partial P}{\partial y_F}} y_F^2 (e^{\sigma^2 T} - 1), \tag{11.18}$$

where

P = Bond or fixed side swap value.
y_F = Forward yield.
T = Time to payment date in years.
σ = Volatility of the forward yield.

Example
Consider a derivative instrument with a single payment five years from now that is based on the notional principal times the yield of a standard four-year swap with annual payments. The forward yield of the four-year swap, starting five years in the future and ending nine years in the future, is 7%. The volatility of the forward swap yield is 18%. What is the convexity adjustment of the swap yield? The value of the fixed side of the swap with annual yield is equal to the value of a bond where the coupon is equal to the forward swap rate/yield y_f:

$$P = \frac{c}{1+y_F} + \frac{c}{(1+y_F)^2} + \frac{c}{(1+y_F)^3} + \frac{1+c}{(1+y_F)^4}$$

The partial derivative of the swap with respect to the yield is

$$\begin{aligned}\frac{\partial P}{\partial y_F} &= -\frac{c}{(1+y_F)^2} - \frac{2c}{(1+y_F)^3} - \frac{3c}{(1+y_F)^4} - \frac{4(1+c)}{(1+y_F)^5}\\ &= -\frac{0.07}{(1+0.07)^2} - \frac{2\times 0.07}{(1+0.07)^3} - \frac{3\times 0.07}{(1+0.07)^4} - \frac{4(1+0.07)}{(1+0.07)^5} = -3.3872,\end{aligned}$$

and the second partial derivative with respect to the forward swap rate is

$$\begin{aligned}\frac{\partial^2 P}{\partial y_F^2} &= \frac{2c}{(1+y_F)^3} + \frac{6c}{(1+y_F)^4} + \frac{12c}{(1+y_F)^5} + \frac{20(1+c)}{(1+y_F)^6}\\ &= \frac{2\times 0.07}{(1+0.07)^3} + \frac{6\times 0.07}{(1+0.07)^4} + \frac{12\times 0.07}{(1+0.07)^5} + \frac{20(1+0.07)}{(1+0.07)^6} = 15.2933\end{aligned}$$

[4]The original formula published by Brotherton-Ratcliffe and Iben (1993) is slightly different.

The convexity adjustment can now be found using equation (11.18):

$$\text{Convexity Adjustment} = -\frac{1}{2}\frac{15.2933}{-3.3872}0.07^2(e^{0.18^2\times 5} - 1) = 0.0019$$

The convexity adjusted rate is then equal to 7.19% (0.07 + 0.0019).

Vega of the Convexity Adjustment
The convexity adjustment's sensitivity to a small change in volatility is given by

$$\text{Vega} = -\frac{\frac{\partial^2 P}{\partial y_F^2}}{\frac{\partial P}{\partial y_F}} y_F^2 \sigma T e^{\sigma^2 T} \tag{11.19}$$

Implied Volatility from the Convexity Value in a Bond
If the convexity adjustment is known, it is possible to calculate the implied volatility by simply rearranging the convexity adjustment formula:

$$\sigma = \sqrt{\ln\left(\frac{\text{Convexity adjustment}}{-\frac{1}{2}\frac{\frac{\partial^2 P}{\partial y_F^2}}{\frac{\partial P}{\partial y_F}} y_F^2} + 1\right)\frac{1}{T}} \tag{11.20}$$

11.3.8 European Short-Term Bond Options

European bond options can be priced with the Black-76 model by using the forward price of the bond at expiration as the underlying asset:

$$c = e^{-rT}[FN(d_1) - XN(d_2)] \tag{11.21}$$

$$p = e^{-rT}[XN(-d_2) - FN(-d_1)], \tag{11.22}$$

where F is the forward price of the bond at the expiration of the option and

$$d_1 = \frac{\ln(F/X) + (\sigma^2/2)T}{\sigma\sqrt{T}} \qquad d_2 = d_1 - \sigma\sqrt{T}$$

This model does not take into consideration the pull-to-par effect of the bond: At maturity, the bond price must equal the principal plus the coupon. The uncertainty of a bond will, for this reason, first increase and then decrease.

The Black-76 model assumes that the variance of the rate of return on the underlying asset increases linearly with time to maturity. Pricing of European bond options using this approach should thus be limited to options with short time to maturity relative to the time to maturity of the bond. A rule of thumb used by some traders is that the time to maturity of the option should be no longer than one-fifth of the time to maturity on the underlying bond.

Example

Consider a European put option with six months to expiration and strike price 122 on a bond with forward price at option expiration equal to 122.5. The volatility of the forward price is 4%, and the risk-free discount rate is 5%. What is the option's value? $F = 122.5$, $X = 122$, $T = 0.5$, $r = 0.05$, and $\sigma = 0.04$.

$$d_1 = \frac{\ln(122.5/122) + (0.04^2/2)0.5}{0.04\sqrt{0.5}} = 0.1587$$

$$d_2 = 0.1587 - 0.04\sqrt{0.5} = 0.1305$$

$$N(-d_1) = N(-0.1587) = 0.4369 \quad N(-d2) = N(-0.1305) = 0.4481$$

$$p = e^{-0.05\times 0.5}[122 \times 0.4481 - 122.5 \times 0.4369] = 1.1155$$

11.3.9 From Price to Yield Volatility in Bonds

The following formulas can be used to find the yield volatility of a bond if one knows the price volatility, and vice versa.

$$\sigma_y = \frac{\sigma_P}{\frac{\partial P}{\partial y}\frac{1}{P}y} = \frac{\sigma_P}{y\frac{\text{Duration}}{(1+y)}} \tag{11.23}$$

$$\sigma_P = \sigma_y \frac{\partial P}{\partial y}\frac{1}{P}y = \sigma_y\left[y\frac{\text{Duration}}{(1+y)}\right], \tag{11.24}$$

where σ_P is the price volatility of the bond price P and σ_y is the yield volatility of the bond yield y.

Example

Consider a government bond where the implied price volatility is 9%. The bond has a duration of six years and a yield to maturity of 8%. What is the equivalent yield volatility of the bond?

$$\sigma_y = \frac{0.09}{0.08\frac{6}{1+0.08}} = 20.25\%$$

11.3.10 The Schaefer and Schwartz Model

Schaefer and Schwartz (1987) developed a modified BSM model for pricing bond options. The model incorporates that the price volatility of a bond increases with its duration:

$$c = Se^{(b-r)T}N(d_1) - Xe^{-rT}N(d_2) \tag{11.25}$$

$$p = Xe^{-rT}N(-d_2) - Se^{(b-r)T}N(-d_1), \tag{11.26}$$

where

$$\sigma = (KS^{\alpha-1})D$$

TABLE 11-1

Comparison of the Black-76 Formula with the Schaefer and Schwartz Volatility-Adjusted Black-76 Formula

$(F = 100, X = 100, T = 2, r = 0.1, b = 0)$

Bond Duration	Base Volatility	Adjusted Volatility	Black-76 Value	Modified Black-76 Value
1	12.0%	1.5%	5.5364	0.6929
2	12.0%	3.0%	5.5364	1.3857
3	12.0%	4.5%	5.5364	2.0783
4	12.0%	6.0%	5.5364	2.7707
5	12.0%	7.5%	5.5364	3.4628
6	12.0%	9.0%	5.5364	4.1545
7	12.0%	10.5%	5.5364	4.8457
8	12.0%	12.0%	5.5364	5.5364

$$d_1 = \frac{\ln(S/X) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T},$$

where D is the duration of the bond after the option expires. K is estimated from the observed price volatility σ_o of the bond. α is a constant that Schaefer and Schwartz suggest should be set equal to 0.5. Moreover,

$$K = \frac{\sigma_o}{S^{\alpha-1}D^*},$$

where D^* is the duration of the bond today.

Example

Assume that the duration of the bond is eight years and that the observed price volatility of the bond is 12%. This gives

$$K = \frac{0.12}{100^{0.5-1}8} = 0.15$$

Table 11-1 uses this value and compares the option prices from the Schaefer and Schwartz formula with option prices from the Black-76 formula.

11.4 ONE-FACTOR TERM STRUCTURE MODELS

11.4.1 The Rendleman and Bartter Model

The Rendleman and Bartter (1980) model is a one-factor equilibrium model that assumes that the short-term interest rate is lognormal:

$$dr = \mu r dt + \sigma r dz,$$

where μ is the expected instantaneous relative change in the short-term interest rate and σ is the instantaneous standard deviation of the change. The model can be implemented in a binomial tree similar to the Cox-Ross-Rubinstein tree described in Chapter 7. The up and down factors are

$$u = e^{\sigma\sqrt{\Delta t}}, \qquad d = e^{-\sigma\sqrt{\Delta t}}$$

The probability of moving up is

$$p = \frac{e^{\mu\Delta t} - d}{u - d}$$

11.4.2 The Vasicek Model

The Vasicek (1977) model is a yield-based one-factor equilibrium model that assumes that the short rate is normally distributed. The model incorporates mean reversion and is popular in the academic community—mainly due to its analytic tractability. The model is not used much by market participants because it is not ensured to be arbitrage-free relative to the underlying securities already in the marketplace.

$$dr = \kappa(\theta - r)dt + \sigma dz \tag{11.27}$$

κ is the speed of the mean reversion, and θ is the mean reversion level.

Bond Prices

The price at time t of a discount bond maturing at time T is $P(t, T)$, where

$$P(t, T) = A(t, T)e^{-B(t,T)r(t)},$$

$r(t)$ is the rate at time t and

$$B(t, T) = \frac{1 - e^{-\kappa(T-t)}}{\kappa}$$

$$A(t, T) = \exp\left[\frac{(B(t, T) - T + t)(\kappa^2\theta - \sigma^2/2)}{\kappa^2} - \frac{\sigma^2 B(t, T)^2}{4\kappa}\right]$$

European Options

The value of a European option maturing at time T on a zero-coupon bond that matures at time τ is

$$c = P(t, \tau)N(h) - XP(t, T)N(h - \sigma_P) \tag{11.28}$$

$$p = XP(t, T)N(-h + \sigma_P) - P(t, \tau)N(-h), \tag{11.29}$$

where

$$h = \frac{1}{\sigma_P}\ln\left[\frac{P(t,\tau)}{P(t,T)X}\right] + \frac{\sigma_P}{2}$$

$$\sigma_P = B(T,\tau)\sqrt{\frac{\sigma^2(1-e^{-2\kappa(T-t)})}{2\kappa}}$$

Example

Consider a European call option on a zero-coupon bond. Time to expiration is two years, the strike price is 92, the volatility is 3%, the mean-reverting level is 9%, and the mean reverting rate is 0.05. The face value of the bond is 100 with time to maturity three years and initial risk-free rate of 8%. $F = 100$, $X = 92$, $T = 2$, $\tau = 3$, $\theta = 0.09$, $\kappa = 0.05$, $r = 0.08$, and $\sigma = 0.03$.

$$B(t,T) = B(0,2) = \frac{1-e^{-0.05(2-0)}}{0.05} = 1.9032$$

$$B(T,\tau) = B(2,3) = \frac{1-e^{-0.05(3-2)}}{0.05} = 0.9754$$

$$B(t,\tau) = B(0,3) = \frac{1-e^{-0.05(3-0)}}{0.05} = 2.7858$$

$$A(t,T) = A(0,2) = \exp\left[\frac{(B(0,2)-2+0)(0.05^2\times 0.09 - 0.03^2/2)}{0.05^2} - \frac{0.03^2 B(0,2)^2}{4\times 0.05}\right] = 0.9924$$

$$A(t,\tau) = A(0,3) = \exp\left[\frac{(B(0,3)-3+0)(0.05^2\times 0.09 - 0.03^2/2)}{0.05^2} - \frac{0.03^2 B(0,3)^2}{4\times 0.05}\right] = 0.9845$$

$$P(t,T) = P(0,2) = A(0,2)e^{-B(0,2)0.08} = 0.8523$$

$$P(t,\tau) = P(0,3) = A(0,3)e^{-B(0,3)0.08} = 0.7878$$

$$\sigma_P = B(2,3)\sqrt{\frac{0.03^2(1-e^{-2\times 0.05\times 2})}{2\times 0.05}} = 0.0394$$

$$h = \frac{1}{\sigma_P}\ln\left[\frac{P(0,3)}{P(0,2)92}\right] + \frac{\sigma_P}{2} = 0.1394$$

The call value for one USD in face value is

$$c = P(0,3)N(h) - 92P(0,2)N(h-\sigma_P) = 0.0143$$

With a face value of 100, the call value is 1.43 USD (100×0.0143)

Jamshidian's Approach for Coupon Bonds
Jamshidian (1989) shows that a European option on a coupon bond can be decomposed into a portfolio of options on zero-coupon bonds, where each coupon is treated as a zero-coupon bond. One proceeds by finding the value of the risk-free interest rate $\hat{r}$ at time T that causes the value of the coupon bond to equal the strike price.

Example
Consider a European call option on a coupon bond. Time to expiration is four years, the strike price 99.5, the volatility is 3%, the mean-reverting level is 10%, and the mean-reverting rate is 0.05. The face value of the bond is 100, and it pays a semiannual coupon of 4. Time to maturity is seven years, and the risk-free rate is initially 9%.

First find the rate $\hat{r}$ that makes the value of the coupon bond equal to the strike price at the option's expiration. Trial and error gives $\hat{r} = 8.0050\%$. To find the value of the option, we have to determine the value of six different options:

1. A four-year option with strike price 3.8427 on a 4.5-year zero-coupon bond with a face value of four
2. A four-year option with strike price 3.6910 on a five-year zero-coupon bond with a face value of four
3. A four-year option with strike price 3.5452 on a 5.5-year zero-coupon bond with a face value of four
4. A four-year option with strike price 3.4055 on a six-year zero-coupon bond with a face value of four
5. A four-year option with strike price 3.2717 on a 6.5-year zero-coupon bond with a face value of four
6. A four-year option with strike price 81.7440 on a seven-year zero-coupon bond with a face value of 104.

The value of the six options are, respectively, 0.0256, 0.0493, 0.0713, 0.0917, 0.1105, and 3.3219. This gives a total value of 3.6703.

11.4.3 The Ho and Lee Model

Ho and Lee (1986) published the first arbitrage-free yield-based model. It assumes a normally distributed short-term rate. This enables analytical solutions for European bond options. The short rate's drift depends on time, thus making the model arbitrage-free with respect to observed prices (the input to the model). The model does not incorporate mean reversion.

$$dr = \theta(t)dt + \sigma dz, \tag{11.30}$$

where $\theta(t)$ is a time-dependent drift.

Bond Prices
The price at time t of a discount bond maturing at time T is $P(T)$, given by

$$P(t,T) = A(t,T)e^{-r(t)(T-t)},$$

where $r(t)$ is the rate at time t and

$$\ln A(t,T) = \ln\left(\frac{P(0,T)}{P(0,t)}\right) - (T-t)\frac{\partial \ln P(0,t)}{\partial t} - \frac{1}{2}\sigma^2 t(T-t)^2$$

European Options
The value of a European option maturing at time T on a zero-coupon bond maturing at time τ is

$$c = P(t,\tau)N(h) - XP(t,T)N(h-\sigma_P) \tag{11.31}$$

$$p = XP(t,T)N(h-\sigma_P) - P(t,\tau)N(h), \tag{11.32}$$

where

$$h = \frac{1}{\sigma_P}\ln\left[\frac{P(t,\tau)}{P(t,T)X}\right] + \frac{\sigma_P}{2}$$

$$\sigma_P = \sigma(\tau - T)\sqrt{T-t}$$

11.4.4 The Hull and White Model

The Hull and White (1990a) model is simply the Ho and Lee model with mean reversion.[5] The Hull and White model allows closed-form solutions for European options on zero-coupon bonds. Jamshidian's approach can be used to price options on coupon bonds.

$$dr = \kappa\left(\frac{\theta(t)}{\kappa} - r\right)dt + \sigma dz, \tag{11.33}$$

where κ is the speed of mean reversion. $(\theta(t)/\kappa)$ is a time-dependent mean-reversion level.

Bond Prices
The price at time t of a discount bond maturing at time T is $P(t,T)$:

$$P(t,T) = A(t,T)e^{-B(t,T)r(t)},$$

where $r(t)$ is the rate at time t and

$$B(t,T) = \frac{1 - e^{-\kappa(T-t)}}{\kappa}$$

---

[5]Alternatively, it can be seen as an extension of the Vasicek (1977) model, with time-dependent drift. For more on this model, see also Hull and White (1992).

$$\ln A(t,T) = \ln\left[\frac{P(0,T)}{P(0,t)}\right] - B(t,T)\frac{\partial P(0,t)}{\partial t} - \frac{v(t,T)^2}{2}$$

$$v(t,T)^2 = \frac{1}{2\kappa^3}\sigma^2(e^{-\kappa T} - e^{-\kappa t})^2(e^{2\kappa t} - 1)$$

European Options
The value of a European option maturing at time T on a zero-coupon bond maturing at time τ is

$$c = P(0,\tau)N(h) - XP(0,T)N(h - v(T,\tau)) \tag{11.34}$$

$$p = XP(0,T)N(-h + v(T,\tau)) - P(0,\tau)N(-h), \tag{11.35}$$

where

$$h = \frac{1}{v(T,\tau)}\ln\left[\frac{P(0,\tau)}{P(0,T)X}\right] + \frac{v(T,\tau)}{2}$$

11.4.5 The Black-Derman-Toy Model

The Black, Derman, and Toy (1990) (BDT) one-factor model is one of the most used yield-based models to price bonds and interest rate options.[6] The model is arbitrage-free and thus consistent with the observed term structure of interest rates. Short rates are lognormally distributed at all times. This makes it difficult to find closed-form solutions for the options prices. The model must be implemented in a recombining binomial tree, for example. The short-rate volatility $\sigma(t)$ is potentially time-dependent, and the continuous process of the short-term interest rate is

$$d\ln(r) = \left[\theta(t) + \frac{\partial\sigma(t)/\partial t}{\sigma(t)}\ln(r)\right]dt + \sigma(t)dz, \tag{11.36}$$

where $\frac{\partial\sigma(t)/\partial t}{\sigma(t)}$ is the speed of mean reversion and $\theta(t)$ divided by the speed of mean reversion is a time-dependent mean-reversion level. The following example shows how to calibrate the BDT binomial tree to the current term structure of zero-coupon yields and zero-coupon volatilities.

Example
What is the value of an American call option on a five-year zero-coupon bond with time to expiration of four years and a strike price of 85.50? The term structure of zero-coupon rates and volatilities is shown in Table 11-2. From the rates and volatilities, we will calibrate the BDT interest rate tree. We will assume the risk-neutral probability of going up on any time step in the tree is $p = 0.5$, and similar the probability of going down on next time steps is $1 - p$.

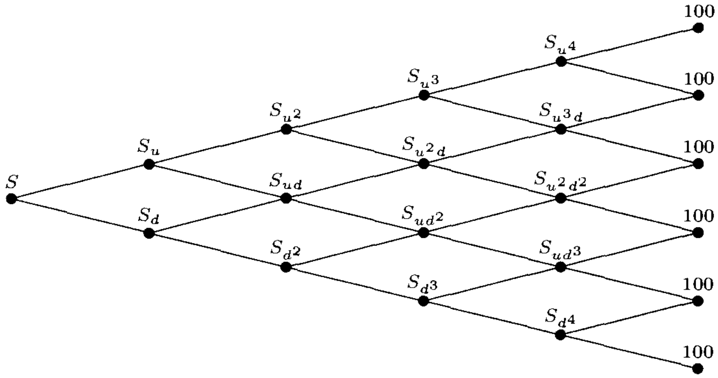
[6] Black and Karasinski (1991) generalize the BDT model.

TABLE 11-2

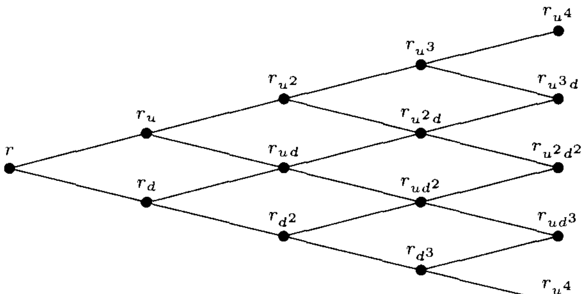
Input to BDT

Years to Maturity	Zero-Coupon Rates	Zero-Coupon Volatilities
1	9.0%	24%
2	9.5%	22%
3	10.0%	20%
4	10.5%	18%
5	11.0%	16%

To price the option by using backward induction, we build a tree for the bond prices, as shown:



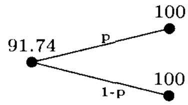
To build the price tree, we have to build the following rate tree:



We start by finding the price of a zero-coupon bond with maturity one year in the future:

$$91.74 = \frac{100 \times 0.5 + 100 \times 0.5}{(1 + 0.09)}$$

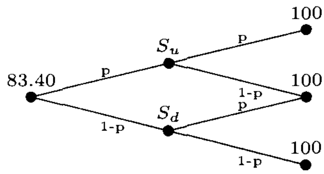
This gives us the one-period price tree:

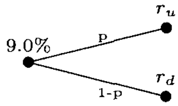


The next step is to build a two-period price tree. From the term structure of zero-coupon rates in Table 11-2, it is clear that the price today of a two-year zero-coupon bond with maturity two years from today must be

$$83.40 = \frac{100}{(1+0.095)^2}$$

To find the second-year bond prices at year one, we need to know the short rates at step one:





Appealing to risk-neutral valuation, the following relationship must hold:

$$83.40 = \frac{0.5S_u + 0.5S_d}{(1+0.09)} \tag{11.37}$$

In a standard binomial tree, we have

$$u = e^{\sigma\sqrt{T/n}} \quad d = e^{-\sigma\sqrt{T/n}}$$

$$u/d = e^{2\sigma\sqrt{T/n}}$$

$$\ln(u/d) = 2\sigma\sqrt{T/n}$$

$$\sigma = \frac{1}{2\sqrt{T/n}}\ln\left(\frac{u}{d}\right)$$

Similarly, in the BDT tree the rates are assumed to be lognormally distributed. This implies that

$$\sigma_n = \frac{1}{2\sqrt{T/n}}\ln\left(\frac{r_u}{r_d}\right) = 0.22$$

$$0.5\ln\left(\frac{r_u}{r_d}\right) = 0.22 \tag{11.38}$$

and

$$S_d = \frac{100}{1 + r_u} \tag{11.39}$$

$$S_u = \frac{100}{1 + r_d} \tag{11.40}$$

Now, substitute (11.39) and (11.40) into (11.37) to obtain

$$83.40 = \frac{0.5\left(\frac{100}{1+r_u}\right) + 0.5\left(\frac{100}{1+r_d}\right)}{(1 + 0.09)} \tag{11.41}$$

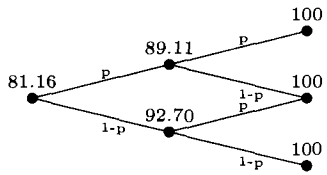
We are left with two equations, (11.38) and (11.41), in two unknowns, r_u and r_d. We know that $r_u = r_d e^{0.22\times2}$, which leads to the following quadratic equation:

$$83.40 = \frac{0.5\left(\frac{100}{1+r_d e^{0.22\times2}}\right) + 0.5\left(\frac{100}{1+r_d}\right)}{(1 + 0.09)}$$

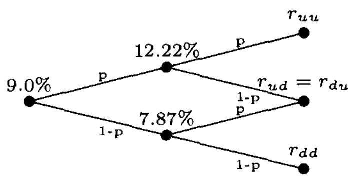
By solving this equation, we get the following rates at step one:

$$r_d = 7.87\% \qquad r_u = 12.22\%$$

Using these solutions, it is now possible to calculate the bond prices that correspond to these rates. The two-step tree of prices then becomes:



The next step is to fill in the two-period rate tree:



Last time, there were two unknown rates, and two sources of information:

1. Zero-coupon rates.
2. The volatility of the zero-coupon rates.

This time, we have three unknown rates, but still only two sources of information. To get around this problem, remember that the BDT model is built on the following assumptions:

- Rates are lognormally distributed.
- The volatility is only dependent on time, not on the level of the short rates. There is thus only one level of volatility at the same time step in the rate tree.

Hence,

$$0.5\ln(r_{uu}/r_{ud}) = 0.5\ln(r_{ud}/r_{dd})$$
$$\frac{r_{uu}}{r_{ud}} = \frac{r_{ud}}{r_{dd}}$$
$$r_{dd} = \frac{r_{ud}^2}{r_{uu}}$$

and we are left with only two unknowns. As with the one-period tree, we start by finding the bond price at the start of the price tree. In three years, the price of the bond must be 100, and the price today must be

$$75.13 = \frac{100}{(1+0.1)^3}$$

Based on the risk-neutral valuation principle, the following relationships must hold:

$$S_{uu} = \frac{100}{1+r_{uu}}, \quad S_{ud} = \frac{100}{1+r_{ud}}, \quad S_{dd} = \frac{100}{1+r_{dd}}, \tag{11.42}$$

$$S_u = \frac{0.5S_{uu} + 0.5S_{ud}}{(1+0.1222)}, \qquad S_d = \frac{0.5S_{dd} + 0.5S_{ud}}{(1+0.0787)}, \tag{11.43}$$

$$75.13 = \frac{0.5S_u + 0.5S_d}{(1+0.09)}. \tag{11.44}$$

If the bond only has two years left to maturity, the bond yield or rate of return must satisfy

$$S_u = \frac{100}{(1+y_u)^2} \quad \text{or} \quad S_d = \frac{100}{(1+y_d)^2}. \tag{11.45}$$

By solving equation (11.45) with respect to the bond yield, we get

$$y_u = \sqrt{\frac{100}{S_u}} - 1, \qquad y_d = \sqrt{\frac{100}{S_d}} - 1. \tag{11.46}$$

As the bond yields must be approximately lognormally distributed, it also follows that

$$0.5 \ln\left(\frac{y_u}{y_d}\right) = 0.20$$
$$\ln\left(\frac{y_u}{y_d}\right) = 0.40$$
$$\frac{y_u}{y_d} = e^{0.40} \tag{11.47}$$

With equations (11.46) and (11.47), y_u can be expressed as

$$y_u = \frac{y_u}{y_d} y_d,$$
$$y_u = e^{0.40}\left(\sqrt{\frac{100}{S_d}} - 1\right),$$

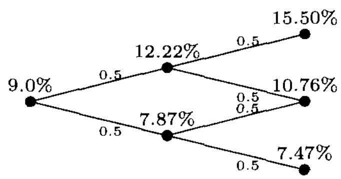
and S_u can be expressed in terms of S_d:

$$S_u = \frac{100}{\left[1 + e^{0.40}\left(\sqrt{\frac{100}{S_d}} - 1\right)\right]^2} \tag{11.48}$$

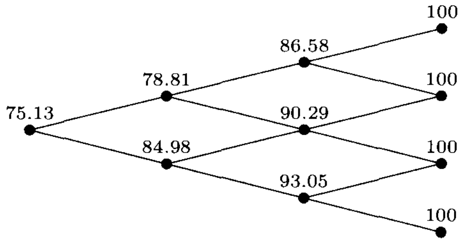
This equation must be solved by trial and error (possible, but not recommended) or, more efficiently, by the Newton–Raphson algorithm. The solution is

$$S_u = 78.81 \quad S_d = 84.98$$
$$r_{dd} = 7.47\% \quad r_{ud} = r_{du} = 10.76\% \quad r_{uu} = 15.50\%$$

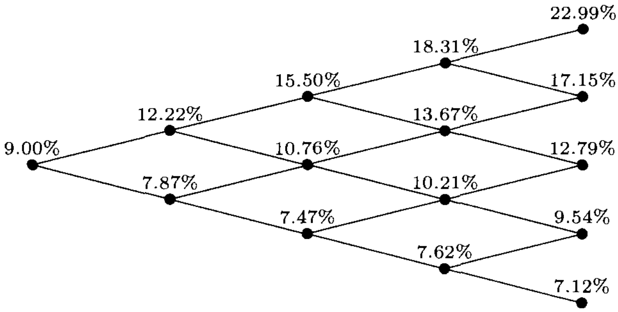
This gives the missing information in the two-period rate tree:



Now it is time to estimate the three-period price tree:



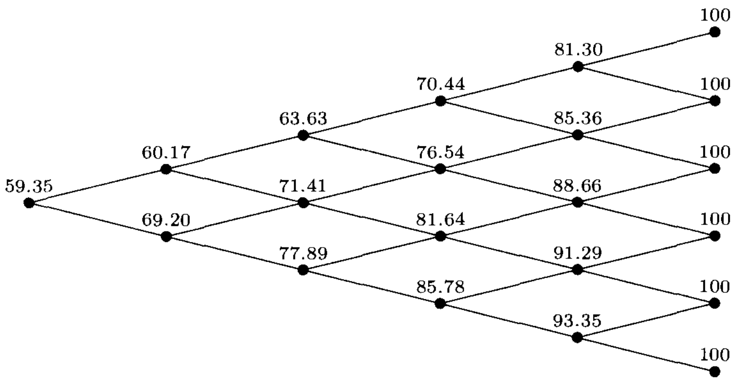
The consecutive time steps can be computed by forward induction, as introduced by Jamshidian (1991), or more easily with the Bjerksund and Stensland (1996) analytical approximation of the short-rate interest rate tree. Finally, we get the four-year short-rate tree:



From the short-rate tree, we can calculate the short-rate volatilities by using the relationship $\sigma_n = \frac{1}{2\sqrt{T/n}} \ln\left(\frac{r_u}{r_d}\right)$:

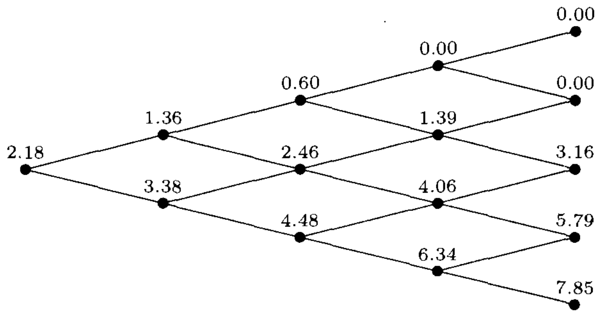
$$\sigma_0 = 24.00\% \quad \sigma_1 = 22.00\% \quad \sigma_2 = 18.24\% \quad \sigma_3 = 14.61\% \quad \sigma_4 = 14.66\%$$

The four-year rate tree supplies input to the solution to the five-year price tree:



The value of the American call option with strike 85.50 and time to expiration of four years can now easily be found by standard backward induction. It follows that

$$C_{j,i} = \max\left[S_{j,i} - 85.50, \frac{(0.5 \times C_{j+1,i+1} + 0.5 \times C_{j+1,i})}{(1 + r_{j,i})}\right]$$



The price of the American call option on the five-year bond is thus 2.18.

Black-Derman-Toy Yield Only

The BDT model can alternatively be calibrated to the yield curve only, and not also to the volatility curve. We can thus instead input a single volatility for the short rate. This simplified version of the BDT model often goes under the name "yield-only BDT." Below is a VBA code illustrating how to implement this simplified model.

Computer algorithm

The *BDTYiledOnly*(·) function returns the short rate tree in the form of a matrix when *ReturnFlag* = "*r*", and a discount factor tree if *ReturnFlag* = "*d*", and a Arrow-Debreu tree when *ReturnFlag* = "*a*".

```
Function BDTYieldOnly(ReturnFlag As String, v As Double, N As Integer, _
  T As Double, InputZeroRates As Variant, YieldMatu As Variant)

    Dim ZeroR() As Double, ZeroBond() As Double, u() As Double
    Dim r() As Double, Lambda() As Double, Df() As Double
    Dim dt As Double, epsilon As Double
    Dim Pi As Double, di As Double
    Dim i As Integer, j As Integer, m As Integer

    ReDim ZeroR(0 To N + 1)
    ReDim r(0 To N * 2, 0 To N * 2)
    ReDim Lambda(0 To N * 2, 0 To N * 2)
    ReDim ZeroBond(0 To N + 1)
    ReDim u(0 To N)
    ReDim Df(0 To N * 2, 0 To N * 2)

    dt = T / (N + 1)
```

```
For i = 1 To N + 1
    ZeroR(i) = InputZeroRates(i)
    ZeroBond(i) = 1 / (1 + ZeroR(i) * dt)^i
Next

Lambda(0, 0) = 1
u(0) = ZeroR(1)
r(0, 0) = ZeroR(1)
Df(0, 0) = 1 / (1 + r(0, 0) * dt)

For i = 1 To N
    '// Calculate the Arrow-Debreu prices by forward induction:
     ' //Arrow-Debreu  at lowest node:
    Lambda(i, 0) = 0.5 * Lambda(i - 1, 0) * Df(i - 1, 0)
    ' //Arrow-Debreu  at uppest node:
    Lambda(i, i) = 0.5 * Lambda(i - 1, i - 1) * Df(i - 1, i - 1)
    ' //Arrow-Debreu  between lowest and uppest node
    For j = 1 To i - 1
        Lambda(i, j) = 0.5 * Lambda(i - 1, j - 1) * Df(i - 1, j - 1) _
    + 0.5 * Lambda(i - 1, j) * Df(i - 1, j)
    Next

    '// Newton-Raphson method to find the unknown median u(i)
    u(i) = u(i - 1)      '// Seed value
    di = 0
    Pi = 0
    For j = 0 To i
        m = j * 2 - i
        Pi = Pi + Lambda(i, j) / (1 + u(i) _
          * Exp(v * m * Sqr(dt)) * dt)
        di = di - Lambda(i, j) * (Exp(v * m * Sqr(dt)) * dt) / (1 + u(i) _
    * Exp(v * m * Sqr(dt)) * dt)^2
   Next
    epsilon = 1e-09
    While Abs(Pi - ZeroBond(i + 1)) > epsilon
        u(i) = u(i) - (Pi - ZeroBond(i + 1)) / di
        di = 0
        Pi = 0
        For j = 0 To i
            m = -i + j * 2
            Pi = Pi + Lambda(i, j) / (1 + u(i) _
              * Exp(v * m * Sqr(dt)) * dt)
            di = di - Lambda(i, j) * (Exp(v * m * Sqr(dt)) * dt) _
    / (1 + u(i) * Exp(v * m * Sqr(dt)) * dt)^2
        Next
    Wend

    '// Given u(i) from the search above we can calculate the short
    '// rates and the corresponding discount factors
    For j = 0 To i
        m = (-i + j * 2)
        r(i, j) = u(i) * Exp(v * m * Sqr(dt))
        Df(i, j) = 1 / (1 + r(i, j) * dt)
    Next j
Next i

'// Output
'// Will return the short rate tree as a matrix:
```

```
    If ReturnFlag = "r" Then
        BDTYieldOnly = Application.Transpose(r())
    '// Will return the discount factor tree as a matrix:
    ElseIf ReturnFlag = "d" Then
            BDTYieldOnly = Application.Transpose(Df())
     '// Will return Arrow-Debreu tree as a matrix:
    ElseIf ReturnFlag = "a" Then
        BDTYieldOnly = Application.Transpose(Lambda())
    End If

End Function
```

CHAPTER

12

VOLATILITY AND CORRELATION

Volatility is blind to the sign of the move—not humans.

Nassim Taleb

This chapter deals with different ways to calculate volatility and correlation—an important topic, as these parameters are central to the valuation of most option contracts.

12.1 HISTORICAL VOLATILITY

12.1.1 Historical Volatility from Close Prices

Calculation of the annualized standard deviation is the most widely used method for estimating historical volatility. Standard deviation is simply the square root of the mean of the squared deviations of members of a sample (population) from their mean.

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\ln\left(\frac{Close_i}{Close_{i-1}}\right)^2 - \frac{1}{n(n-1)}\left[\sum_{i=1}^{n}\ln\left(\frac{Close_i}{Close_{i-1}}\right)\right]^2}, \qquad (12.1)$$

where n is the number of observations.

Example

Calculate the annualized volatility based on the close prices in Table 12-1.

$$\sigma = \sqrt{\frac{1}{20-1}\sum_{i=1}^{20}\ln\left(\frac{Close_i}{Close_{i-1}}\right)^2 - \frac{1}{20(20-1)}\left[\sum_{i=1}^{20}\ln\left(\frac{Close_i}{Close_{i-1}}\right)\right]^2} = 0.0173$$

When assuming 252 trading days in a year, the annualized close volatility is

$$\sigma = 0.0173\sqrt{252} = 0.2743.$$

TABLE 12-1

High, Low, and Close Prices

Day	Close	High	Low	$\ln(\text{Close}_i/\text{Close}_{i-1})$	ln(High/Low)
01.Oct.XX	132.5	132.5	131.0		0.011385
04.Oct.XX	133.5	134.0	131.0	0.007519	0.022642
05.Oct.XX	135.0	136.0	134.0	0.011173	0.014815
06.Oct.XX	133.0	137.0	133.0	−0.014926	0.029632
07.Oct.XX	133.0	136.0	133.0	0.000000	0.022306
08.Oct.XX	137.0	137.0	133.0	0.029632	0.029632
11.Oct.XX	135.0	136.5	135.0	−0.014706	0.011050
12.Oct.XX	135.0	136.0	135.0	0.000000	0.007380
13.Oct.XX	142.5	143.5	137.0	0.054067	0.046354
14.Oct.XX	143.0	145.0	142.0	0.003503	0.020907
15.Oct.XX	144.5	147.0	142.0	0.010435	0.034606
18.Oct.XX	145.0	147.5	145.0	0.003454	0.017094
19.Oct.XX	146.0	147.0	143.0	0.006873	0.027588
20.Oct.XX	149.0	150.0	148.0	0.020340	0.013423
21.Oct.XX	148.0	149.0	146.5	−0.006734	0.016921
22.Oct.XX	147.0	149.5	147.0	−0.006780	0.016864
25.Oct.XX	147.0	147.5	146.0	0.000000	0.010222
26.Oct.XX	147.0	149.0	146.5	0.000000	0.016921
27.Oct.XX	145.0	147.5	144.5	−0.013699	0.020549
28.Oct.XX	145.0	145.0	144.0	0.000000	0.006920
29.Oct.XX	150.0	150.0	143.5	0.033902	0.044300

Computer algorithm

```
Function CloseVolatility(ClosePrices As Object, _
Optional DataPerYear As Variant) As Double
    '
    ' Volatility (= standard deviation of logarithmic changes )
    ' Default adjustment is based on data for 252 days a year
    ' (Calendar day volatility)
    '
    If IsMissing(DataPerYear) Then
        DataPerYear = 252
    End If

    CloseVolatility = Application.StDev(LogChange(ClosePrices)) _
     * Sqr(DataPerYear)

End Function

Function LogChange(DataVector As Object)
    '
    ' Returns the natural logarithm of the changes in DataVector
    '
    Dim nRow As Integer
    Dim nCol As Integer
    Dim nVec As Integer
    Dim Element As Integer
    Dim TmpVec() As Double
    If DataVector.Areas.Count <> 1 Then
        ' Multiple selections not allowed
        LogChange = CVErr(xlErrValue)
    Else
```

```
        nRow = DataVector.Rows.Count
        nCol = DataVector.Columns.Count
        If (nRow = 1 And nCol >= 2) Or (nCol = 1 And nRow >= 2) Then
            nVec = Application.Max(nRow, nCol)
            ReDim TmpVec(nVec - 1)
            For Element = 1 To nVec - 1
                TmpVec(Element) = Log(DataVector(Element + 1) _
                / DataVector(Element))
            Next Element
            LogChange = Application.Transpose(TmpVec)
        Else
            ' DataVector is not a vector
            LogChange = CVErr(xlErrValue)
        End If
    End If

End Function
```

12.1.2 High-Low Volatility

Parkinson (1980) suggests estimating the standard deviation by

$$\sigma = \frac{1}{2n\sqrt{\ln(2)}} \sum_{i=1}^{n} \ln\left(\frac{High_i}{Low_i}\right) \tag{12.2}$$

Example

Calculate the annualized volatility based on the high and low prices found in Table 12-1.

$$\sigma = \frac{1}{2 \times 21\sqrt{\ln(2)}} \sum_{i=1}^{21} \ln\left(\frac{High_i}{Low_i}\right) = 0.0126$$

When assuming 252 trading days in a year, the annualized high-low volatility is

$$\sigma = 0.0126\sqrt{252} = 0.2004$$

The high-low method is statistically much more efficient than the standard close method.[1] However, it assumes continuous trading and observations of high and low prices. The method can therefore underestimate the true volatility.[2] The same is true for the high-low-close method described next.

Computer algorithm

This function returns the historical high-low volatility from a series of historical high and low prices. These can be daily prices, weekly

[1]In terms of number of observations needed to get the same interval compared with the standard close method.

[2]See Marsh and Rosenfield (1986). To get around some of its shortcomings, the high-low and the high-low-close methods have later been extended by Beckers (1983), Rogers and Satchell (1991), and Kunitomo (1992), among others.

prices, or of any other frequency. As default, it is assumed the input prices are daily prices and that there are 252 trading days per year. The function calls the *LogHighLow*(·) function, which calculates the natural logarithm of the high-low price series.

```
Function HighLowVolatility(HighPrices As Object, _
LowPrices As Object, Optional DataPerYear As Variant) As Double

  Dim n As Integer
  n = HighPrices.Rows.Count

  If IsMissing(DataPerYear) Then
      DataPerYear = 252
  End If
  HighLowVolatility = 1 / (2 * n * Sqr(Log(2))) _
    * Application.sum(LogHighLow(HighPrices, LowPrices)) _
    * Sqr(DataPerYear)

End Function

Function LogHighLow(HighPrices As Object, LowPrices As Object)

   Dim nRow As Integer
   Dim nCol As Integer
   Dim nVec As Integer
   Dim Element As Integer
   Dim TmpVec() As Double

   If HighPrices.Areas.Count <> 1 Or LowPrices.Areas.Count <> 1 Then
       ' Multiple selections not allowed
       LogHighLow = CVErr(xlErrValue)
   Else
       nRow = HighPrices.Rows.Count
       nCol = HighPrices.Columns.Count
       If (nRow = 1 And nCol >= 2) Or (nCol = 1 And nRow >= 2) Then
           nVec = Application.Max(nRow, nCol)
           ReDim TmpVec(nVec)
           For Element = 1 To nVec
               TmpVec(Element) = Log(HighPrices(Element) _
                 / LowPrices(Element))
           Next Element
           LogHighLow = Application.Transpose(TmpVec)
       Else
           ' Vector is not a vector
           LogHighLow = CVErr(xlErrValue)
       End If
   End If

End Function
```

12.1.3 High-Low-Close Volatility

Garman and Klass (1980) suggest using the estimator

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\left[\ln\left(\frac{High_i}{Low_i}\right)\right]^2 - \frac{1}{n}\sum_{i=1}^{n}[2\ln(2)-1]\left[\ln\left(\frac{Close_i}{Close_{i-1}}\right)\right]^2} \qquad (12.3)$$

Example
Calculate the annualized volatility based on the high, low, and close prices reported in Table 12-1.

$$\sigma = \sqrt{\frac{1}{20}\sum_{i=1}^{20}\frac{1}{2}\left[\ln\left(\frac{High_i}{Low_i}\right)\right]^2 - \frac{1}{20}\sum_{i=1}^{20}[2\ln(2)-1]\left[\ln\left(\frac{Close_i}{Close_{i-1}}\right)\right]^2}$$
$$= 0.0128$$

Assuming 252 trading days in a year, the annualized high-low-close volatility is

$$\sigma = 0.0128\sqrt{252} = 0.2038$$

Computer algorithm
This function returns the historical high-low-close volatility from a series of historical high, low, and close prices. These can, for example, be daily or weekly prices. As default, it is assumed the input prices are daily prices and that there are 252 trading days per year. The function calls the *LogHighLow*(·) function, which calculates the natural logarithm of the high-low price series.

```
Function HighLowCloseVolatility(HighPrices As Object, LowPrices As Object, _
    ClosePrices As Object, Optional DataPerYear As Variant) As Double

    Dim n As Integer
    n = HighPrices.Rows.Count
     If IsMissing(DataPerYear) Then
        DataPerYear = 252
    End If
    HighLowCloseVolatility = Sqr(1 / n * 1 / 2 _
    * Application.SumSq(LogHighLow(HighPrices, LowPrices)) _
    - 1 / n * (2 * Log(2) - 1) _
    * Application.SumSq(LogChange(ClosePrices))) * Sqr(DataPerYear)

End Function
```

12.1.4 Exponential Weighted Historical Volatility

Exponentially weighted volatility—also called exponentially weighted moving average volatility (EWMA)—puts more weight on more recent observations. An exponential moving average is given by

$$\frac{x_{t-1} + \lambda x_{t-2} + \lambda^2 x_{t-3} + \lambda^3 x_{t-4} + \cdots + \lambda^{n-1} x_{t-n}}{1 + \lambda + \lambda^2 + \lambda^3 + \cdots + \lambda^{n-1}},$$

where the x_i's are the observations in the time series and λ is a constant. The annualized exponential weighted volatility can be calculated as

$$\sigma_t^2 = \lambda\sigma_{t-1}^2 + (1-\lambda)[\ln(S_t/S_{t-1})]^2, \tag{12.4}$$

where σ_t is the current volatility and σ_{t-1} is the volatility as calculated one observation ago. If one is using daily data, the volatility is annualized by multiplying the result with the square root of the number of days per year. A rule of thumb is that in most markets λ should typically be between 0.75 and 0.98; see Alexander (2001). The popular RiskMetrics developed by J.P. Morgan uses an EWMA with $\lambda = 0.94$. Using $\lambda = 0.94$ and the close prices in Table 12-1 yields an EWMA of 24.56% when we use 252 trading days per year.

Computer algorithm

```
Function ExponentiallyWeightedVol(PriceVector As Object, _
  Lambda As Double, Optional DataPerYear As Variant) As Double

    Dim ExpVol As Double
    Dim nRow As Integer
    Dim nCol As Integer
    Dim nVec As Integer
    Dim Element As Integer

    If IsMissing(DataPerYear) Then
        DataPerYear = 252
    End If

    ExpVol = Log(PriceVector(2) / PriceVector(1))^2


    If PriceVector.Areas.Count <> 1 Then
        ' Multiple selections not allowed
        ExponentiallyWeightedVol = CVErr(xlErrValue)
    Else
        nRow = PriceVector.Rows.Count
        nCol = PriceVector.Columns.Count
        If (nRow = 1 And nCol >= 2) Or (nCol = 1 And nRow >= 2) Then
            nVec = Application.Max(nRow, nCol)
            For Element = 2 To nVec - 1
                ExpVol = Lambda * ExpVol + (1 - Lambda) _
          * Log(PriceVector(Element + 1) / PriceVector(Element))^2
            Next Element
            ExponentiallyWeightedVol = Sqr(ExpVol * DataPerYear)
        Else
            ' Vector is not a vector
            ExponentiallyWeightedVol = CVErr(xlErrValue)
        End If
    End If
End Function
```

12.1.5 From Annual Volatility to Daily Volatility

The market standard is to express a measure of volatility, whether historical or implied, as a standard deviation on an annual basis. In practice, a trader typically has a much better feeling for daily moves in the market. For this reason, many traders prefer to convert annual volatility to daily volatility. To do this, one simply divides the annualized volatility by the square root of the number of days in the year, alternatively trading days a year, n.

$$\sigma_{\text{daily}} = \frac{\sigma_{\text{annual}}}{\sqrt{n}} \tag{12.5}$$

Example
Assume a market maker tells you the volatility for a given option is 40% (implied Black-Scholes-Merton volatility). What is the equivalent daily volatility? With $\sigma = 0.4$, $n = 365$, we get

$$\sigma_{\text{daily}} = \frac{0.4}{\sqrt{365}} \approx 0.0209$$

The equivalent daily volatility is thus 2.09%.

12.1.6 Confidence Intervals for the Volatility Estimate

The next formula can be used to find the confidence intervals around the estimated close volatility (standard deviation), s, based on the number of observations, n, used in the estimate. The formula uses the chi-square distribution and is based on the assumption that the percentage changes in the asset price are normally distributed.

$$P\left[s\sqrt{\frac{(n-1)}{\chi^2_{(n-1;\alpha/2)}}} \le \sigma \le s\sqrt{\frac{(n-1)}{\chi^2_{(n-1;1-\alpha/2)}}}\right] = 1-\alpha, \tag{12.6}$$

where $\chi^2_{(n-1;\alpha/2)}$ is the value of the chi-square distribution with $n-1$ degrees of freedom and a confidence level of $1-\alpha$.

Example
Consider the 27.43% volatility estimate from the close prices in Table 12-1. The estimate is based on 21 close prices and 20 price changes. What is the 95% confidence interval ($\alpha = 0.05$) of this estimate? $s = 0.2743$, $\alpha = 0.05$, $n = 20$. Thus:

$$P\left[0.2743\sqrt{\frac{(20-1)}{\chi^2_{(20-1;0.05/2)}}} \le \sigma \le 0.2743\sqrt{\frac{(20-1)}{\chi^2_{(20-1;1-0.05/2)}}}\right] = 0.95$$

$$P[0.2086 \le \sigma \le 0.4006] = 0.95$$

With 20 observations there is a 95% probability that the real volatility will lie between 20.86% and 40.06%, based on an estimate of 27.43%.[3]

[3]That is, if we estimate the true volatility 100 times, with independent samples of size 20, the true volatility will lie within the confidence interval approximately 95 out of the 100 samples.

Comment

Every now and then I hear some traders, market makers, or salespeople talk about how low or high the volatility has been over last week, based on a volatility estimate involving five observations. Based on the sampling error from such an estimate, we can conclude that such statements are nonsense.

Computer algorithm

This function returns the confidence interval for the historical volatility estimate. *UpperLower* = "*L*" returns the lower confidence interval and *UpperLower* = "*U*" returns the upper confidence interval. n is the number of observations you have used to calculate the historical volatility.

```
ConfidenceIntervalVolatility(alfa As Double, n As Integer, _
    VolatilityEstimate As Double, UpperLower As String)

    'UpperLower  ="L" gives the lower cofidence interval
    'UpperLower  ="U" gives the upper cofidence interval
    'n: number of observations

    If UpperLower = "L" Then
        ConfidenceIntervalVolatility = VolatilityEstimate _
    * Sqr((n - 1) / (Application.ChiInv(alfa / 2, n - 1)))
    ElseIf UpperLower = "U" Then
        ConfidenceIntervalVolatility = VolatilityEstimate _
    * Sqr((n - 1) / (Application.ChiInv(1 - alfa / 2, n - 1)))
    End If

End Function
```

12.1.7 Volatility Cones

The volatility cone represents a method to determine if the current implied volatility is “cheap” or “expensive.” A volatility cone based on historical volatility finds the highest and lowest rolling historical volatility for different days/period. These are then typically plotted. If the current implied volatility for a given maturity is lower than the lowest observed volatility for that time period, the option is possibly underpriced. Similarly, if the implied volatility is above the highest observed volatility, it may be overpriced. The method has several obvious weaknesses and is only one of several tools one should keep in mind when analyzing volatility. The method does not take into account the volatility smile observed in many markets. It thus typically works best for at-the-money options. Volatility cones comes in many varieties. See Burghardt and Lane (1990), Haug (1992), and Hodges and Tompkins (2002) for a more detailed discussions.

Computer algorithm

This function returns the highest and lowest observed volatility for the various time periods, the average volatility over the period and the latest volatility for the period. This function called the *CloseVolatility*(·) function is given earlier in this chapter. See accompanying CD and spreadsheet to see examples of how to use this function.

```
Function VolatilityCone(DataVec As Object, VolPeriod As Integer, _
    Optional DataPerYear As Variant)
    '
    ' Volatility cone calculation with a volatility period,
    ' default adjustment is based on data for 252 days a year
    ' (trading days). Returns a vector which contains minimum,
    ' maximum, average and last volatility
    '
    Dim nRow As Integer, nCol As Integer, nVec As Integer
    Dim i As Integer, j As Integer, Elem As Integer
    Dim VolVec(4) As Double
    Dim TmpVec() As Double

    If IsMissing(DataPerYear) Then
        DataPerYear = 252
    End If
    If DataVec.Areas.Count <> 1 Then
        ' Multiple selections not allowed
        VolatilityCone = CVErr(xlErrValue)
    Else
        nRow = DataVec.Rows.Count
        nCol = DataVec.Columns.Count
        If (nRow = 1 And nCol >= 2) Or (nCol = 1 And nRow >= 2) Then
            nVec = Application.Max(nRow, nCol)
            If VolPeriod <= nVec - 1 Then
                ReDim TmpVec(nVec - VolPeriod)
                For j = 1 To nVec - VolPeriod
                    TmpVec(j) = VolatilityCone(DataVec.Range(Cells(j, 1), _
            Cells(j + VolPeriod, 1)), DataPerYear)
                Next j
                VolVec(1) = Application.Min(TmpVec)
                VolVec(2) = Application.Max(TmpVec)
                VolVec(3) = Application.Average(TmpVec)
                VolVec(4) = TmpVec(nVec - VolPeriod)
            Else
                ' Not enough data for a volatility period this long
                VolVec(1) = CVErr(xlErrValue)
            End If
            VolatilityCone = Application.Transpose(VolVec)
        Else
            ' DataVec or VolPeriods are not a vector
            VolatilityCone = CVErr(xlErrValue)
        End If
    End If

End Function
```

12.2 IMPLIED VOLATILITY

12.2.1 The Newton-Raphson Method

The Newton-Raphson method is an efficient way to find the implied volatility of an option contract. It is nothing more than a simple iteration technique for solving one-dimensional nonlinear equations (any introductory textbook in calculus will offer an intuitive explanation). The method seldom uses more than two to three iterations before it converges to the implied volatility. Let

$$\sigma_{i+1} = \sigma_i - \frac{c(\sigma_i) - c_m}{\partial c/\partial \sigma_i} \tag{12.7}$$

until $|c_m - c(\sigma_{i+1})| \leq \epsilon$, at which point σ_{i+1} is the implied volatility, ϵ is the desired degree of accuracy, c_m is the market price of the option, and $\partial c/\partial \sigma_i$ is the vega of the option evaluated at σ_i (the sensitivity of the option value for a small change in volatility).

Manaster and Koehler Start Value

Manaster and Koehler (1982) have developed an efficient seed value when the Newton-Raphson method is used to compute implied volatility. The seed value will guarantee convergence (if the implied volatility exists) for European Black-Scholes stock options. The seed value is simply

$$\sigma_1 = \sqrt{|\ln(S/X) + rT|\frac{2}{T}} \tag{12.8}$$

In the case of options on futures or forwards, the seed value should be set to

$$\sigma_1 = \sqrt{|\ln(F/X)|\frac{2}{T}} \tag{12.9}$$

Computer algorithm

This function returns the implied volatility of a European plain vanilla call or a put stock option.

```
Function GImpliedVolatilityNR(CallPutFlag As String, S As Double, X _
  As Double, T As Double, r As Double, b As Double, _
  cm As Double, epsilon As Double)

    Dim vi As Double, ci As Double
    Dim vegai As Double
    Dim minDiff As Double

    'Manaster and Koehler seed value (vi)
    vi = Sqr(Abs(Log(S / X) + r * T) * 2 / T)
    ci = GBlackScholes(CallPutFlag, S, X, T, r, b, vi)
    vegai = GVega(S, X, T, r, b, vi)
    minDiff = Abs(cm - ci)

    While Abs(cm - ci) >= epsilon And Abs(cm - ci) <= minDiff
        vi = vi - (ci - cm) / vegai
        ci = GBlackScholes(CallPutFlag, S, X, T, r, b, vi)
        vegai = GVega(S, X, T, r, b, vi)
        minDiff = Abs(cm - ci)
    Wend
```

```
    If Abs(cm - ci) < epsilon Then
        GImpliedVolatilityNR = vi
    Else
        GImpliedVolatilityNR = "NA"
    End If

End Function
```

12.2.2 The Bisection Method

The Newton-Raphson method requires knowledge of the partial derivative of the option pricing formula with respect to volatility (vega) when searching for the implied volatility. For some options (exotic and American options in particular), vega is not known analytically. The bisection method is an even simpler method to estimate implied volatility when vega is unknown. The bisection method requires two initial volatility estimates (seed values):

1. A "low" estimate of the implied volatility, σ_L, corresponding to an option value, c_L
2. A "high" volatility estimate, σ_H, corresponding to an option value, c_H

The option market price, c_m, lies between c_L and c_H. The bisection estimate is given as the linear interpolation between the two estimates:

$$\sigma_{i+1} = \sigma_L + (c_m - c_L)\frac{\sigma_H - \sigma_L}{c_H - c_L} \tag{12.10}$$

Replace σ_L with σ_{i+1} if $c(\sigma_{i+1}) < c_m$, or else replace σ_H with σ_{i+1} if $c(\sigma_{i+1}) > c_m$ until $|c_m - c(\sigma_{i+1})| \le \epsilon$, at which point σ_{i+1} is the implied volatility and ϵ is the desired degree of accuracy.

Computer algorithm

This function returns the implied volatility of a European plain vanilla call or put option. With small modifications, the function can also be used to find the implied volatility for American and exotic options. The variable *counter* keeps track of how many loops have been done. If the implied volatility is not found to the specified accuracy ϵ within 100 loops, the algorithm stops and returns "NA" (not available).

```
Function GBlackScholesImpVolBisection(CallPutFlag As String, S As Double, _
        X As Double, T As Double, r As Double, _
        b As Double, cm As Double) As Variant

    Dim vLow As Double, vHigh As Double, vi As Double
```

```
    Dim cLow As Double, cHigh As Double, epsilon As Double
    Dim counter As Integer

    vLow = 0.005
    vHigh = 4
    epsilon = 1e-08
    cLow = GBlackScholes(CallPutFlag, S, X, T, r, b, vLow)
    cHigh = GBlackScholes(CallPutFlag, S, X, T, r, b, vHigh)
    counter = 0

    vi = vLow + (cm - cLow) * (vHigh - vLow) / (cHigh - cLow)
    While Abs(cm - GBlackScholes(CallPutFlag, S, X, T, r, b, vi)) > epsilon
        counter = counter + 1
        If counter = 100 Then
            GBlackScholesImpVolBisection = "NA"
            Exit Function
        End If
        If GBlackScholes(CallPutFlag, S, X, T, r, b, vi) < cm Then
            vLow = vi
        Else
            vHigh = vi
        End If
        cLow = GBlackScholes(CallPutFlag, S, X, T, r, b, vLow)
        cHigh = GBlackScholes(CallPutFlag, S, X, T, r, b, vHigh)
        vi = vLow + (cm - cLow) * (vHigh - vLow) / (cHigh - cLow)
    Wend
    GBlackScholesImpVolBisection = vi

End Function
```

12.2.3 Implied Volatility Approximations

At-the-Money Forward Approximation

Brenner and Subrahmanyam (1988) and Feinstein (1988) suggest a simple formula that can be used to find the implied volatility of a plain vanilla option that is at-the-money forward:

$$\sigma \approx \frac{c_m\sqrt{2\pi}}{Se^{(b-r)T}\sqrt{T}}, \tag{12.11}$$

where c_m is the market price of an at-the-money-forward call or put option. At-the-money-forward is defined as $S = Xe^{-bT}$.

Example

Consider an at-the-money-forward stock call option with three months to expiration. The stock price is 59, the strike price is 60, the risk-free interest rate is 6.7% per year, and the market price of the option is 2.82. $S = 59$, $X = 60$, $T = 0.25$, $r = b = 0.067$, and $c_m = 2.82$. What is the implied volatility?

$$\sigma \approx \frac{2.82\sqrt{2\pi}}{59e^{(0.067-0.067)0.25}\sqrt{0.25}} = 23.96\%$$

For comparison, the exact implied volatility is 23.99%.

Extended Moneyness Approximation

The implied volatility approximation due to Corrado and Miller (1996a) extends the range of accuracy to a range of moneyness.[4] For a call option, the approximation is

$$\sigma \approx \frac{\sqrt{2\pi}}{Se^{(b-r)T} + Xe^{-rT}} \left\{ c_m - \frac{Se^{(b-r)T} - Xe^{-rT}}{2} + \left[\left(c_m - \frac{Se^{(b-r)T} - Xe^{-rT}}{2} \right)^2 - \frac{(Se^{(b-r)T} - Xe^{-rT})^2}{\pi} \right]^{\frac{1}{2}} \right\} / \sqrt{T}, \tag{12.12}$$

where c_m is the market price for a call option. The approximation for a put option is

$$\sigma \approx \frac{\sqrt{2\pi}}{Se^{(b-r)T} + Xe^{-rT}} \left\{ p_m - \frac{Xe^{-rT} - Se^{(b-r)T}}{2} + \left[\left(p_m - \frac{Xe^{-rT} - Se^{(b-r)T}}{2} \right)^2 - \frac{(Xe^{-rT} - Se^{(b-r)T})^2}{\pi} \right]^{\frac{1}{2}} \right\} / \sqrt{T}, \tag{12.13}$$

where p_m is the market price of a put option.

Example

Consider a put option with six months to expiration. The futures price is 108, the strike price is 100, the risk-free interest rate is 10.50% per year, and the market price of the put option is 5.08. $S = 108$, $X = 100$, $T = 0.5$, $r = 0.105$, $b = 0$, and $p_m = 5.08$. What is the implied volatility?

$$\sigma \approx \frac{\sqrt{2\pi}}{108e^{(0-0.105)0.5} + 100e^{-0.105\times 0.5}} \left\{ 5.08 - \frac{100e^{-0.105\times 0.5} - 108e^{(0-0.105)0.5}}{2} + \left[\left(5.08 - \frac{100e^{-0.105\times 0.5} - 108e^{(0-0.105)0.5}}{2} \right)^2 - \frac{(100e^{-0.105\times 0.5} - 108e^{(0-0.105)0.5})^2}{\pi} \right]^{\frac{1}{2}} \right\} / \sqrt{0.5} \approx 29.90\%$$

For comparison, the exact implied volatility is 30.00%.

[4]See also Corrado and Miller (1996b).

12.2.4 Implied Forward Volatility

Implied volatility is often considered the market's best estimate of future volatility. Similarly, the implied forward volatility can be seen as the market's best estimate of the volatility at a future date:

$$\sigma_F = \sqrt{\frac{\sigma_2^2 T_2 - \sigma_1^2 T_1}{T_2 - T_1}} \tag{12.14}$$

Haug and Haug (1996) show that implied forward volatility takes into consideration information embedded in the slope of the term structure of implied volatilities (in a Black-Scholes-Merton economy).

Example
Suppose we have a six-month option with 12% implied volatility, and a three-month option with 15% implied volatility. The implied three-month volatility three months forward is

$$\sigma_F = \sqrt{\frac{0.12^2 \times 0.5 - 0.15^2 \times 0.25}{0.5 - 0.25}} = 0.0794 = 7.94\%$$

By rearranging the implied forward volatility formula, it's possible to get a lower boundary of implied volatilities with time to maturity T_2, given an implied volatility with time to maturity T_1, where $T_2 > T_1$:

$$\sigma_2 = \sigma_1 \sqrt{\frac{T_1}{T_2}} \tag{12.15}$$

Example
Suppose we have a three-month option with 15% implied volatility. What is the lower boundary of the six-month implied volatility?

$$\sigma_2 = 0.15\sqrt{\frac{0.25}{0.5}} = 0.1061$$

Breakage of this lower bound signals a possible arbitrage opportunity.

12.2.5 From Implied Volatility Surface to Local Volatility Surface

Assume the following local volatility model, where the local volatility is a function of both the price level of the underlying asset and time:

$$dS = \mu(t)Sdt + \sigma(S, t)Sdz$$

Wilmott (1998) has published a useful formula for calculating local volatility from global volatility (BSM implied volatility surface):

$$\sigma(X, t) = \sqrt{\frac{\frac{\partial c}{\partial T} + bX\frac{\partial c}{\partial X} - (b + r)c}{\frac{1}{2}X^2\frac{\partial^2 c}{\partial X^2}}}$$

This gives us

$$\sigma(X,t)=\sqrt{\frac{\sigma^2+2(T-t)\sigma\frac{\partial\sigma}{\partial T}+2bX(T-t)\sigma\frac{\partial\sigma}{\partial X}}{\left(1+Xd_1\sqrt{T-t}\frac{\partial\sigma}{\partial X}\right)^2+X^2(T-t)\sigma\left(\frac{\partial^2\sigma}{\partial X^2}-d_1\left(\frac{\partial\sigma}{\partial X}\right)^2\sqrt{T-t}\right)}}, \quad (12.16)$$

where

$$d_1=\frac{\ln(S/X)+(b+\sigma^2/2)T}{\sigma\sqrt{T}}$$

The partial derivatives of the implied volatility surface can be found by numerical approximations, as discussed in Chapter 2. Because we in practice have a limited set of implied volatilities, it is typically necessary to interpolate or smooth the Black-Scholes-Merton volatility surface. However, the local volatility surface can be very sensitive to the input; see Wilmott (1998). What type of interpolation or surface smoothing is used can therefore be important. Rebonato (2004) offers more details on how to go from global to local volatility surface.

12.3 CONFIDENCE INTERVAL FOR THE ASSET PRICE

Given the assumption of lognormally distributed security prices, we can compute the confidence interval for the asset price S as

$$S_{min}=Se^{(b-\sigma^2/2)T-n\sigma\sqrt{T}} \quad (12.17)$$

and

$$S_{max}=Se^{(b-\sigma^2/2)T+n\sigma\sqrt{T}}, \quad (12.18)$$

where n is the standard deviation around the expected price (some commonly used standard deviations are $1=68.3\%$, $1.65=90\%$, $2=95.4\%$, and $3=99.7\%$).

Example
What is the confidence interval six months from now, with two standard deviations, for a stock that trades at 80 today? Let the risk-free interest rate be 8% per year and the volatility be 25% per year. $S=80$, $T=0.5$, $r=b=0.08$, $\sigma=0.25$, and $n=2$.

$$S_{min}=80e^{(0.08-0.25^2/2)0.5-2\times0.25\sqrt{0.5}}=57.5612$$

$$S_{max}=80e^{(0.08-0.25^2/2)0.5+2\times0.25\sqrt{0.5}}=116.7407$$

12.4 BASKET VOLATILITY

The volatility of the portfolio (basket) of several risky assets can be expressed as

$$\sigma_{Index} \approx \sqrt{\sum_{i=1}^{n} \sigma_i^2 Q_i^2 + 2\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} Q_i Q_j \rho_{i,j}\sigma_i\sigma_j}, \qquad (12.19)$$

where Q is the quantity/weight of the asset, $\rho_{i,j}$ is the correlation between the return of assets i and j and σ_i and σ_j are the volatility of the individual assets. It is sometimes suggested one use this volatility as input in the Black-76 option formula when valuing basket options. This means one assumes the basket is lognormally distributed. This is not consistent with the assumption that each asset is lognormally distributed but is often used as a rough approximation. See Allen (2003) for a more detailed discussion on this idea.

Computer algorithm

This function returns the volatility of basket of correlated assets.

```
Function BasketVolatility(Weights As Variant, Vols As Variant, _
    Correlations As Variant) As Double

        Dim n As Integer, i As Integer, j As Integer
        Dim sum As Double

        n = Application.Count(Weights)
        For i = 1 To n
                sum = sum + Weights(i)^2 * Vols(i)^2
            For j = i + 1 To n
                sum = sum + 2 * Weights(i) * Weights(j) _
        * Vols(i) * Vols(j) * Correlations(i, j)
            Next
        Next
        BasketVolatility = Sqr(sum)

End Function
```

12.5 HISTORICAL CORRELATION

This function returns the historical correlation from two price (time) series. It calculates the historical correlation for the natural logarithm of the price changes.

Computer algorithm

```
Function HistoricalCorrelation(PricesAsset1 As Object, _
        PricesAsset2 As Object) As Double

    HistoricalCorrelation = Application.Correl(LogChange(PricesAsset1), _
```

```
    LogChange( PricesAsset2 ))
End Function
```

12.5.1 Distribution of Historical Correlation Coefficient

If you run a two-asset Monte Carlo simulation and calculate the correlation coefficient, you will get a different correlation coefficient every time you run the simulation. The reason for this is the sampling. Only with continuous-time sampling will the correlation coefficient be uniquely determined. Based on the number of observations, n, as well as the population correlation coefficient ρ, Rao (1973) has published a method of how to calculate the theoretical distribution for the sample estimate of the correlation coefficient; see also Zhang (1998). When the population correlation coefficient is equal to zero $\rho = 0$, we have the following density function:

$$\eta(\gamma) = \frac{\Gamma[(n-1/2]}{\sqrt{\pi}\Gamma[(n-2)/2]}(1-\gamma^2)^{(n-4)/2}, \tag{12.20}$$

where ρ is the population correlation coefficient, γ is the estimated correlation coefficient, and n is the number of observations used to calculate the correlation coefficient. For $\rho \neq 0$ the density function is

$$\eta(\rho) = \frac{2^{n-3}}{\pi(n-3)!}(1-\rho^2)^{(n-1)/2}(1-\gamma^2)^{(n-4)}/2$$
$$\times \sum_{i=0}^{\infty} \Gamma^2\left(\frac{i+n-1}{2}\frac{(2\rho\gamma)^i}{i!}\right) \tag{12.21}$$

Even if this involves an infinite sum, it converges very fast and is therefore of practical use.

Computer algorithm

The function below returns the correlation density function.

```
Function CorrDen(n As Integer , gamma As Double , rho As Double) As Double
    Dim sum As Double
    Dim i As Integer

    If rho = 0 Then
        CorrDen = Exp(Application.GammaLn((n - 1) / 2)) _
        / (Sqr(Application.Pi()) * Exp(Application.GammaLn((n - 2) / 2))) _
        * (1 - gamma^2)^((n - 4) / 2)
    Else
    sum = 0
        For i = 0 To 10
```

```
            sum = sum + Exp(Application.GammaLn((n + i - 1) / 2))^2 _
    * (2 * rho * gamma) ^ i / Application.Fact(i)
        Next
            CorrDen = 2^(n - 3) / (Application.Pi() * Application.Fact(n - 3)) _
     * (1 - rho^2)^((n - 1) / 2) * (1 - gamma^2)^((n - 4) / 2) * sum
        End If

End Function
```

12.6 IMPLIED CORRELATIONS

12.6.1 Implied Correlation from Currency Options

Implied correlation from liquid currency options is useful as an estimate of future correlation (Haug, 1996).

$$\sigma_{EUR/JPY} = \sqrt{\sigma^2_{USD/EUR} + \sigma^2_{USD/JPY} - 2\rho\sigma_{USD/EUR}\sigma_{USD/JPY}} \tag{12.22}$$

$$\rho_{EUR/JPY} = \frac{\sigma^2_{USD/EUR} + \sigma^2_{USD/JPY} - \sigma^2_{EUR/JPY}}{2\sigma_{USD/EUR}\sigma_{USD/JPY}} \tag{12.23}$$

Example

Consider three currency options, all with six months to expiration. The implied volatility of the USD/EUR option is 14.90%, the implied volatility of the USD/JPY option is 15.30%, and the implied volatility of the EUR/JPY option is 12.30%. What is the implied correlation for the next six months between USD/EUR and USD/JPY? $\sigma_{USD/EUR} = 0.1490$, $\sigma_{USD/JPY} = 0.1530$, and $\sigma_{EUR/JPY} = 0.1230$.

$$\rho_{EUR/JPY} = \frac{0.1490^2 + 0.1530^2 - 0.1230^2}{2 \times 0.1490 \times 0.1530} = 0.6685$$

12.6.2 Average Implied Index Correlation

The volatility of a portfolio containing two risky assets:

$$\sigma = \sqrt{\sigma_1^2 Q_1^2 + \sigma_2^2 Q_2^2 + 2Q_1 Q_2 \rho\sigma_1\sigma_2}, \qquad \rho = \frac{\sigma^2 - \sigma_1^2 Q_1^2 - \sigma_2^2 Q_2^2}{2Q_1 Q_2 \sigma_1 \sigma_2},$$

where Q_1 is the quantity of asset 1 and Q_2 is the quantity of asset 2. ρ is the correlation between the return of assets 1 and 2. When the correlation coefficient is 0, 1, or -1, the formula for the volatility of a portfolio of two assets can be simplified to the following. (This is a special case of the basket volatility covered earlier in this chapter.)

$$\rho = 0: \qquad \sigma = \sqrt{\sigma_1^2 Q_1^2 + \sigma_2^2 Q_2^2}$$

$$\rho = 1: \qquad \sigma = \sigma_1 Q_1 + \sigma_2 Q_2$$

$$\rho = -1: \qquad \sigma = \sigma_1 Q_1 - \sigma_2 Q_2$$

The volatility of a portfolio containing three risky assets is

$$\sigma^2 = \sigma_1^2 Q_1^2 + \sigma_2^2 Q_2^2 + \sigma_3^2 Q_3^2 + 2Q_1Q_2\rho_{1,2}\sigma_1\sigma_2 + 2Q_1Q_3\rho_{1,3}\sigma_1\sigma_3 + 2Q_2Q_3\rho_{2,3}\sigma_2\sigma_3$$

The volatility of a portfolio containing several risky assets is

$$\sigma_{\text{Index}} = \sqrt{\sum_{i=1}^{n} \sigma_i^2 Q_i^2 + 2\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} Q_i Q_j \rho_{i,j}\sigma_i\sigma_j}$$

$$\rho_{\text{Average}} = \frac{\sigma_{\text{Index}}^2 - \sum_{i=1}^{n} \sigma_i^2 Q_i^2}{2\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} Q_i Q_j \sigma_i \sigma_j} \tag{12.24}$$

12.7 VARIOUS FORMULAS

12.7.1 Probability of High or Low, the Arctangent Rule

If we assume the returns follow a Brownian motion without drift, Acar and Toffel (1999) gives a formula for the probability that a high or a low occurs after a time $t = \tau$ out of a trading session T,

$$\frac{2}{\pi}\arctan\left(\frac{\sqrt{T-\tau}}{\sqrt{\tau}}\right), \tag{12.25}$$

where $0 <= \tau <= T$. In Excel the arctan function is: $= ATAN()$, so the whole formula could be written directly in Excel cell $A3$ as

$$= 2/PI() * ATAN(SQRT(A1 - A2)/SQRT(A2)),$$

assuming we have T in cell $A1$ and τ in cell $A2$. In the case of drift, no closed-form solution is available. Monte Carlo simulation is one way to solve it. See Acar and Toffel (1999) for more details.

Example
Assume a 10-hour trading day. What is the probability that we will have a high or low price in the last hour of the day, assuming zero drift and normally distributed returns? Intuition may suggest to you 1/10=10%, but the right answer is: $T = 10$, $\tau = 9$, yielding

$$\frac{2}{\pi}\arctan\left(\frac{\sqrt{10-9}}{\sqrt{9}}\right) = 0.205.$$

The probability of a new high or low in the last hour is thus 20.5%, *assuming* the asset follows a Brownian motion without drift.

12.7.2 Siegel's Paradox and Volatility Ratio Effect

We next consider the ratio of two realized volatilities, something that is partly related to Siegel's Paradox:

$$E\left[\frac{Rand()}{Rand()}\right] > \frac{E[Rand()]}{E[Rand()]},$$

where $E[]$ is the expectations operator, and $Rand()$ is a random number.

Example

Consider a coin toss for which there is a 50% probability of heads and 50% probability of tails. Next assume you are flipping two coins; heads is attached value 2, tails value 1. What is the expected ratio? Did you say one? Wrong, we have

- Tails and heads: $\frac{1}{2} * 0.25 = 0.125$
- Plus tails and tails: $\frac{1}{1} * 0.25 = 0.25$
- Plus heads and heads: $\frac{2}{2} * 0.25 = 0.25$
- Plus heads and tails: $\frac{2}{1} * 0.25 = 0.5$

The expected value is $0.125 + 0.25 + 0.25 + 0.5 = 1.125$. The same effect "naturally" shows up if you, for example, estimate volatility from a Monte Carlo simulation with constant volatility. Consider, for example, 20 observations. Calculate the volatility ratio for the first 10 days and for the 10 last days. Even if the problem is more complex than the coin example, the principle is the same; the expected value of the ratio is higher than expected volatility for one period divided by the expected volatility for the second period. Monte Carlo simulation shows that the expected ratio

$$\frac{E[\sigma_2^2]}{E[\sigma_1^2]} = 1 \qquad E\left[\frac{\sigma_2^2}{\sigma_1^2}\right] > 1 \qquad E\left[\frac{\sigma_1^2}{\sigma_2^2}\right] > 1$$

Using Monte Carlo simulation, we find that the expected value of the ratio in the example is about 1.07. For a different set of observations, the ratio would naturally be different. People more clever than myself can probably easily find an exact solution using probability theory.

Siegel's paradox can easily lead to confusion when hidden in derivatives instruments. A currency rate, for instance, can be seen as a ratio of two stochastic variables. For this reason, Siegel's paradox has received some attention in relation to currency options. See Dumas, Jennergren, and Näslund (1995) and Berdhan (1995).

CHAPTER

13

DISTRIBUTIONS

Quants don't crack jokes; they crack codes.

Quant

13.1 THE CUMULATIVE NORMAL DISTRIBUTION FUNCTION

The cumulative normal distribution function is given by the integral

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp(-z^2/2) dz \tag{13.1}$$

The integral has no closed-form solution, so a numerical approximation must be used. We present three possible approximations.[1] Recent research has shown that a high precision approximation can be of great importance in options valuation (West, 2005b). For this reason, we recommend the use of the Hart algorithm, which offers double precision. I have also included third- and fifth-degree polynomial approximations that are well known from the options literature. The polynomial approximations are known to suffer from inaccuracies far out in the tails and can in some special situations result in completely wrong options prices.

13.1.1 The Hart Algorithm

Hart (1968) presents an approximation that offers double precision throughout the real line (14 to 16 decimal places accuracy). This is the algorithm that I have chosen to use for all option pricing formulas that comes with the accompanying CD. The approximation is

[1]You can easily make your own, if you so desire. One common and simple approach is to express the integrand as a Taylor series. This is a polynomial that can easily be integrated. The approximation then involves truncating the resulting infinite series.

given by

$$N(x) = \begin{cases} e^{-y^2/2}\dfrac{A}{B}, & \text{when } y < 7.07106781186547 \\ e^{-y^2/2}\dfrac{1}{2.506628274631C} & \text{when } y \geq 7.07106781186547 \\ 0 & \text{when } y > 37 \end{cases}$$

where $y = |x|$. When $x > 0$ then $N(x) = 1 - N(x)$. Moreover,

$$A = ((((((a_1 y + a_2)y + a_3)y + a_4)y + a_5)y + a_6)y + a_7)$$

$$B = (((((((b_1 y + b_2)y + b_3)y + b_4)y + b_5)y + b_6)y + b_7)y + b_8)$$

$$C = y + 1/(y + 2/(y + 3/(y + 4/(y + 0.65))))$$

and

$$\begin{aligned} a_1 &= 0.0352624965998911 \\ a_2 &= 0.700383064443688 \\ a_3 &= 6.37396220353165 \\ a_4 &= 33.912866078383 \\ a_5 &= 112.079291497871 \\ a_6 &= 221.213596169931 \\ a_7 &= 220.206867912376 \end{aligned}$$

$$\begin{aligned} b_1 &= 0.0883883476483184 \\ b_2 &= 1.75566716318264 \\ b_3 &= 16.064177579207 \\ b_4 &= 86.7807322029461 \\ b_5 &= 296.564248779674 \\ b_6 &= 637.333633378831 \\ b_7 &= 793.826512519948 \\ b_8 &= 440.413735824752 \end{aligned}$$

Computer algorithm

The cumulative normal distribution function $CND(\cdot)$ returns values of $N(\cdot)$ to within double-precision accuracy.

```
Function CND(X As Double) As Double

    Dim y As Double, Exponential As Double
```

```
    Dim SumA As Double, SumB As Double

    y = Abs(X)

    If y > 37 Then
        CND = 0
    Else
        Exponential = Exp(-y^2/2)

        If y < 7.07106781186547 Then
            SumA = 0.0352624965998911 * y + 0.700383064443688
            SumA = SumA * y + 6.37396220353165
            SumA = SumA * y + 33.912866078383
            SumA = SumA * y + 112.079291497871
            SumA = SumA * y + 221.213596169931
            SumA = SumA * y + 220.206867912376
            SumB = 0.0883883476483184 * y + 1.75566716318264
            SumB = SumB * y + 16.064177579207
            SumB = SumB * y + 86.7807322029461
            SumB = SumB * y + 296.564248779674
            SumB = SumB * y + 637.333633378831
            SumB = SumB * y + 793.826512519948
            SumB = SumB * y + 440.413735824752
            CND = Exponential * SumA / SumB
        Else
            SumA = y + 0.65
            SumA = y + 4 / SumA
            SumA = y + 3 / SumA
            SumA = y + 2 / SumA
            SumA = y + 1 / SumA
            CND = Exponential / (SumA * 2.506628274631)
        End If

    End If

    If X > 0 Then CND = 1 - CND

End Function
```

13.1.2 Polynomial Approximations

The following approximation of the cumulative normal distribution function $N(x)$ produces values to within four-decimal-place accuracy.

$$N(x) = \begin{cases} 1 - n(x)(a_1 k + a_2 k^2 + a_3 k^3) & \text{when } x \geq 0 \\ 1 - N(-x) & \text{when } x < 0 \end{cases}$$

where

$$k = \frac{1}{1 + 0.33267x}$$

$$a_1 = 0.4361836$$

$$a_2 = -0.1201676$$

$$a_3 = 0.9372980$$

$$n(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

The next approximation, described by Abromowitz and Stegun (1974), produces values of $N(x)$ to within six decimal places of the true value.

$$N(x) = \begin{cases} 1 - n(x)(a_1 k + a_2 k^2 + a_3 k^3 + a_4 k^4 + a_5 k^5) & \text{when } x \geq 0 \\ 1 - N(-x) & \text{when } x < 0 \end{cases}$$

$$k = \frac{1}{1 + 0.2316419x}$$

$$a_1 = 0.319381530$$

$$a_2 = -0.356563782$$

$$a_3 = 1.781477937$$

$$a_4 = -1.821255978$$

$$a_5 = 1.330274429$$

Example

Calculate the cumulative normal probability for

$$d_1 = \frac{\ln(S/X) + (b + \sigma^2/2)T}{\sigma\sqrt{T}},$$

where $S = 88$, $X = 90$, $T = 0.5$, $r = 0.1$, $b = 0.1$, $\sigma = 0.4$.

$$d_1 = \frac{\ln(88/90) + (0.1 + 0.4^2/2)0.5}{0.4\sqrt{0.5}} = 0.2387$$

$$k = \frac{1}{1 + 0.33267 \times 0.2387} = 0.9264$$

$$n(d_1) = n(0.2387) = \frac{1}{\sqrt{2\pi}} e^{-0.2387^2/2} = 0.3877$$

$$N(d_1) = 1 - 0.3877(0.4361836 \times k + (-0.1201676) \times k^2 + 0.9372980 \times k^3) = 0.5943$$

Computer algorithm

The following cumulative normal distribution function $CND2(\cdot)$ returns values of $N(\cdot)$ to within six-decimal-place accuracy.

```
Function CND2(x As Double) As Double

    If x = 0 Then
```

```
        CND2 = 0.5
    Else
        Dim L As Double, k As Double
        Const a1 = 0.31938153:  Const a2 = -0.356563782
        Const a3 = 1.781477937: Const a4 = -1.821255978:
        Const a5 = 1.330274429

        L = Abs(x)
        k = 1 / (1 + 0.2316419 * L)
        CND2 = 1 - 1 / Sqr(2 * Pi) * Exp(-L^2 / 2) _
        * (a1 * k + a2 * k^2 + a3 * k^3 + a4 * k^4 + a5 * k^5)

        If x < 0 Then
            CND2 = 1 - CND2
        End If
    End If

End Function
```

13.2 THE INVERSE CUMULATIVE NORMAL DISTRIBUTION FUNCTION

The inverse cumulative normal distribution function is useful in several applications. In Monte Carlo simulations it is often used to transform uniformly distributed random numbers into normally distributed random numbers. The function is also needed in several closed-form calculations—for example, to find the strike from the delta in the BSM formula.

The cumulative normal distribution function is given by $N(x)$ given in expression (13.1). The inverse of $N(x)$ is found in the usual way by solving $N(x) = y$ for x as a function of y, where $0 \leq y \leq 1$. We will call the function $x(y)$ for $N^{-1}()$, which is more suggestive of the inversion operation. Moro (1995) has developed a fast and accurate approximation of the inverse cumulative normal distribution. Moro's approximation returns values of $N^{-1}(\cdot)$ to within largest absolute error of 3×10^{-9} accuracy for up to seven standard deviations. More accurate higher-order polynomial approximations, with up to 14 digits accuracy, can easily be implemented. These higher-order approximations will be much slower to compute, however.

Computer algorithm

Below is VBA code for the Moro (1995) inverse cumulative normal distribution approximation function $CNDEV(\cdot)$.

```
Option Base 0

Function CNDEV(U As Double) As Double

    Dim X As Double, r As Double
```

```
Dim A As Variant, b As Variant, c As Variant

A = Array(2.50662823884, -18.61500062529, 41.39119773534, -25.44106049637)
b = Array(-8.4735109309, 23.08336743743, -21.06224101826, 3.13082909833)
c = Array(0.337475482272615, 0.976169019091719, 0.160797971491821, _
0.0276438810333863, 0.0038405729373609, 0.0003951896511919, _
3.21767881767818e-05, 2.888167364e-07, 3.960315187e-07)

X = U - 0.5
If Abs(X) < 0.92 Then
    r = X * X
    r = X * (((A(3) * r + A(2)) * r + A(1)) * r + A(0)) _
    / ((((b(3) * r + b(2)) * r + b(1)) * r + b(0)) * r + 1)
    CNDEV = r
    Exit Function
End If

r = U
If X >= 0 Then r = 1 - U
r = Log(-Log(r))
r = c(0) + r * (c(1) + r * (c(2) + r * (c(3) + r + (c(4) + _
    r * (c(5) + r * (c(6) + r * (c(7) + r * c(8)))))))
If X < 0 Then r = -r
CNDEV = r

End Function
```

13.3 THE BIVARIATE NORMAL DENSITY FUNCTION

The bivariate normal density function is given by

$$F(x, y) = \frac{1}{2\pi\sqrt{1-\rho^2}}\exp\left[-\frac{1}{2(1-\rho^2)}(x^2 - 2\rho xy + y^2)\right] \tag{13.2}$$

13.3.1 The Cumulative Bivariate Normal Distribution Function

The standardized cumulative normal distribution function returns the probability that one random variable is less than a and that a second random variable is less than b when the correlation between the two variables is ρ:

$$M(a, b; \rho) = \frac{1}{2\pi\sqrt{1-\rho^2}}\int_{-\infty}^{a}\int_{-\infty}^{b}\exp\left[-\frac{x^2 - 2\rho xy + y^2}{2(1-\rho^2)}\right]dx \quad dy$$

Since no closed-form solution exists for the bivariate cumulative normal distribution, we present three approximations. The first one is the well-known Drezner (1978) algorithm. The second one is the more efficient Drezner and Wesolowsky (1990) algorithm. The third is the Genz (2004) algorithm, which is the most accurate one and therefore our recommended algorithm. West (2005b) and Agca and Chance (2003) discuss the speed and accuracy of bivariate normal distribution approximations for use in option pricing in more detail.

The Drezner 1978 Algorithm

Drezner (1978) has developed a method for approximating the cumulative bivariate normal distribution function. This approximation produces values of $M(a, b; \rho)$ to within six decimal places accuracy.

$$\phi(a, b; \rho) = \frac{\sqrt{1-\rho^2}}{\pi} \sum_{i=1}^{5} \sum_{j=1}^{5} x_i x_j f(y_i, y_j), \tag{13.3}$$

where

$$f(y_i, y_j) = \exp[a_1(2y_i - a_1) + b_1(2y_j - b_1) + 2\rho(y_i - a_1)(y_j - b_1)]$$

$$a_1 = \frac{a}{\sqrt{2(1-\rho^2)}}, \qquad b_1 = \frac{b}{\sqrt{2(1-\rho^2)}}$$

$$\begin{aligned} x_1 &= 0.24840615 & y_1 &= 0.10024215 \\ x_2 &= 0.39233107 & y_2 &= 0.48281397 \\ x_3 &= 0.21141819 & y_3 &= 1.0609498 \\ x_4 &= 0.033246660 & y_4 &= 1.7797294 \\ x_5 &= 0.00082485334 & y_5 &= 2.6697604 \end{aligned}$$

If the product of a, b, and ρ is nonpositive, compute the cumulative bivariate normal probability using the following rules:

1. If $a \le 0, b \le 0$, and $\rho \le 0$, then

$$M(a, b; \rho) = \phi(a, b; \rho)$$

2. If $a \le 0, b \ge 0$, and $\rho \ge 0$, then

$$M(a, b; \rho) = N(a) - \phi(a, -b; -\rho)$$

3. If $a \ge 0, b \le 0$, and $\rho \ge 0$, then

$$M(a, b; \rho) = N(b) - \phi(-a, b; -\rho)$$

4. If $a \ge 0, b \ge 0$, and $\rho \le 0$, then

$$M(a, b; \rho) = N(a) + N(b) - 1 + \phi(-a, -b; \rho)$$

In circumstances where the product of a, b, and ρ is positive, compute the cumulative bivariate normal function as

$$M(a, b; \rho) = M(a, 0; \rho_1) + M(b, 0; \rho_2) - \delta,$$

TABLE 13-1

The Cumulative Normal Distribution $N(x)$ when $x \le 0$

d	0.00	−0.01	−0.02	−0.03	−0.04	−0.05	−0.06	−0.07	−0.08	−0.09
−4.5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−4.4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−4.3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−4.2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−4.1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−4.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−3.9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−3.8	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
−3.7	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
−3.6	0.0002	0.0002	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
−3.5	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
−3.4	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0002
−3.3	0.0005	0.0005	0.0005	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0003
−3.2	0.0007	0.0007	0.0006	0.0006	0.0006	0.0006	0.0006	0.0005	0.0005	0.0005
−3.1	0.0010	0.0009	0.0009	0.0009	0.0008	0.0008	0.0008	0.0008	0.0007	0.0007
−3.0	0.0013	0.0013	0.0013	0.0012	0.0012	0.0011	0.0011	0.0011	0.0010	0.0010
−2.9	0.0019	0.0018	0.0018	0.0017	0.0016	0.0016	0.0015	0.0015	0.0014	0.0014
−2.8	0.0026	0.0025	0.0024	0.0023	0.0023	0.0022	0.0021	0.0021	0.0020	0.0019
−2.7	0.0035	0.0034	0.0033	0.0032	0.0031	0.0030	0.0029	0.0028	0.0027	0.0026
−2.6	0.0047	0.0045	0.0044	0.0043	0.0041	0.0040	0.0039	0.0038	0.0037	0.0036
−2.5	0.0062	0.0060	0.0059	0.0057	0.0055	0.0054	0.0052	0.0051	0.0049	0.0048
−2.4	0.0082	0.0080	0.0078	0.0075	0.0073	0.0071	0.0069	0.0068	0.0066	0.0064

−2.3	0.0107	0.0104	0.0102	0.0099	0.0096	0.0094	0.0091	0.0089	0.0087	0.0084
−2.2	0.0139	0.0136	0.0132	0.0129	0.0125	0.0122	0.0119	0.0116	0.0113	0.0110
−2.1	0.0179	0.0174	0.0170	0.0166	0.0162	0.0158	0.0154	0.0150	0.0146	0.0143
−2.0	0.0228	0.0222	0.0217	0.0212	0.0207	0.0202	0.0197	0.0192	0.0188	0.0183
−1.9	0.0287	0.0281	0.0274	0.0268	0.0262	0.0256	0.0250	0.0244	0.0239	0.0233
−1.8	0.0359	0.0351	0.0344	0.0336	0.0329	0.0322	0.0314	0.0307	0.0301	0.0294
−1.7	0.0446	0.0436	0.0427	0.0418	0.0409	0.0401	0.0392	0.0384	0.0375	0.0367
−1.6	0.0548	0.0537	0.0526	0.0516	0.0505	0.0495	0.0485	0.0475	0.0465	0.0455
−1.5	0.0668	0.0655	0.0643	0.0630	0.0618	0.0606	0.0594	0.0582	0.0571	0.0559
−1.4	0.0808	0.0793	0.0778	0.0764	0.0749	0.0735	0.0721	0.0708	0.0694	0.0681
−1.3	0.0968	0.0951	0.0934	0.0918	0.0901	0.0885	0.0869	0.0853	0.0838	0.0823
−1.2	0.1151	0.1131	0.1112	0.1093	0.1075	0.1056	0.1038	0.1020	0.1003	0.0985
−1.1	0.1357	0.1335	0.1314	0.1292	0.1271	0.1251	0.1230	0.1210	0.1190	0.1170
−1.0	0.1587	0.1562	0.1539	0.1515	0.1492	0.1469	0.1446	0.1423	0.1401	0.1379
−0.9	0.1841	0.1814	0.1788	0.1762	0.1736	0.1711	0.1685	0.1660	0.1635	0.1611
−0.8	0.2119	0.2090	0.2061	0.2033	0.2005	0.1977	0.1949	0.1922	0.1894	0.1867
−0.7	0.2420	0.2389	0.2358	0.2327	0.2296	0.2266	0.2236	0.2206	0.2177	0.2148
−0.6	0.2743	0.2709	0.2676	0.2643	0.2611	0.2578	0.2546	0.2514	0.2483	0.2451
−0.5	0.3085	0.3050	0.3015	0.2981	0.2946	0.2912	0.2877	0.2843	0.2810	0.2776
−0.4	0.3446	0.3409	0.3372	0.3336	0.3300	0.3264	0.3228	0.3192	0.3156	0.3121
−0.3	0.3821	0.3783	0.3745	0.3707	0.3669	0.3632	0.3594	0.3557	0.3520	0.3483
−0.2	0.4207	0.4168	0.4129	0.4090	0.4052	0.4013	0.3974	0.3936	0.3897	0.3859
−0.1	0.4602	0.4562	0.4522	0.4483	0.4443	0.4404	0.4364	0.4325	0.4286	0.4247
0.0	0.5000	0.4960	0.4920	0.4880	0.4840	0.4801	0.4761	0.4721	0.4681	0.4641

TABLE 13-2

The Cumulative Normal Distribution $N(x)$ when $x \geq 0$

d	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
2.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.1	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.2	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890

2.3	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.4	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.5	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952
2.6	0.9953	0.9955	0.9956	0.9957	0.9959	0.9960	0.9961	0.9962	0.9963	0.9964
2.7	0.9965	0.9966	0.9967	0.9968	0.9969	0.9970	0.9971	0.9972	0.9973	0.9974
2.8	0.9974	0.9975	0.9976	0.9977	0.9977	0.9978	0.9979	0.9979	0.9980	0.9981
2.9	0.9981	0.9982	0.9982	0.9983	0.9984	0.9984	0.9985	0.9985	0.9986	0.9986
3.0	0.9987	0.9987	0.9987	0.9988	0.9988	0.9989	0.9989	0.9989	0.9990	0.9990
3.1	0.9990	0.9991	0.9991	0.9991	0.9992	0.9992	0.9992	0.9992	0.9993	0.9993
3.2	0.9993	0.9993	0.9994	0.9994	0.9994	0.9994	0.9994	0.9995	0.9995	0.9995
3.3	0.9995	0.9995	0.9995	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	0.9997
3.4	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9998
3.5	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998
3.6	0.9998	0.9998	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
3.7	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
3.8	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
3.9	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4.0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4.1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4.2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4.3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4.4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4.5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

where $M(a, 0; \rho_1)$ and $M(b, 0; \rho_2)$ are computed from the rules where the product of a, b, and ρ is negative, and

$$\rho_1 = \frac{(\rho a - b)Sign(a)}{\sqrt{a^2 - 2\rho ab + b^2}} \qquad \rho_2 = \frac{(\rho b - a)Sign(b)}{\sqrt{a^2 - 2\rho ab + b^2}}$$

$$\delta = \frac{1 - Sign(a) \times Sign(b)}{4} \qquad Sign(x) = \begin{cases} +1 & \text{when } x \geq 0 \\ -1 & \text{when } x < 0 \end{cases}$$

Computer algorithm

The *CBND3(a, b, rho)* function returns the standardized bivariate normal probability that the first variable is less than a and the second variable is less than b, where *rho* is the correlation between the variables.

```
Function CBND3(A As Double, b As Double, rho As Double) As Double

    Dim X As Variant, y As Variant
    Dim rho1 As Double, rho2 As Double, delta As Double
    Dim a1 As Double, b1 As Double, sum As Double
    Dim i As Integer, j As Integer

    X = Array(0.24840615, 0.39233107, 0.21141819, 0.03324666, 0.00082485334)
    y = Array(0.10024215, 0.48281397, 1.0609498, 1.7797294, 2.6697604)
    a1 = A / Sqr(2 * (1 - rho^2))
    b1 = b / Sqr(2 * (1 - rho^2))

    If A <= 0 And b <= 0 And rho <= 0 Then
        sum = 0
        For i = 1 To 5
            For j = 1 To 5
                sum = sum + X(i) * X(j) * Exp(a1 * (2 * y(i) - a1) _
                + b1 * (2 * y(j) - b1) + 2 * rho * (y(i) - a1) * (y(j) - b1))
            Next
        Next
        CBND3 = Sqr(1 - rho^2) / Pi * sum
    ElseIf A <= 0 And b >= 0 And rho >= 0 Then
        CBND3 = CND(A) - CBND3(A, -b, -rho)
    ElseIf A >= 0 And b <= 0 And rho >= 0 Then
        CBND3 = CND(b) - CBND3(-A, b, -rho)
    ElseIf A >= 0 And b >= 0 And rho <= 0 Then
        CBND3 = CND(A) + CND(b) - 1 + CBND3(-A, -b, rho)
    ElseIf A * b * rho > 0 Then
        rho1 = (rho * A - b) * Sgn(A) / Sqr(A^2 - 2 * rho * A * b + b^2)
        rho2 = (rho * b - A) * Sgn(b) / Sqr(A^2 - 2 * rho * A * b + b^2)
        delta = (1 - Sgn(A) * Sgn(b)) / 4
        CBND3 = CBND3(A, 0, rho1) + CBND3(b, 0, rho2) - delta
    End If

End Function
```

The Drezner and Wesolowsky 1990 Algorithm

Drezner and Wesolowsky (1990) suggest two algorithms for calculating the bivariate cumulative normal distribution function. Their first algorithm is much simpler and four to five times as fast as the Drezner (1978) algorithm. However, West (2005b) has carefully tested out both the Drezner and Wesolowsky (1990) algorithms and showed that they can run into serious problems for certain combinations of parameters and options contracts. I still give a short presentation of the algorithms here. The simplest version of the Drezner and Wesolowsky (1990) algorithm is

$$\phi(a, b; \rho) = N(a)N(b) + \rho \sum_{j=1}^{5} \frac{x_i e^{\frac{2aby_i\rho - a^2 - b^2}{2(1-y_i^2\rho^2)}}}{\sqrt{1 - y_i^2\rho^2}}, \tag{13.4}$$

where $N(\cdot)$ is the cumulative normal distribution function given earlier in this chapter and

$$\begin{aligned} x_1 &= 0.018854042 & y_1 &= 0.04691008 \\ x_2 &= 0.038088059 & y_2 &= 0.23076534 \\ x_3 &= 0.0452707394 & y_3 &= 0.5 \\ x_4 &= 0.038088059 & y_4 &= 0.76923466 \\ x_5 &= 0.018854042 & y_5 &= 0.95308992 \end{aligned}$$

Computer algorithm

The *CBND2*(a, b, rho) function returns the standardized bivariate normal probability that the first variable is less than a and the second variable is less than b, where rho is the correlation between the variables.

```
Option Base 0

Function CBND2(A As Double, b As Double, rho As Double) As Double

  Dim g As Double, P As Double, x, y, sum As Double
  Dim i As Integer

   x = Array(0.018854042, 0.038088059, 0.0452707394, _
     0.038088059, 0.018854042)
   y = Array(0.04691008, 0.23076534, 0.5, 0.76923466, 0.95308992)

   sum = 0
   For i = 0 To 4
       P = y(i) * rho
       g = 1 - P ^ 2
       sum = sum + x(i) * Exp((2 * A * b * P _
        - A ^ 2 - b ^ 2) / g / 2) / Sqr(g)
   Next

   CBND2 = rho * sum + CND(A) * CND(b)

End Function
```

The Genz 2004 Algorithm

Genz (2004) has provided a modification of the Drezner and Wesolowsky (1990) single-precision algorithm with nearly double precision (about 14-decimal accuracy). This is the recommended bivariate algorithm and the one used for option valuation on the accompanying CD.

Computer algorithm

The computer algorithm below is based on the original FORTRAN code by Genz (2004) and was converted into VBA by West (2005b). The *CBND(x, y, rho)* function returns the standardized bivariate normal probability that the first variable is less than *x* and the second variable is less than *y*, where *rho* is the correlation between the variables.

```
Function CBND(X As Double, y As Double, rho As Double) As Double
'
'     A function for computing bivariate normal probabilities.
'
'       Alan Genz
'       Department of Mathematics
'       Washington State University
'       Pullman, WA 99164-3113
'       Email : alangenz@wsu.edu
'
'   This function is based on the method described by
'   Drezner, Z and G.O. Wesolowsky, (1990),
'   On the computation of the bivariate normal integral,
'   Journal of Statist. Comput. Simul. 35, pp. 101-107,
'   with major modifications for double precision,
'   and for |R| close to 1.
' This code was originally translated into VBA by Graeme West

    Dim i As Integer, ISs As Integer, LG As Integer, NG As Integer
    Dim XX(10, 3) As Double, W(10, 3) As Double
    Dim h As Double, k As Double, hk As Double, hs As Double
    Dim BVN As Double, Ass As Double, asr As Double, sn As Double
    Dim A As Double, b As Double, bs As Double, c As Double, d As Double
    Dim xs As Double, rs As Double

    W(1, 1) = 0.17132449237917
    XX(1, 1) = -0.932469514203152
    W(2, 1) = 0.360761573048138
    XX(2, 1) = -0.661209386466265
    W(3, 1) = 0.46791393457269
    XX(3, 1) = -0.238619186083197

    W(1, 2) = 0.0471753363865118
    XX(1, 2) = -0.981560634246719
    W(2, 2) = 0.106939325995318
    XX(2, 2) = -0.904117256370475
    W(3, 2) = 0.160078328543346
    XX(3, 2) = -0.769902674194305
    W(4, 2) = 0.203167426723066
    XX(4, 2) = -0.587317954286617
    W(5, 2) = 0.233492536538355
    XX(5, 2) = -0.36783149899818
    W(6, 2) = 0.249147045813403
    XX(6, 2) = -0.125233408511469
```

```
W(1, 3) = 0.0176140071391521
XX(1, 3) = -0.993128599185095
W(2, 3) = 0.0406014298003869
XX(2, 3) = -0.963971927277914
W(3, 3) = 0.0626720483341091
XX(3, 3) = -0.912234428251326
W(4, 3) = 0.0832767415767048
XX(4, 3) = -0.839116971822219
W(5, 3) = 0.10193011981724
XX(5, 3) = -0.746331906460151
W(6, 3) = 0.118194531961518
XX(6, 3) = -0.636053680726515
W(7, 3) = 0.131688638449177
XX(7, 3) = -0.510867001950827
W(8, 3) = 0.142096109318382
XX(8, 3) = -0.37370608871542
W(9, 3) = 0.149172986472604
XX(9, 3) = -0.227785851141645
W(10, 3) = 0.152753387130726
XX(10, 3) = -0.0765265211334973

If Abs(rho) < 0.3 Then
  NG = 1
  LG = 3
ElseIf Abs(rho) < 0.75 Then
  NG = 2
  LG = 6
Else
  NG = 3
  LG = 10
End If

h = -X
k = -y
hk = h * k
BVN = 0

If Abs(rho) < 0.925 Then
  If Abs(rho) > 0 Then
    hs = (h * h + k * k) / 2
    asr = ArcSin(rho)
    For i = 1 To LG
      For ISs = -1 To 1 Step 2
        sn = Sin(asr * (ISs * XX(i, NG) + 1) / 2)
        BVN = BVN + W(i, NG) * Exp((sn * hk - hs) / (1 - sn * sn))
      Next ISs
    Next i
    BVN = BVN * asr / (4 * Pi)
  End If
  BVN = BVN + CND(-h) * CND(-k)
Else
  If rho < 0 Then
    k = -k
    hk = -hk
  End If
  If Abs(rho) < 1 Then
    Ass = (1 - rho) * (1 + rho)
    A = Sqr(Ass)
    bs = (h - k)^2
```

```
        c = (4 - hk) / 8
        d = (12 - hk) / 16
        asr = -(bs / Ass + hk) / 2
       If asr > -100 Then BVN = A * Exp(asr) * (1 - c * (bs - Ass) _
       * (1 - d * bs / 5) / 3 + c * d * Ass * Ass / 5)
        If -hk < 100 Then
          b = Sqr(bs)
          BVN = BVN - Exp(-hk / 2) * Sqr(2 * Pi) * CND(-b / A) _
         * b * (1 - c * bs * (1 - d * bs / 5) / 3)
       End If
        A = A / 2
        For i = 1 To LG
        For ISs = -1 To 1 Step 2
           xs = (A * (ISs * XX(i, NG) + 1))^2
            rs = Sqr(1 - xs)
          asr = -(bs / xs + hk) / 2
          If asr > -100 Then
               BVN = BVN + A * W(i, NG) * Exp(asr) _
               * (Exp(-hk * (1 - rs) / (2 * (1 + rs))) / rs _
               - (1 + c * xs * (1 + d * xs)))
            End If
          Next ISs
        Next i
        BVN = -BVN / (2 * Pi)
     End If
      If rho > 0 Then
        BVN = BVN + CND(-Max(h, k))
      Else
        BVN = -BVN
       If k > h Then BVN = BVN + CND(k) - CND(h)
     End If
    End If
    CBND = BVN

End Function

Function ArcSin(X As Double) As Double

  If Abs(X) = 1 Then
    ArcSin = Sgn(X) * Pi / 2
  Else
    ArcSin = Atn(X / Sqr(1 - X^2))
  End If

End Function
```

Table Values Cumulative Bivariate Function

Those who wish to retype or program any of the cumulative bivariate functions on their own can check their results against Table 13-3 on p. 481. In the bottom of the column named Drezner-78, you will find "#VALUE!"; this is because the Drezner-78 algorithm was not able to calculate any values for these input parameters.

TABLE 13-3

Bivariate Normal Probabilities

a	b	ρ	Drezner-78	DreWes-1	DreWes-2	Genze
0	0	0	0.2499999988	0.2500000000	0.2500000000	0.2500000000
0	0	−0.5	0.1666666617	0.1666666679	0.1666666679	0.1666666667
0	0	0.5	0.3333333383	0.3333333321	0.3333333321	0.3333333333
0	−0.5	0	0.1542687690	0.1542687694	0.1542687694	0.1542687694
0	−0.5	−0.5	0.0816597373	0.0816597613	0.0816597613	0.0816597607
0	−0.5	0.5	0.2268778014	0.2268777774	0.2268777774	0.2268777781
0	0.5	0	0.3457312310	0.3457312306	0.3457312306	0.3457312306
0	0.5	−0.5	0.2731221986	0.2731222226	0.2731222226	0.2731222219
0	0.5	0.5	0.4183402627	0.4183402387	0.4183402387	0.4183402393
−0.5	0	0	0.1542687690	0.1542687694	0.1542687694	0.1542687694
−0.5	0	−0.5	0.0816597373	0.0816597613	0.0816597613	0.0816597607
−0.5	0	0.5	0.2268778014	0.2268777774	0.2268777774	0.2268777781
−0.5	−0.5	0	0.0951954128	0.0951954128	0.0951954128	0.0951954128
−0.5	−0.5	−0.5	0.0362981449	0.0362981869	0.0362981869	0.0362981865
−0.5	−0.5	0.5	0.1633194746	0.1633195203	0.1633195203	0.1633195213
−0.5	0.5	0	0.2133421259	0.2133421259	0.2133421259	0.2133421259
−0.5	0.5	−0.5	0.1452180641	0.1452180185	0.1452180185	0.1452180174
−0.5	0.5	0.5	0.2722393939	0.2722393519	0.2722393519	0.2722393522
0.5	0	0	0.3457312310	0.3457312306	0.3457312306	0.3457312306
0.5	0	−0.5	0.2731221986	0.2731222226	0.2731222226	0.2731222219
0.5	0	0.5	0.4183402627	0.4183402387	0.4183402387	0.4183402393
0.5	−0.5	0	0.2133421259	0.2133421259	0.2133421259	0.2133421259
0.5	−0.5	−0.5	0.1452180641	0.1452180185	0.1452180185	0.1452180174
0.5	−0.5	0.5	0.2722393939	0.2722393519	0.2722393519	0.2722393522
0.5	0.5	0	0.4781203354	0.4781203354	0.4781203354	0.4781203354
0.5	0.5	−0.5	0.4192230674	0.4192231094	0.4192231094	0.4192231090
0	−0.9999999	−0.9999999	0.0000000000	0.0001874324	0.0000000000	0.0000000000
0.00001	−0.9999999	−0.9999999	#VALUE!	0.0001874671	0.0000000000	0.0000000000

13.4 THE TRIVARIATE CUMULATIVE NORMAL DISTRIBUTION FUNCTION

For certain exotic options, one needs the trivariate cumulative normal distribution function. I will not go into detail on this function here but simply give you the VBA algorithm. The main function $CTND(\cdot)$ returns the trivariate cumulative normal probability. The function also calls the bivariate cumulative normal distribution function $CBND(\cdot)$ given in this chapter.

Computer algorithm

```
Function CTND(LIMIT1 As Double, LIMIT2 As Double, LIMIT3 As Double, _
        SIGMA1 As Double, SIGMA2 As Double, SIGMA3 As Double) As Double
'
'     A function for computing trivariate normal probabilities.
'     This function uses an algorithm given in the paper
'        "Numerical Computation of Bivariate and
'              Trivariate Normal Probabilities",
'     by
'       Alan Genz
'       Department of Mathematics
'       Washington State University
'       Pullman, WA 99164-3113
'       Email : alangenz@wsu.edu
'
'   Thanks to Graeme West for help with VBA version
'
' CTND calculates the probability that X(I) < LIMIT(I), I = 1, 2, 3.
'
' Parameters
'
'   LIMIT  DOUBLE PRECISION array of three upper integration limits.
'   SIGMA  DOUBLE PRECISION array of three correlation coefficients,
'          SIGMA should contain the lower left portion of the
'          correlation matrix R.
'          SIGMA(1) = R(2,1), SIGMA(2) = R(3,1), SIGMA(3) = R(3,2).
'
'    CTND cuts the outer integral over -infinity to B1 to
'      an integral from -8.5 to B1 and then uses an adaptive
'      integration method to compute the integral of a bivariate
'      normal distribution function.
'
Dim TAIL As Boolean
'
'     Bivariate normal distribution function CBND is required.
'
Dim SQ21 As Double, SQ31 As Double, rho As Double
Dim B1 As Double, B2 As Double, B3 As Double, b2p As Double, b3p As Double
Dim RHO21 As Double, RHO31 As Double, RHO32 As Double
Const SQTWPI = 2.506628274631
Const XCUT = -8.5
Const EPS = 5e-16

        'COMMON /TRVBKD/B2P, B3P, RHO21, RHO31, RHO
```

```
    B1 = LIMIT1
    B2 = LIMIT2
    B3 = LIMIT3
    RHO21 = SIGMA1
    RHO31 = SIGMA2
    RHO32 = SIGMA3
    If Abs(B2) > Max(Abs(B1), Abs(B3)) Then
       B1 = B2
       B2 = LIMIT1
       RHO31 = RHO32
       RHO32 = SIGMA2
    ElseIf Abs(B3) > Max(Abs(B1), Abs(B2)) Then
       B1 = B3
       B3 = LIMIT1
       RHO21 = RHO32
       RHO32 = SIGMA1
    End If

  TAIL = False

    If B1 > 0 Then
       TAIL = True
       B1 = -B1
       RHO21 = -RHO21
       RHO31 = -RHO31
    End If

    If B1 > XCUT Then
       If 2 * Abs(RHO21) < 1 Then
          SQ21 = Sqr(1 - RHO21^2)
       Else
          SQ21 = Sqr((1 - RHO21) * (1 + RHO21))
       End If

       If 2 * Abs(RHO31) < 1 Then
          SQ31 = Sqr(1 - RHO31^2)
       Else
          SQ31 = Sqr((1 - RHO31) * (1 + RHO31))
       End If

       rho = (RHO32 - RHO21 * RHO31) / (SQ21 * SQ31)
       b2p = B2 / SQ21
       RHO21 = RHO21 / SQ21
       b3p = B3 / SQ31
       RHO31 = RHO31 / SQ31
       CTND = ADONED(XCUT, B1, EPS, b2p, b3p, RHO21, RHO31, rho) / SQTWPI
    Else
       CTND = 0
    End If

    If TAIL = True Then
      CTND = CBND(B2, B3, RHO32) - CTND
    End If

End Function


Function ADONED(A As Double, b As Double, TOL As Double, _
            b2p As Double, b3p As Double, RHO21 As Double, _
```

```
        RHO31 As Double, rho As Double) As Double
'
'       One Dimensional Globally Adaptive Integration Function
'

Dim i As Integer, IM As Integer, IP As Integer
Const NL = 100
Dim EI(NL) As Double, AI(NL) As Double, BI(NL) As Double, FI(NL) As Double
Dim FIN As Double, ERR As Double

       IP = 1
       AI(IP) = A
       BI(IP) = b
       FI(IP) = KRNRDD(AI(IP), BI(IP), EI(IP), b2p, b3p, RHO21, RHO31, rho)
       IM = 1
10     IM = IM + 1
       BI(IM) = BI(IP)
       AI(IM) = (AI(IP) + BI(IP)) / 2
       BI(IP) = AI(IM)
       FIN = FI(IP)
       FI(IP) = KRNRDD(AI(IP), BI(IP), EI(IP), b2p, b3p, RHO21, RHO31, rho)
       FI(IM) = KRNRDD(AI(IM), BI(IM), EI(IM), b2p, b3p, RHO21, RHO31, rho)
       ERR = Abs(FIN - FI(IP) - FI(IM)) / 2
       EI(IP) = EI(IP) + ERR
       EI(IM) = EI(IM) + ERR
       IP = 1
       ERR = 0
       FIN = 0
       For i = 1 To IM
          If EI(i) > EI(IP) Then
             IP = i
          End If

          FIN = FIN + FI(i)
          ERR = ERR + EI(i)
       Next i
       If ERR > TOL And IM < NL Then
         GoTo 10
       End If

       ADONED = FIN

End Function

Function KRNRDD(A As Double, b As Double, ABSERR As Double, _
            b2p As Double, b3p As Double, RHO21 As Double, _
       RHO31 As Double, rho As Double) As Double
'
'       Kronrod Rule
'

Dim ABSCIS As Double, CENTER As Double, FC As Double, _
  FUNSUM As Double, HFLGTH As Double
Dim RESLTG As Double, RESLTK As Double
'
'              The abscissae and weights are given for the interval (-1,1)
'              because of symmetry only the positive abscisae and their
'              corresponding weights are given.
'
```

```
'          XGK   - abscissae of the 2N+1-point Kronrod rule:
'                  XGK(2), XGK(4), ...  N-point Gauss rule abscissae;
'                  XGK(1), XGK(3), ...  abscissae optimally added
'                  to the N-point Gauss rule.
'
'          WGK   - weights of the 2N+1-point Kronrod rule.
'
'          WG    - weights of the N-point Gauss rule.
'
Dim J As Integer
Const N = 11
'
Dim WG(0 To (N + 1) / 2) As Double
Dim WGK(0 To N) As Double, XGK(0 To N) As Double

      WG(1) = 0.0556685671161745
      WG(2) = 0.125580369464905
      WG(3) = 0.186290210927735
      WG(4) = 0.233193764591991
      WG(5) = 0.262804544510248
      WG(0) = 0.272925086777901
'
      XGK(1) = 0.996369613889543
      XGK(2) = 0.978228658146057
      XGK(3) = 0.941677108578068
      XGK(4) = 0.887062599768095
      XGK(5) = 0.816057456656221
      XGK(6) = 0.730152005574049
      XGK(7) = 0.630599520161965
      XGK(8) = 0.519096129206812
      XGK(9) = 0.397944140952378
      XGK(10) = 0.269543155952345
      XGK(11) = 0.136113000799362
      XGK(0) = 0#
'
      WGK(1) = 0.00976544104596129
      WGK(2) = 0.0271565546821044
      WGK(3) = 0.0458293785644267
      WGK(4) = 0.0630974247503748
      WGK(5) = 0.0786645719322276
      WGK(6) = 0.0929530985969007
      WGK(7) = 0.105872074481389
      WGK(8) = 0.116739502461047
      WGK(9) = 0.125158799100319
      WGK(10) = 0.131280684229806
      WGK(11) = 0.135193572799885
      WGK(0) = 0.136577794711118
'
'
'          List of major variables
'
'          CENTER  - mid point of the interval
'          HFLGTH  - half-length of the interval
'          ABSCIS  - abscissae
'          RESLTG  - result of the N-point Gauss formula
'          RESLTK  - result of the 2N+1-point Kronrod formula
'
'
     HFLGTH = (b - A) / 2
```

```
      CENTER = (b + A) / 2
'
'           compute the 2N+1-point Kronrod approximation to
'           the integral, and estimate the absolute error.
'
      FC = TRVFND(CENTER, b2p, b3p, RHO21, RHO31, rho)
      RESLTG = FC * WG(0)
      RESLTK = FC * WGK(0)
      For J = 1 To N
         ABSCIS = HFLGTH * XGK(J)
         FUNSUM = TRVFND(CENTER - ABSCIS, b2p, b3p, RHO21, RHO31, rho) + _
                  TRVFND(CENTER + ABSCIS, b2p, b3p, RHO21, RHO31, rho)
         RESLTK = RESLTK + WGK(J) * FUNSUM
         If J Mod 2 = 0 Then
            RESLTG = RESLTG + WG(J / 2) * FUNSUM
         End If
      Next J
      KRNRDD = RESLTK * HFLGTH
      ABSERR = 3 * Abs((RESLTK - RESLTG) * HFLGTH)

End Function

Function TRVFND(T As Double, _
          B2 As Double, B3 As Double, RHO21 As Double, _
      RHO31 As Double, rho As Double)

      TRVFND = Exp(-T * T / 2) _
        * CBND(-T * RHO21 + B2, -T * RHO31 + B3, rho)

End Function
```

CHAPTER

14

SOME USEFUL FORMULAS

14.1 INTERPOLATION

14.1.1 Linear Interpolation

Linear interpolation consists of constructing a straight line *between* two data points. (Draw the line further, and you get an extrapolation.)

$$r_i = (r_2 - r_1)\frac{T_i - T_1}{T_2 - T_1} + r_1 \quad (14.1)$$

Example
Suppose we have a three-year interest rate of 6.3% and a four-year rate of 7.2%. What is the linear interpolated 3.5 year rate? $r_1 = 0.063$, $r_2 = 0.072$, $T_1 = 3$, $T_2 = 4$, and $T_i = 3.5$.

$$r_i = (0.072 - 0.063)\frac{3.5 - 3}{4 - 3} + 0.063 = 6.7500\%$$

14.1.2 Log-Linear Interpolation

$$r_i = \left(\frac{r_2}{r_1}\right)^{\left(\frac{T_i - T_1}{T_2 - T_1}\right)} r_1 \quad (14.2)$$

Example
What is the log-linear interpolated 3.5-year rate with the same parameters as in the linear interpolation example?

$$r_i = \left(\frac{0.072}{0.063}\right)^{\left(\frac{3.5-3}{4-3}\right)} 0.063 = 6.7350\%$$

14.1.3 Exponential Interpolation

Some trading systems use exponential interpolation when interpolating directly from discount factors.

$$D_i = D_1^{\frac{T_i}{T_1} \times \frac{T_2 - T_i}{T_2 - T_1}} \times D_2^{\frac{T_i}{T_2} \times \frac{T_i - T_1}{T_2 - T_1}} \quad (14.3)$$

Example
Suppose we have a three-year zero-coupon rate of 6.3% and a four-year zero-coupon rate of 7.2%. What is the exponential interpolated 3.5-year discount factor? $r_1 = 0.063$, $r_2 = 0.072$, $T_1 = 3$, $T_2 = 4$, $T_i = 3.5$. First we have to calculate the discount factors. The three-year discount factor is $D_1 = \frac{1}{(1+0.063)^3} = 0.8325$, and the four-year discount factor is $D_2 = \frac{1}{(1+0.072)^4} = 0.7572$. Now we can interpolate the 3.5 year discount factor:

$$D_i = 0.8325^{\frac{3.5}{3} \times \frac{4-3.5}{4-3}} \times 0.7572^{\frac{3.5}{4} \times \frac{3.5-3}{4-3}} = 0.7957$$

14.1.4 Cubic Interpolation: Lagrange's Formula

Cubic interpolation fits a third-order curve to four data points in a row, with the interpolation being between the center two points, see Lagrange (1795). The cubic interpolation on Lagrangian form is given by

$$\begin{aligned} r_i = {} & \frac{(T_i - T_2)(T_i - T_3)(T_i - T_4)}{(T_1 - T_2)(T_1 - T_3)(T_1 - T_4)} r_1 + \frac{(T_i - T_1)(T_i - T_3)(T_i - T_4)}{(T_2 - T_1)(T_2 - T_3)(T_2 - T_4)} r_2 \\ & + \frac{(T_i - T_1)(T_i - T_2)(T_i - T_4)}{(T_3 - T_1)(T_3 - T_2)(T_3 - T_4)} r_3 \\ & + \frac{(T_i - T_1)(T_i - T_2)(T_i - T_3)}{(T_4 - T_1)(T_4 - T_2)(T_4 - T_3)} r_4 \end{aligned} \tag{14.4}$$

Example
Suppose we have a two-year interest rate of 6.4%, a three-year rate of 6.3%, a four-year rate of 7.2%, and a five-year rate of 8.0%. What is the cubic interpolated 3.5-year rate? $r_1 = 0.064$, $r_2 = 0.063$, $r_3 = 0.072$, $r_4 = 0.08$, $T_1 = 2$, $T_2 = 3$, $T_3 = 4$, $T_4 = 5$, and $T_i = 3.5$.

$$\begin{aligned} r_i = {} & \frac{(3.5-3)(3.5-4)(3.5-5)}{(2-3)(2-4)(2-5)} 0.064 + \frac{(3.5-2)(3.5-4)(3.5-5)}{(3-2)(3-4)(3-5)} 0.063 \\ & + \frac{(3.5-2)(3.5-3)(3.5-5)}{(4-2)(4-3)(4-5)} 0.072 \\ & + \frac{(3.5-2)(3.5-3)(3.5-4)}{(5-2)(5-3)(5-4)} 0.08 = 6.6938\% \end{aligned}$$

14.1.5 Cubic-Spline Interpolation

A more sophisticated (but not necessarily better) "interpolation technique" is the cubic-spline method. The cubic-spline method uses all the available points to get a smooth function that fits all the input points. As this book mainly is about derivatives pricing, I will not go into detail about cubic-spline interpolation but simply present a small VBA algorithm that can be used for cubic-spline interpolation.

Computer algorithm

The following function can be used for cubic-spline interpolation. *XArray* is the array that will typically be the time to maturities (for example, in number of years) for the variables you want to interpolate. *YArray* is the values you want to interpolate from (for example, the observed part of the yield curve). The *XArray* and the *YArray* must have the same size. *X* is the time to maturity you want to find the cubic-spline value; this can be a single point or a whole array of values. The best way to get started with this code is to take a look at the ready-made Excel spreadsheet.

```
Function CubicSpline(XArray As Variant, YArray As Variant, X As Variant)

    nRates = Application.Count(XArray) - 1

    Dim M() As Variant, N() As Variant
    Dim Alfa() As Variant, Beta() As Variant, Delta() As Variant
    Dim Q() As Variant
    Dim A() As Variant, B() As Variant, C() As Variant

    ReDim M(0 To nRates + 1)
    ReDim N(0 To nRates + 1)
    ReDim Alfa(0 To nRates + 1)
    ReDim Beta(0 To nRates + 1)
    ReDim Delta(0 To nRates + 1)
    ReDim Q(0 To nRates + 1)
    ReDim A(0 To nRates + 1)
    ReDim B(0 To nRates + 1)
    ReDim C(0 To nRates + 1)

    For i = 0 To nRates - 1
        M(i) = XArray(i + 2) - XArray(i + 1)
        N(i) = YArray(i + 2) - YArray(i + 1)
    Next

    For i = 1 To nRates - 1
        Q(i) = 3 * (N(i) / M(i) - N(i - 1) / M(i - 1))
    Next

        Alfa(0) = 1
        Beta(0) = 0
        Delta(0) = 0

    For i = 1 To nRates - 1
        Alfa(i) = 2 * (M(i - 1) + M(i)) - M(i - 1) * Beta(i - 1)
        Beta(i) = M(i) / Alfa(i)
        Delta(i) = (Q(i) - M(i - 1) * Delta(i - 1)) / Alfa(i)
    Next

    Alfa(nRates) = 0
    B(nRates) = 0
    Delta(nRates) = 0

    For j = (nRates - 1) To 0 Step -1
        B(j) = Delta(j) - Beta(j) * B(j + 1)
        A(j) = N(j) / M(j) - M(j) / 3 * (B(j + 1) + 2 * B(j))
        C(j) = (B(j + 1) - B(j)) / (3 * M(j))
```

```
Next

nn = Application.Count(X)

Dim z() As Double
ReDim z(0 To nn + 1)

For i = 1 To nn
    ArrayNo = Application.Match(X(i), XArray)
    ti = Application.Index(XArray, ArrayNo)
    y1 = Application.Index(YArray, ArrayNo)
    ai = Application.Index(A(), ArrayNo)
    bi = Application.Index(B(), ArrayNo)
    ci = Application.Index(C(), ArrayNo)
    z(i - 1) = y1 + ai * (X(i) - ti) _
    + bi * (X(i) - ti)^2 + ci * (X(i) - ti)^3
Next

CubicSpline = Application.Transpose(z())

End Function
```

14.1.6 Two-Dimensional Interpolation

The formula below is a straight-line interpolation between four data points in the plane.

$$v_{t_i,T_i} = \frac{(t_2 - t_i)(T_2 - T_i)v_{t_1,T_1} + (t_2 - t_i)(T_i - T_1)v_{t_1,T_2}}{(t_2 - t_1)(T_2 - T_1)} + \frac{(t_i - t_1)(T_2 - T_i)v_{t_2,T_1} + (t_i - t_1)(T_i - T_1)v_{t_2,T_2}}{(t_2 - t_1)(T_2 - T_1)} \tag{14.5}$$

Example

Consider four bond options: Option 1 with three months to expiration, implied volatility 6%, on an underlying bond with three years to maturity; option 2 with three months to expiration, implied volatility 8% on an underlying bond with five years to maturity; option 3 with nine months to expiration, implied volatility 5% on an underlying bond with three years to maturity; and option 4 with nine months to expiration, implied volatility 7% on an underlying bond with five years to maturity. The two-dimensional interpolated volatility of an option with six months to expiration on a bond with four years to maturity is $t_1 = 0.25$, $t_2 = 0.75$, $T_1 = 3$, $T_2 = 5$, $v_{t_1,T_1} = 0.06^2$, $v_{t_1,T_2} = 0.08^2$, $v_{t_2,T_1} = 0.05^2$, $v_{t_2,T_2} = 0.07^2$, $t_i = 0.5$, and $T_i = 4$.

$$v_{0.5,4} = \frac{(0.75 - 0.5)(5 - 4)0.06^2 + (0.75 - 0.5)(4 - 3)0.08^2}{(0.75 - 0.25)(5 - 3)} + \frac{(0.5 - 0.25)(5 - 4)0.05^2 + (0.5 - 0.25)(4 - 3)0.07^2}{(0.75 - 0.25)(5 - 3)} = 0.0044$$

$$\sigma_{0.5,4} = \sqrt{0.0044} = 0.0660 = 6.60\%$$

14.2 INTEREST RATES

14.2.1 Future Value of Annuity

$$FV = C + C(1+r) + C(1+r)^2 + \cdots + C(1+r)^{n-1} = C\left[\frac{(1+r)^n - 1}{r}\right], \tag{14.6}$$

where C is the cash flow and n is the number of cash flows.

14.2.2 Net Present Value of Annuity

$$NPV = \frac{C}{(1+r)} + \frac{C}{(1+r)^2} + \cdots + \frac{C}{(1+r)^n} = C\left\{\frac{1-\left[\frac{1}{(1+r)^n}\right]}{r}\right\}, \tag{14.7}$$

where C is the cash flow and n is the number of cash flows. This formula plays an important role in many types of swaps and swaptions.

14.2.3 Continuous Compounding

From compounding m times per year to continuous compounding:

$$r_c = m \ln\left(1 + \frac{r_m}{m}\right) \tag{14.8}$$

From continuous compounding to compounding m times per year:

$$r_m = m(e^{r_c/m} - 1) \tag{14.9}$$

Example
Consider an interest rate that is quoted 8% per year with quarterly compounding. The equivalent rate with continuous compounding is

$$r_c = 4 \ln\left(1 + \frac{0.08}{4}\right) = 0.0792 = 7.92\%$$

Next, consider an interest rate that is quoted 12% per year with continuous compounding. The equivalent rate with annual compounding is

$$r_1 = 1(e^{0.12/1} - 1) = 0.1275 = 12.75\%$$

14.2.4 Compounding Frequency

From Compounding *m* Times Per Year to Annual Compounding

$$r = \left(1 + \frac{r_m}{m}\right)^m - 1 \tag{14.10}$$

From Annual Compounding to Compounding *m* Times per Year

$$r_m = \left[(1+r)^{(1/m)} - 1\right] m \qquad (14.11)$$

Example
Consider an interest rate that is quoted 8% per year with quarterly compounding. The equivalent rate with annual compounding is

$$r = \left(1 + \frac{0.08}{4}\right)^4 - 1 = 0.0824 = 8.24\% \qquad (14.12)$$

From *m* to *n* Compoundings per Year
The formula below can be used to transform a rate r_n with n compoundings per year to a rate r_m with m compoundings per year.

$$r_n = \left[\left(1 + \frac{r_m}{m}\right)^{m/n} - 1\right] n \qquad (14.13)$$

Example
Consider a rate with compounding frequency four times per year. If the rate is 7%, what is the equivalent rate with semiannual compounding?

$$r_2 = \left[\left(1 + \frac{0.07}{4}\right)^{4/2} - 1\right] 2 = 0.0706$$

The equivalent rate with semiannual compounding is 7.06%.

14.2.5 Zero-Coupon Rates from Par Bonds/Par Swaps

To calculate zero-coupon rates from par bonds or swaps, the bootstrapping method originally introduced by Caks (1977) is often used.

$$100 = C_1 e^{-r_1 T_1} + C_2 e^{-r_2 T_2} \cdots + (100 + C) e^{-r_n T_n}$$

$$100 - C \sum_{i=1}^{n-1} e^{-r_i T_i} = (100 + C) e^{-r_n T_n}$$

$$r_n = -\ln\left(\frac{100 - C \sum_{i=1}^{n-1} e^{-r_i T_i}}{100 + C}\right) / T_n, \qquad (14.14)$$

where C is the bond coupon and r_i is the continuous compounding zero-coupon rate with time to maturity T_i.

Example
What are the zero-coupon rates given the following five par coupon bonds?

1. A bond with one year to maturity and an annual coupon of 6.0%
2. A bond with two years to maturity and an annual coupon of 7.0%
3. A bond with three years to maturity and an annual coupon of 7.5%
4. A bond with four years to maturity and an annual coupon of 8.0%
5. A bond with five years to maturity and an annual coupon of 8.5%

The continuous compounding zero rate at year 1 is $\ln(1 + 0.06) = 0.0583$. The continuous compounding zero-coupon rates at years 2, 3, 4, and 5 are

$$r_2 = -\ln\left(\frac{100 - 7.0\sum_{i=1}^{2-1} e^{-r_i T_i}}{100 + 7.0}\right)/2 = 0.0680$$

$$r_3 = -\ln\left(\frac{100 - 7.5\sum_{i=1}^{3-1} e^{-r_i T_i}}{100 + 7.5}\right)/3 = 0.0729$$

$$r_4 = -\ln\left(\frac{100 - 8.0\sum_{i=1}^{4-1} e^{-r_i T_i}}{100 + 8.0}\right)/4 = 0.0780$$

$$r_5 = -\ln\left(\frac{100 - 8.5\sum_{i=1}^{5-1} e^{-r_i T_i}}{100 + 8.5}\right)/5 = 0.0834$$

14.3 RISK-REWARD MEASURES

Below are quick descriptions of some risk-reward measures often described in the literature and promoted by hedge fund managers marketing their strategies. For more detailed discussion on these, see Lhabitant (2004).

14.3.1 Treynor's Measure

Treynor (1965) introduced the following risk-reward ratio:

$$\text{Treynor ratio} = \frac{r_p - r}{\beta_p}, \tag{14.15}$$

where r_p is the average return of the portfolio during a specific time period, r is the risk-free return over the same period, and β_p is the beta of the portfolio.

14.3.2 Sharpe Ratio

One of the most used (and abused) risk-reward measures, frequently used by investors and portfolio managers, is the Sharpe (1966) ratio:

$$\text{Sharpe ratio} = \frac{r_p - r}{\sigma_p}, \tag{14.16}$$

where r_p is the average return of the portfolio during a specific time period, r is the risk-free return over the same period, and σ_p is the portfolio standard deviation (volatility of the portfolio). For option Sharpe ratios, see Chapter 2.

Example
Assume a hedge fund reports a portfolio return of 28% and that the risk-free rate over the same period is 5%. Further, the portfolio standard deviation is 27.43%. What is the Sharpe ratio? $r_p = 0.12, r = 0.05$, and $\sigma_p = 0.2743$.

$$\text{Sharpe ratio} = \frac{0.28 - 0.05}{0.2743} = 0.8385$$

14.3.3 Confidence Ratio

The confidence ratio is similar to the Sharpe ratio, but it takes into account the number of sampling points used to calculate the ratio. For example a hedge fund that has used only 12 monthly sample points to calculate its Sharpe ratio is more or less useless from a statistical standpoint because of the large confidence interval in the standard deviation.

$$\text{Confidence ratio}_{n,\alpha} = \left[\frac{r_p - r}{\sigma_U}, \frac{r_p - r}{\sigma_L}\right], \tag{14.17}$$

where σ_U is the upper confidence interval of the portfolio standard deviation and σ_L is the lower confidence interval based on n observations and a confidence interval α decided by the user. Here we have simply assumed normal distributed returns. In practice, the returns can naturally follow other types of stochastic processes that can have significant effect on the distribution of the Sharpe ratio, see Lo (2002) and Lhabitant (2004).

Example
Assume the same example as under the Sharpe ratio but with the additional information that the ratio was calculated based on only 20

observations. What is the 95% confidence ratio? $r_p = 0.12$, $r = 0.05$, $\sigma_p = 0.2743$, $n = 20$, and $\alpha = 0.05$. As we already know, the Sharpe ratio is 0.8385. From Chapter 12 we actually calculated the confidence interval from exactly this volatility estimate using 20 observations, and it was $\sigma_L = 0.2086$ and $\sigma_U = 0.4006$. This gives us a confidence ratio of

$$\text{Confidence ratio}_{n=20,\alpha=0.05} = \left[\frac{0.28-0.05}{0.4006}, \frac{0.28-0.05}{0.2086}\right]$$
$$= [0.5741, 1.1026]$$

As we can see the 95% confidence interval of the Sharpe ratio is 0.5741 to 1.1026, this is a very simple extension of reporting the Sharpe ratio that gives much more information and will be positive for funds with a long and stable track record. By reporting a ratio in this way, investors can much more easily initially screen a lot of funds, as it also says something about the length of the track record.

14.3.4 Sortino Ratio

The Sortino ratio introduced by Sortino and Price (1994) is a variation on the Sharpe ratio that differentiates harmful volatility (typically downside volatility) from volatility in general. In other words, the Sortino ratio does not penalize a fund or portfolio for its upside volatility.

$$\text{Sortino ratio} = \frac{r_p - r}{\sigma_{Down}}, \tag{14.18}$$

where σ_{Down} is the downside volatility, or in the academic literature, better known as known as the semistandard deviation.

14.3.5 Burke Ratio

Burke (1994) introduces a risk-return ratio where the returns are divided by the square root of the sum of the squared drawdowns:

$$\text{Burke ratio} = \frac{r_p - r}{\sqrt{\sum_{i=1}^{N} D_i^2}}, \tag{14.19}$$

where D_i are the drawdowns, for example, daily or weekly drawdowns.

14.3.6 Return on VaR

Another risk-return measure often used by hedge funds are simply portfolio returns R_p divided by the value at risk:

$$\text{Return on } VaR = \frac{r_p}{VaR}, \tag{14.20}$$

where VaR naturally is the value at risk, typically calculated at 95% or 99% confidence.

14.3.7 Jensen's Measure

Jensen (1965) introduced the following risk-reward measure:

$$\text{Jensen measure} = r_p - [r + \beta_p(r_M - r)], \tag{14.21}$$

where r_p is the expected total portfolio return, r the risk-free rate, β_p the beta of the portfolio, and r_m the expected market return. This measure reports the difference between the portfolio's average return and its expected return. In other words, this is a risk-adjusted performance measure that represents the average return on a portfolio over and above that predicted by the CAPM, given the portfolio's beta and the average market return. This is the portfolio's alpha (Jensen alpha).

14.4 APPENDIX C: BASIC USEFUL INFORMATION

Greek Alphabet

α	A	alpha	ι	I	iota	ρ	P	rho
β	B	beta	κ	K	kappa	σ	$\sum$	sigma
γ	Γ	gamma	λ	Λ	lambda	τ	T	tau
δ	Δ	delta	μ	M	mu	υ	Y	upsilon
ϵ	E	epsilon	ν	N	nu	ϕ	Φ	phi
ζ	Z	zeta	ξ	Ξ	xi	χ	X	chi
η	H	eta	o	O	omicron	ψ	Ψ	psi
θ	Θ	theta	π	Π	pi	ω	Ω	omega

The Natural Logarithm

$\ln(xy)$	$=$	$\ln(x) + \ln(y)$	$e^{\ln(x)}$	$=$	x
$\ln\left(\frac{x}{y}\right)$	$=$	$\ln(x) - \ln(y)$	$e^x e^y$	$=$	e^{x+y}
$\ln(x^b)$	$=$	$b\ln(x)$	$(e^x)^y$	$=$	e^{xy}
$\ln(e^x)$	$=$	x	$\frac{e^x}{e^y}$	$=$	e^{x-y}

Some Differentiation Rules

$f(x)$	$f'(x)$	$f(x)$	$f'(x)$	$f(x)$	$f'(x)$
x^a	ax^{a-1}	e^x	e^x	$\ln(u(x))$	$\frac{u'(x)}{u(x)}$
$\frac{1}{x^a}$	$-\frac{a}{x^{a+1}}$	$e^{u(x)}$	$e^{u(x)}u'(x)$		
$\sqrt{x}$	$\frac{1}{2\sqrt{x}}$	$\ln(x)$	$\frac{1}{x}$		

THE OPTION PRICING SOFTWARE

HARDWARE REQUIREMENTS

The spreadsheets require a Windows compatible computer or a Macintosh that run Microsoft Excel 2003 (or later versions).

COMPUTER CODE AND READY TO USE SPREADSHEETS

This book contains a large collection of option pricing formulas. To ease the *use* of the formulas, the book includes a CD with ready to use Excel spreadsheets with Visual Basic for Application source code for most of the formulas presented in the book. These ready-to-use spreadsheets can be used by anyone—regardless of his or her knowledge of computer languages. To start pricing options, only a minimum knowledge of using a spreadsheet is required. All that is required is typing in the input variables for the relevant formula. The computer will do the rest.

The various spreadsheets call small computer programs that are written in Visual Basic for Application (also available in Excel).

For programmers with experience in other programming languages, it is worth spending a few words on syntax particular to Visual Basic.

Declaration of variables

Visual Basic does not require the variables to be declared before they are used (C++, for instance, requires that all variables are initially declared). However, it is a good idea to declare all variables for maximum speed and to reduce possible programming errors. Most variables are thus declared in the computer code included on the CD that comes with this book.

Power
To take the power of a number, Visual Basic uses the symbol "^". Example: 4^2 corresponds to 4^2. Further, Visual Basic uses "**Exp()**" for the exponential function. Example: S * **Exp(-r * T)** corresponds to Se^{-rT}.

Line break
To instruct the computer that the command continues on the next line, Visual Basic uses a space followed by the character "_". In C++, for instance, the code for a command continues until it's terminated by ";".

Square root
Visual Basic applies the **Sqr**() command. Example: **Sqr**(T) is equal to $\sqrt{T}$.

Natural logarithm
Visual Basic uses the **Log**() command. Example: **Log**(S/X) is equal to $\ln(S/X)$.

For getting up to speed in Visual Basic for Applications programming most introduction books on the topic will do.

BIBLIOGRAPHY

AASE, K. K. (1988): "Contingent Claims Valuation When the Security Price Is a Combination of an Ito Process and a Random Point Process," *Stochastic Processes and Their Applications*, 28, 185–220.

—— (2004): "Negative Volatility and the Survival of the Western Financial Markets," *Wilmott Magazine*.

—— (2005): "The Perpetual American Put Option For Jump-Diffusions With Applications," Working Paper, UCLA and Norwegian School of Economics and Business Administration.

ABROMOWITZ, M., AND I. A. STEGUN (1974): *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*. Dover.

ACAR, E., AND R. TOFFEL (1999): "Highs and Lows: Times of the Day in the Currency CME Market," Chap. 5 in *Financial Markets Tick by Tick*, ed. Pierre Lequeux (New York: John Wiley & Sons).

ADAMCHUK, A. (1998): "From Supernova to Discovery of Supersymmetry in Finance," *New Vistas in Mathematical Foundations of Finance*, unpublished, University of Chicago.

AGCA, S., AND D. M. CHANCE (2003): "Speed and Accuracy Comparison of Bivariate Normal Distribution Approximations for Option Pricing," *Journal of Computational Finance*, 6(4), 61–96.

ALEXANDER, C. (2001): *Market Models* (New York: John Wiley & Sons).

ALLEN, S. (2003): *Financial Risk Management* (New York: John Wiley & Sons).

AMIN, K. I. (1993): "Jump Diffusion Option Valuation in Discrete Time," *Journal of Finance*, 48, 1833–1864.

AMIN, K. I., AND R. A. JARROW (1992): "Pricing Options on Risky Assets in a Stochastic Interest Rate Economy," *Mathematical Finance*, 2, 217–237.

AMIN, K. I., V. NG, AND S. C. PIRRONG (1995): "Valuing Energy Derivatives," *Managing Energy Price Risk*, Risk Publications and Enron.

ASAY, M. R. (1982): "A Note on the Design of Commodity Option Contracts," *Journal of Futures Markets*, 52, 1–7.

BACHELIER, L. (1900): *Theory of speculation* in *The Random Character of Stock Market Prices* (Cambridge, MA: MIT Press, 1964).

BACKUS, D., S. FORESI, AND L. WU (1997): "Accounting for Biases in Black–Scholes," Working paper, Stern School of Business.

BALL, C. A., AND W. N. TOROUS (1983): "A Simplified Jump Process for Common Stock Returns," *Journal of Financial and Quantitative Analysis*, 18(1), 53–66.

—— (1985): "On Jumps in Common Stock Prices and Their Impact on Call Option Pricing," *Journal of Finance*, 40, 155–173.

BARDHAN, I., A. BERGIER, E. DERMAN, C. DOSEMBET, AND I. KAN (1994): "Valuing Convertible Bonds as Derivatives," Quantitative strategies research notes.

BARLE, S., AND N. CAKICI (1995): "Growing a Smiling Tree," *Risk Magazine*, 8(10).

BARONE-ADESI, G., AND R. E. WHALEY (1987): "Efficient Analytic Approximation of American Option Values," *Journal of Finance*, 42(2), 301–320.

BATES, D. S. (1991): "The Crash of '87: Was It Expected? The Evidence from Options Markets," *Journal of Finance*, 46(3), 1009–1044.

BECKERS, S. (1983): "Variances of Security Price Returns Based on High, Low, and Closing Prices," *Journal of Business*, 56, 96–109.

BENEDER, R., AND T. VORST (2001): "Options on Dividend Paying Stocks," *Proceedings of the International Conference on Mathematical Finance* (Singapore: World Scientific Publishing Company).

BENSOUSSAN, A., M. CROUHY, AND D. GALAI (1995): "Black-Scholes Approximation of Warrant Prices," *Advances in Futures and Options Research*, 8, 1–14.

—— (1997): "Black-Scholes Approximation of Complex Option Values: The Cases of European Compound Call Options and Equity Warrants," in *Option Embedded Bonds*, ed. I. Nelken (Chicago: Irwin).

BERDHAN, I. (1995): "Exchange Rate Shocks, Currency Options and the Siegel Paradox," *Journal of International Money and Finance*, 14(3), 441–458.

BERMIN, H. P. (1996a): "Combining Lookback Options and Barrier Options: The Case of Look-Barrier Options," Working paper, Department of Economics, Lund University Sweden.

—— (1996b): "Exotic Lookback Options: The Case of Extreme Spread Options," Working paper, Department of Economics, Lund University Sweden.

—— (1996c): "Time and Path Dependent Options: The Case of Time Dependent Inside and Outside Barrier Options," Paper presented at the Third Nordic Symposium on Contigent Claims Analysis in Finance, Iceland, May.

BHAGAVATULA, R. S., AND P. CARR (1995): "Valuing Double Barrier Options with Time-Dependent Parameters," Discussion paper, Cornell University: Johnson Graduate School of Management.

BJERKSUND, P., AND G. STENSLAND (1993a): "American Exchange Options and a Put-Call Tranformation: A Note," *Journal of Business Finance and Accounting*, 20(5), 761–764.

—— (1993b): "Closed-Form Approximation of American Options," *Scandinavian Journal of Management*, 9, 87–99.

—— (1994): "An American Call on the Difference of Two Assets," *International Review of Economics and Finance*, 3(1), 1–26.

—— (1996): "Implementation of the Black-Derman-Toy Interest Rate Model," *Journal of Fixed Income*, 6, 67–75.

—— (2002): "Closed-Form Valuation of American Options," Working paper NHH.

BLACK, F. (1975): "Fact and Fantasy In the Use of Options," *Financial Analysts Journal*, 36–72.

—— (1976): "The Pricing of Commodity Contracts," *Journal of Financial Economics*, 3, 167–179.

BLACK, F., E. DERMAN, AND W. TOY (1990): "A One-Factor Model of Interest Rates and Its Application to Treasury Bond Options," *Financial Analysts Journal*, 33–39.

BLACK, F., AND P. KARASINSKI (1991): "Bond and Option Pricing when Short Rates are Lognormal," *Financial Analysts Journal*, 52–59.

BLACK, F., AND M. SCHOLES (1973): "The Pricing of Options and Corporate Liabilities," *Journal of Political Economy*, 81, 637–654.

BLAU, G. (1994–1945): "Some Aspects of The Theory of Futures Trading," *The Review of Economic Studies*, 12(1).

BONESS, A. (1964): "Elements of a Theory of Stock-Option Value," *Journal of Political Economy*, 72, 163–175.

BOS, M., AND S. VANDERMARK (2002): "Finessing Fixed Dividends," *Risk Magazine*, 157–158.

BOS, R., A. GAIRAT, AND A. SHEPELEVA (2003): "Dealing with Discrete Dividends," *Risk Magazine*, 109–112.

BOUAZIZ, L., E. BRIYS, AND M. GROUHY (1994): "The Pricing of Forward Starting Asian Options," *Journal of Banking and Finance*, 18, 823–839.

BOYLE, P., M. BROADIE, AND P. GLASSERMAN (1997): "Monte Carlo Methods for Security Pricing," *Journal of Economics Dynamics and Control*, 21, 1257–1321.

BOYLE, P. P. (1977): "Options: A Monte Carlo Approch," *Journal of Financial Economics*, 4, 323–338.

—— (1986): "Option Valuation Using a Three Jump Process," *International Options Journal*, 3, 7–12.

—— (1988): "A Lattice Framework for Option Pricing with Two State Variables," *Journal of Financial and Quantitative Analysis*, 23, 1–12.

BOYLE, P. P., J. EVNINE, AND S. GIBBS (1989): "Numerical Evaluation of Multivariate Contingent Claims," *Review of Financial Studies*, 2, 241–50.

BOYLE, P. P., AND S. H. LAU (1994): "Bumping Up Against the Barrier with the Binomial Method," *Journal of Derivatives*, 1, 6–14.

BOYLE, P. P., AND Y. K. TSE (1990): "An Algorithm for Computing Values of Options on the Maximum or Minimum of Several Assets," *Journal of Financial and Quantitative Analysis*, 25, 215–27.

BREEDEN, D. T., AND R. H. LITZENBERGER (1978): "Price of State-Contingent Claims Implicit in Option Prices," *Journal of Business*, 51, 621–651.

BRENNAN, M. J., AND E. S. SCHWARTZ (1978): "Finite Difference Methods and Jump Processes Arising in the Pricing of Contingent Claims: A Synthesis," *Journal of Financial and Quantitative Analysis*, 13(3), 461–4.

BRENNER, M., AND M. G. SUBRAHMANYAM (1988): "A Simple Solution to Compute the Implied Standard Deviation," *Financial Analysts Journal*, 80–3.

—— (1994): "A Simple Approach to Option Valuation and Hedging in the Black-Scholes Model," *Financial Analysts Journal*, 25–28.

BROADIE, M., AND P. GLASSERMAN (1997): "Pricing American Style Securities Using Simulation," *Journal of Economics Dynamics and Control*, 21, 1323–1352.

BROADIE, M., P. GLASSERMAN, AND S. KOU (1995): "A Continuity Correction for Discrete Barrier Options," Working paper.

—— (1997): "A Continuity Correction for Discrete Barrier Options," *Mathematical Finance*, 325–349.

BROCKHAUS, O., M. FARKAS, A. FERRARIS, D. LONG, AND M. OVERHAUS (2000): *Equity Derivatives and Market Risk Models* (London: Risk Books).

BROCKHAUS, O., A. FERRARIS, C. GALLUS, D. LONG, R. MARTIN, AND M. OVERHAUS (1999): *Modelling and Hedging Equity Derivatives* (London: Risk Books).

BROCKHAUS, O., AND D. LONG (2000): "Volatility Swaps Made Simple," *Risk Magazine*, January.

BROTHERTON-RATCLIFFE, R., AND B. IBEN (1993): "Yield Curve Applications of Swap Products," in *Advanced Strategies in Financial Risk Management*, ed. Robert J. Schwartz and Clifford W. Smith, Jr. (New York: New York Institute of Finance).

BROWN, C., AND D. ROBINSON (2002): "Skewness and Kurtosis Implied by Option Prices: A Correction," *Journal of Financial Research*, XXV(2).

BUCHEN, P., AND M. KELLY (1996): "The Maximum Entropy Distribution of an Asset Inferred from Option Prices," *Journal of Financial and Quantitative Analysis*, 31, 143–159.

BURGHARDT, G., AND W. HOSKINS (1994): "The Convexity Bias in Eurodollar Futures," Carr Futures Research Note, September.

BURGHARDT, G., AND M. LANE (1990): "How to Tell if Options Are Cheap," *Journal of Portfolio Management*, 72–78.

BURGHARDT, G., AND L. LIU (2002): "New Convexity Bias Series," Carr Futures Research Note, February.

BURGHARDT, G., AND G. PANOS (2001): "Hedging Convexity Bias," Carr Futures Research Note, August.

BURKE, G. (1994): "A Sharper Sharpe Ratio," *The Computerized Trader*, March.

CAKS, J. (1977): "The Coupon Effect on Yield to Maturity," *Journal of Finance*, March, 103–115.

CARR, P. (1988): "The Valuation of Sequential Exchange Opportunities," *Journal of Finance*, 43, 1235–1256.

—— (1994): "European Put Call Symmetry," Working paper, Cornell University.

CARR, P., AND J. BOWIE (1994): "Static Simplicity," *Risk Magazine*, 7(8).

CARR, P., K. ELLIS, AND V. GUPTA (1998): "Static Hedging of Exotic Options," *Journal of Finance*, 53.

CARR, P., AND D. MADAN (1998): "Towards a Theory of Volatility Trading," in *Volatility* (London: Risk Books).

CHALASANI, P., S. JHA, F. EGRIBOYUN, AND A. VARIKOOTY (1999): "A Refined Binomial Lattice for Pricing American Asian Options," *Review of Derivatives Research*, 3(1), 85–105.

CHEUK, T. H. F., AND T. C. VORST (1996): "Complex Barrier Options," *Journal of Derivatives*, 4, 8–22.

CHO, H. Y., AND H. Y. LEE (1997): "A Lattice Model for Pricing Geometric and Arithmetic Average Options," *Journal of Financial Engineering*, 6, 179–191.

CHO, H. Y., AND K. W. LEE (1995): "An Extension of the Three-Jump Process Model for Contingent Claim Valuation," *Journal of Derivatives*, 3, 102–108.

CHRISS, N. A. (1996): *Black-Scholes and Beyond* (Chicago: Irwin Professional Publishing).

CLEWELOW, L., AND C. STRICKLAND (1998): *Implementing Derivatives Models* (New York: John Wiley & Sons).

CONZE, A., AND VISWANATHAN (1991): "Path Dependent Options: The Case of Lookback Options," *Journal of Finance*, 46, 1893–1907.

CORRADO, C. J., AND T. W. MILLER (1996a): "A Note on a Simple, Accurate Formula to Compute Implied Standard Deviations," *Journal of Banking and Finance*, 20, 595–603.

—— (1996b): "Volatility Without Tears," *Risk Magazine*, 9(7).

CORRADO, C. J., AND T. SU (1996): "Skeewness and Kurtosis in S&P 500 Index Returns Implied by Option Prices," *Journal of Financial Research*, XIX, 175–192.

CORTAZAR, G., AND E. S. SCHWARTZ (1994): "The Valuation of Commodity-Contigent Claims," *Journal of Derivatives*, 1, 27–39.

COX, J. (1975): "Notes on Option Pricing I: Constant Elasticity of Variance Diffusions." Working Paper, Stanford University.

COX, J. C., J. E. INGERSOLL, AND S. A. ROSS (1985): "A Theory of the Term Structure of Interest Rates," *Econometrica*, 53, 385–407.

Cox, J. C., and S. A. Ross (1976): "The Valuation of Options for Alternative Stochastic Processes," *Journal of Financial Economics*, 3, 145–166.

Cox, J. C., S. A. Ross, and M. Rubinstein (1979): "Option Pricing: A Simplified Approach," *Journal of Financial Economics*, 7, 229–263.

Cox, J. C., and M. Rubinstein (1985): *Options Markets*, Chap. 8 (Upper Saddle River, NJ: Prentice Hall).

Crack, T. F. (1997): "Derivatives Securities Pricing," Course notes, Indiana University.

—— (2004): *Heard on the Street: Quantitative Questions from Wall Street Job Interviews*, 9th ed. (Timothy Crack).

Curran, M. (1992): "Beyond Average Intelligence," *Risk Magazine*, 5(10).

—— (1993): "Greeks in Monte Carlo," *Risk Magazine*, April.

—— (1994): "Valuing Asian and Portfolio Options by Conditioning on the Geometric Mean Price," *Management Science*, 40(12), 1705–1711.

Dai, M., Y. Kwok, and L. X. Wu (2003): "Options With Multiple Reset Rights," *International Journal of Theoretical and Applied Finance*, 6(6), 637–653.

Dai, M., and Y. K. Kwok (2004): "Knock-in American Option," *The Journal of Futures Markets*, 24(2), 179–192.

Dai, T., G. Huang, and Y. Lyuu (2002): "Extremely Accurate and Efficient Algorithms for European-Style Asian Options with Range," Working paper.

Dai, T., and Y. Lyuu (2002): "Efficient, Exact Algorithms for Asian Options with Multiresolution Lattices," *Review of Derivatives Research*, 5, 181–203.

Demeterifi, K., E. Derman, M. Kamal, and J. Zou (1999): "More Than You Ever Wanted to Know about Volatility Swaps," Working paper, Goldman Sachs.

Derman, E. (1992): "Outperformance Options," Working paper, Goldman Sachs.

—— (1999): "Regimes of Volatility," *Risk Magazine*, April.

Derman, E., and N. Taleb (2005): "The Illusion of Dynamic Delta Replication," *Quantitative Finance*, 5(4), 323–326.

DERMAN, E., I. BARDHAN, D. ERGENER, AND I. KANI (1995): "Enhanced Numerical Methods for Options with Barriers," *Financial Analysts Journal*, November–December, 65–74.

DERMAN, E., D. ERGENER, AND I. KANI (1995): "Static Options Replication," *Journal of Derivatives*, 2, 78–95.

DERMAN, E., AND M. KAMAL (1999): "When You Cannot Hedge Continuously: The Corrections of Black-Scholes," *Risk Magazine*, 12, 82–85.

DERMAN, E., AND I. KANI (1994): "Riding on a Smile," *Risk Magazine*, 7(2).

—— (1998): "Stochastic Implied Trees: Arbitrage Pricing with Stochastic Term and Strike Structure of Volatility," *International Journal of Theoretical and Applied Finance*, 1(1), 61–110.

DERMAN, E., I. KANI, AND N. CHRISS (1996): "Implied Trinomial Trees of the Volatility Smile," *Journal of Derivatives*, 3(4), 7–22.

DERMAN, E., P. KARASINSKI, AND J. S. WECKER (1990): "Understanding Guaranteed Exchange-Rate Contracts in Foreign Stock Investments," *International Equity Strategies*, Goldman Sachs, June.

DEROSA, D. (2000): *Options on Foreign Exchange* (New York: John Wiley & Sons).

DRAPER, H. (1721): "$\acute{\varepsilon}\tau o\varsigma$–Volume and Two Greeks Gives $\delta\acute{\omega}\delta\varepsilon\kappa\alpha$ Numbers, $\theta\varepsilon\acute{\alpha}$," *Analen der Noisulli*, 67, 3–500.

DRAVID, A., M. RICHARDSON, AND T. S. SUN (1993): "Pricing Foreign Index Contigent Claims: An Application to Nikkei Index Warrants," *Journal of Derivatives*, 1(1), 33–51.

DREZNER, Z. (1978): "Computation of the Bivariate Normal Integral," *Mathematics of Computation*, 32, 277–279.

DREZNER, Z., AND G. O. WESOLOWSKY (1990): "On the Computation of the Bivariate Normal Integral," *The Journal of Statistical Computation and Simulation*, 35(1, 2), 101–107.

DUFFY, D. J. (2006): *Finite Difference Methods in Financial Engineering*. (New York: John Wiley & Sons).

DUMAS, B., L. P. JENNERGREN, AND B. NÄSLUND (1995): "Siegel's Paradox and The Pricing of Currency Options," *Journal of International Money and Finance*, 14(3), 213–223.

DUPIRE, B. (1994): "Pricing with a Smile," *Risk Magazine*, 7(1).

ENGLE, R. F., AND J. MEZRICH (1995): "Grappling with GARCH," *Risk Magazine*, September.

ENGLE, R. F., AND V. K. NG (1993): "Measuring and Testing the Impact of News on Volatility," *Journal of Finance*, 48(5), 1749–1779.

ESSER, A. (2003): "General Valuation Principles for Arbitrary Payoffs and Applications to Power Options Under Stochastic Volatility," Working paper, Goethe University.

EYDELAND, A., AND K. WOLYNIEC (2003): *Energy and Power Risk Management* (New York: John Wiley & Sons).

FAURE, H. (1982): "Discrépence de Suites Associées á un Systéme de Numération (en Simension s)," *Acta Arithmetica*, 41, 337–351.

FEINSTEIN, S. (1988): "A Source of Unbiased Implied Volatility Forecasts," Working paper, 88–9, Federal Reserve Bank of Atlanta.

FIGLEWSKI, S., AND B. GAO (1999): "The Adaptive Mesh Model: A New Approach to Efficient Option Pricing," *Journal of Financial Economics*, Elsevier, 53(3), September, 313–351.

FLESAKER, B. (1993): "Arbitrage Free Pricing of Interest Rate Derivatives and Forward Contracts," *The Journal of Futures Markets*, 13, 77–91.

FOUQUE, J., G. PAPANICOLAOU, AND K. R. SIRCAR (2000): *Derivatives in Financial Markets with Stochastic Volatility* (Cambridge, UK: Cambridge University Press).

FRENCH, D. W. (1984): "The Weekend Effect on the Distribution of Stock Prices," *Journal of Financial Economics*, 13, 547–559.

FRENCH, K. R. (1980): "Stock Returns and the Weekend Effect," *Journal of Financial Economics*, 8, 55–69.

FRENCH, K. R., AND R. ROLL (1986): "Stock Return Variances," *Journal of Financial Economics*, 17, 5–26.

FRISHLING, V. (2002): "A Discrete Question," *Risk Magazine*, January.

GAO, B., J. HUANG, AND M. SUBRAHMANYAM (2000): "The Valuation of American Barrier Options Using the Decomposition Technique," *Journal of Economics Dynamics & Control*, 24, 1783–1827.

GARMAN, M. (1989): "Recollection in Tranquility," *Risk Magazine*, 2(3).

—— (1992): "Charm School," *Risk Magazine*, 5(7), 53–56.

GARMAN, M., AND S. W. KOHLHAGEN (1983): "Foreign Currency Option Values," *Journal of International Money and Finance*, 2, 231–237.

GARMAN, M. B., AND M. J. KLASS (1980): "On the Estimation of Security Price Volatilities from Historical Data," *Journal of Business*, 53(1), 67–78.

GATHERAL, J. (2006): *The Volatility Surface: A Practitioner's Guide*. (New York: John Wiley & Sons).

GEMAN, H., AND A. EYDELAND (1995): "Domino Effect," *Risk Magazine*, 8(4).

GEMAN, H., AND M. YOR (1993): "Bessel Processes, Asian Options, and Perpetuities," *Mathematical Finance*, 3(4), 349–375.

—— (1996): "Pricing and Hedging Double-Barrier Options: A Probabilistic Approach," *Mathematical Finance*, 6(4), 365–378.

GENZ, A. (2004): "Numerical Computation of Rectangular Bivariate and Trivariate Normal *t* Probabilities," *Statistics and Computing*, 14, 151–160.

GERBER, H. U., AND S. W. SHIU (1994): "Martingale Approach to Pricing Perpetual American Options," *Astin Bulletin*, 24(2), 195–220.

GESKE, R. (1977): "The Valuation of Corporate Liabilities as Compound Options," *Journal of Financial and Quantitative Analysis*, 541–552.

—— (1979a): "A Note on an Analytical Formula for Unprotected American Call Options on Stocks with Known Dividends," *Journal of Financial Economics*, 7, 375–80.

—— (1979b): "The Valuation of Compound Options," *Journal of Financial Economics*, 7, 63–81.

GIBBONS, M. R., AND P. HESS (1981): "Day of the Week Effect and Asset Returns," *Journal of Business*, 54, 579–596.

GIBSON, R., AND E. S. SCHWARTZ (1990): "Stochastic Convenience Yield and the Pricing of Oil Contingent Claims," *Journal of Finance*, 45, 959–976.

GLASSERMAN, P. (2003): *Monte Carlo Methods in Financial Engineering* (New York: Springer-Verlag).

GLASSERMAN, P., P. HEIDELBERG, AND P. SHAHABUDDIN (2000): "Importance Sampling in the Heath-Jarrow-Merton Framework," Working paper, Columbia University.

GOLDMAN, B. M., H. B. SOSIN, AND M. A. GATTO (1979): "Path Dependent Options: "Buy at the Low Sell at the High"," *Journal of Finance*, 34(5), 1111–1127.

GRABBE, O. J. (1983): "The Pricing of Put and Call Options on Foreign Exchange," *Journal of International Money and Finance*, 2(3), 239–253.

GRANDITS, P., AND W. SCHACHINGER (2001): "Leland's Approach to Option Pricing: The Evolution of a Discontinuity," *Mathematical Finance*, 11(3).

GRAY, S. F., AND R. E. WHALEY (1997): "Valuing Bear Market Reset Warrants with a Periodic Reset," *Journal of Derivatives*, 5, 229–263.

—— (1999): "Reset Put Options: Valuation, Risk Characteristics, and Application," *Australian Journal of Management*, 24, 1–20.

HAGAN, P. S., D. KUMAR, A. S. LESNIEWSKI, AND D. E. WOODWARD (2002): "Managing Smile Risk," *Wilmott Magazine*, September, 1(1).

HAGAN, P. S., AND D. WOODWARD (1999): "Equivalent Black Volatilities," *Applied Mathematical Finance*, 6, 113–129.

HAKANSSON, N. (1991): "Supershares," Discussion paper, Institute of Business and Economic Research, University of California at Berkeley.

HAKANSSON, N. H. (1976): "The Purchasing Power Fund: A New Kind of Financial Intermediary," *Financial Analysts Journal*, 32, 49–59.

HALTON, J. H. (1960): "On the Efficiency of Certain Quasi-Random Sequences of Points in Evaluating Multi-Dimensional Integrals," *Numerische Mathematik*, 2, 84–94.

HANDLEY, J. C. (2001): "Variable Purchase Options," *Review of Derivatives Research*, (4), 219–230.

HANSEN, A. T., AND P. L. JØRGENSEN (1997): "Analytical Valuation of American-Style Asian Options," Working paper, University of Aarhus, Denmark.

HART, I., AND M. ROSS (1994): "Striking Continuity," *Risk Magazine*, 7(6).

HART, J. (1968): *Computer Approximations* (Algorithm 5666 for the error function). (New York: John Wiley & Sons).

HAUG, E., A. JAVAHERI, AND P. WILMOTT (2004): "GARCH and Volatility Swaps," *Quantitative Finance*, 4, 589–595.

HAUG, E. G. (1992): "Volatilitetskjegler som Analyseverktøy," *Beta*, 6(1), 27–35.

—— (1993): "Opportunities and Perils of Using Option Sensitivities," *Journal of Financial Engineering*, 2(3), 253–269.

—— (1996): "Implisitt Korrelasjon i Valutamarkedet," *Beta*, 9(1), 39–43.

—— (1998): "Put-Call Barrier Transformations," Working paper, Tempus Financial Engeneering.

—— (2001a): "Closed-Form Valuation of American Barrier Options," *International Journal of Theoretical and Applied Finance*, XIX, 175–192.

—— (2001b): "First-then-Knockout Options," *Wilmott Magazine*, August.

—— (2001c): "The Options Genius," *Wilmott Magazine*, 1(1).

—— (2002): "A Look in the Antimatter Mirror," *Wilmott Magazine*, December, www.wilmott.com.

—— (2003): "Know Your Weapon, Part 1 and 2," *Wilmott Magazine*, May and August.

—— (2004): "Why So Negative to Negative Probabilities?" *Wilmott Magazine*, September/October.

—— (2005a): "Practical Valuation of Power Derivatives" *Wilmott Magazine*, November.

—— (2005b): "Valuation of Complex Barrier Options Using Symmetry," Unpublished working paper.

—— (2006): *Derivatives: Models on Models.* (New York: John Wiley & Sons).

HAUG, E. G., AND J. HAUG (1996): "Implied Forward Volatility," Paper presented at the Third Nordic Symposium on Contingent Claims Analysis in Finance, Iceland, May.

—— (1998): "Closed Form Approximation for European Options with Discrete Dividend," Unpublished working paper.

—— (2001): "Resetting Strikes, Barriers, and Time," www.wilmott.com.

—— (2002): "Knock-in/out Margrabe," *Wilmott Magazine*, 1(2), 183–204.

HAUG, E. G., J. HAUG, AND A. LEWIS (2003): "Back to Basics A New Approach to the Discrete Dividend Problem," *Wilmott Magazine*, September.

HAUG, E. G., J. HAUG, AND W. MARGRABE (2003): "Asian Pyramid Power," *Wilmott Magazine*.

HAYKOV, J. M. (1993): "A Better Control Variate for Pricing Standard Asian Options," *Journal of Financial Engineering*, 2(3), 207–216.

HEENK, B. A., A. G. Z. KEMNA, AND A. C. F. VORST (1990): "Asian Options on Oil Spreads," *Review of Futures Markets*, 9, 510–528.

HENDERSON, V., AND R. WOJAKOWSKI (2001): "On the Equivalence of Floating and Fixed-Strike Asian Options," *Journal of Finance*, 52(3), 923–973.

HESTON, S., AND G. ZHOU (2000): "On the Rate of Convergence of Discrete-Time Contingent Claims," *Mathematical Finance*, 10, 53–75.

HEYNEN, R. C., AND H. M. KAT (1994a): "Crossing Barriers," *Risk Magazine*, 7.

—— (1994b): "Partial Barrier Options," *Journal of Financial Engineering*, 3, 253–274.

—— (1994c): "Selective Memory," *Risk Magazine*, 7(11).

—— (1996a): "Brick by Brick," *Risk Magazine*, 9(6).

—— (1996b): "Discrete Partial Barrier Options with a Moving Barrier," *Journal of Financial Engineering*, 5(3), 199–210.

—— (1996c): "Pricing and Hedging Power Options," *Financial Engineering and Japanese Markets*, 3, 253–261.

HIGGINS, L. R. (1902): *The Put-and-Call*. (London: E. Wilson).

HO, T. S. Y., AND S.-B. LEE (1986): "Term Structure Movements and Pricing Interest Rate Contingent Claims," *Journal of Finance*, 41, 1011–29.

HODGES, S. D., AND M. J. P. SELBY (1987): "On the Evaluation of Compound Options," *Management Science*, 33(3), 347–355.

HODGES, S. D., AND R. G. TOMPKINS (2002): "Volatility Cones and Their Sampling Properties," *Journal of Derivatives*, Fall.

HOGGARD, T., A. E. WHALLEY, AND P. WILMOTT (1994): "Hedging Option Portfolios in the Presence of Transaction Costs," *Advances in Futures and Options Research*, 7, 21–35.

HUI, C. H. (1996): "One-Touch Barrier Binary Option Values," *Applied Financial Economics*, 6, 343–346.

—— (1997): "Time-Dependent Barrier Option Values," *Journal of Futures Markets*, 17, 667–688.

HULL, J. (2005): *Option, Futures, and Other Derivatives*, 6th ed. (Upper Saddle River, NJ: Prentice Hall).

HULL, J., AND A. WHITE (1987): "The Pricing of Options on Assets with Stochastic Volatilities," *Journal of Finance*, XLII(2), 281–300.

—— (1988): "An Analysis of the Bias in Option Pricing Caused by a Stochastic Volatility," *Advances in Futures and Options Research*, 2.

—— (1990a): "Pricing Interest Rate Derivative Securities," *Review of Financial Studies*, 3(4), 573–92.

—— (1990b): "Valuing Derivative Securities Using the Explicit Finite Difference Method," *Journal of Financial and Quantitative Analysis*, 25(1), 87–100.

—— (1992): "In the Common Interest," *Risk Magazine*, 5(3).

—— (1993): "Efficient Procedures for Valuing European and American Path-Dependent Options," *Journal of Derivatives*, 1, 21–31.

IKEDA, M., AND N. KUINTOMO (1992): "Pricing Options with Curved Boundaries," *Mathematical Finance*, 2, 275–298.

JÄCKEL, P. (2002): *Monte Carlo Methods in Finance* (New York: John Wiley & Sons).

JACKWERTH, J. C., AND M. RUBINSTEIN (1996): "Recovering Probability Distributions from Option Prices," *Journal of Finance*, 51, 1611–1631.

JAMES, P. (2003): *Option Theory* (New York: John Wiley & Sons).

JAMSHIDIAN, F. (1989): "An Exact Bond Option Formula," *Journal of Finance*, 44, 205–9.

—— (1991): "Forward Induction and Construction of Yield Curve Diffusion Models," *Journal of Fixed Income*, pp. 62–74.

—— (1996): "Sorting Out Swaptions," *Risk Magazine*, 9(3).

JARROW, R., AND A. RUDD (1982): "Approximate Option Valuation for Arbitrary Stochastic Processes," *Journal of Financial Economics*, 10, 347–369.

—— (1983): *Option Pricing* (Chicago: Irwin).

JARROW, R., AND S. TURNBULL (1996): *Derivatives Securities*. South Western College Publishing.

JACKSON, M. and J. STAUNTON (2001): *Advanced Modelling in Finance Using Excel and VBA* (New York: John Wiley & Sons).

JAVAHERI, A. (2005): *Inside Volatility Arbitrage: The Secrets of Skewness* (New York: John Wiley & Sons).

JENNERGREN, L. P., AND B. NASLUND (1993): "A Comment on Valuation of Executive Stock Options and the FASB Proposal," *The Accounting Review*, 68(1), 179–183.

JENSEN, M. (1965): "The Performance of Mutual Funds in the Period 1945–1964," *Journal of Finance*, May, 389–415.

JOHNSON, H. (1987): "Options on the Maximum or the Minimum of Several Assets," *Journal of Financial and Quantitative Analysis*, 22(3), 277–283.

JURCZENKO, E., B. MAILLET, AND B. NEGREA (2004): "A Note on Skewness and Kurtosis Adjusted Option Pricing Models under the Martingale Restriction," *Quantitative Finance*, 4(5), 479–488.

KABANOV, Y. M., AND M. M. SAFARIAN (1997): "On Leland's Strategy of Option Pricing with Transactions Costs," *Finance and Stochastics*, 1(2).

KAT, H., AND L. VERDONK (1995): "Tree Surgery," *Risk Magazine*, 8(2).

KEMNA, A., AND A. VORST (1990): "A Pricing Method for Options Based on Average Asset Values," *Journal of Banking and Finance*, 14, 113–129.

KEYNES, J. M. (1924): *A Tract on Monetary Reform*. Reprinted 2000. (Amherst, NY: Prometheus Books).

KIRIKOS, G., AND D. NOVAK (1997): "Convexity Conundrums," *Risk Magazine*, March, 60–61.

KIRK, E. (1995): "Correlation in the Energy Markets," in *Managing Energy Price Risk*. Risk Publications and Enron.

KNIGHT, J., AND S. SATCHELL (2001): *Return Distributions in Finance*, Chap. 9 (Burlington, MA: Butterworth-Heinemann).

KUNITOMO, N. (1992): "Improving the Parkinson Method of Estimating Security Price Volatilities," *Journal of Business*, 65, 295–302.

LAGRANGE, J. L. (1795): *Leçons élémentaires sur les mathématiques*.

LEISEN, D. P. J., AND M. REIMER (1996): "Binomial Models for Option Valuation Examining and Improving Convergence," *Applied Mathematical Finance*, 3, 319–346.

LELAND, H. (1985): "Option Pricing and Replication with Transactions Costs," *Journal of Finance*, XL(5), 1283–1301.

LEVY, E. (1992): "Pricing European Average Rate Currency Options," *Journal of International Money and Finance*, 11, 474–491.

—— (1997): "Asian Options," in *Exotic Options: The State of the Art*, ed. L. Clewlow and C. Strickland (Washington, DC: International Thomson Business Press).

LEVY, E., AND S. TURNBULL (1992): "Average Intelligence," *Risk Magazine*, 5(2).

LEWIS, A. (2000): *Option Valuation under Stochastic Volatility* (Newport Beach, CA: Finance Press).

LHABITANT, F. (2004): "Hedge Funds Quantitative Insight" (New York: John Wiley & Sons).

LIAO, S.-L., AND C.-W. WANG (2003): "The Valuation of Reset Options with Multiple Strike Resets and Reset Dates," *The Journal of Futures Markets*, 23(1), 87–107.

LIEU, D. (1990): "Option Pricing with Futures-Style Margining," *Journal of Futures Markets*, 10, 327–328.

LINDSET, S., AND S.-A. PERSSON (2006): "A Note On a Barrier Exchange Option: The World's Simplest Option Formula?," *Financial Research Letters*, 3.

LINETSKY, V. (2004): "Spectral Expansions for Asian (Average Price) Options," *Operations Research*.

LO, A. (2002): "The Statistics of Sharpe Ratios," *Financial Analyst Journal*, 58, 36–52.

LO, A., AND J. WANG (1995): "Implementing Option Pricing when Asset Returns Are Predictable," *Journal of Finance*, 50, 87–129.

LONGSTAFF, F. A. (1990): "Pricing Options with Extendible Maturities: Analysis and Applications," *Journal of Finance*, 45(3), 935–957.

—— (1995): "Option Pricing and the Martingale Restriction," *Review of Financial Studies*, 8, 1091–1124.

LONGSTAFF, F. A., P. SANTA-CLARE, AND E. SCHWARTZ (2000): "The Relative Valuation of Caps and Swaptions: Theoretical and Empirical Evidence," Working paper.

MANASTER, S., AND G. KOEHLER (1982): "The Calculation of Implied Variances from the Black-Scholes Model," *Journal of Finance*, 37(1), 227–230.

MANZANO, J. (2001): "Mirror Options," *Wilmott Magazine*, October, www.wilmott.com.

MARGRABE, W. (1978): "The Value of an Option to Exchange One Asset for Another," *Journal of Finance*, 33(1), 177–186.

MARSH, T. A., AND E. R. ROSENFIELD (1986): "Non-Trading, Market Making, and Estimates of Stock Price Volatility," *Journal of Financial Economics*, 15, 359–372.

MCDONALD, R. L. (2002): *Derivatives Markets* (Upper Saddle River, NJ: Addison-Wesley).

MCKEAN, H. P. (1965): "A Free Boundary Problem For The Heat Equation Arising From A Problem in Mathematical Economics," *Industrial Management Review*, 6(2), 32–39.

MEIER, S. (2000): "Implementing the Broadie-Glasserman Approach for Pricing Multi-Asset American Options Using Monte Carlo Simulation," *Semesterarbeit*, University of Zurich.

MELLO, A. S., AND H. J. NEUHAUS (1998): "A Portfolio Approach to Risk Reduction in Discretely Rebalanced Option Hedges," *Management Science*, 44(7), 921–934.

MERTON, R. C. (1971): "Optimum Consumption and Portfolio Rules in a Continuous-Time Model," *Journal of Economic Theory*, 3, 373–413.

—— (1973): "Theory of Rational Option Pricing," *Bell Journal of Economics and Management Science*, 4, 141–183.

—— (1976): "Option Pricing When Underlying Stock Returns Are Discontinuous," *Journal of Financial Economics*, 3, 125–144.

MILTERSEN, K., K. SANDMANN, AND D. SONDERMANN (1997): "Closed Form Solutions for Term Structure Derivatives with Log-Normal Interest Rates," *Journal of Finance*, 52(1).

MILTERSEN, K., AND E. S. SCHWARTZ (1998): "Pricing of Options on Commodity Futures with Stochastic Term Structures of Convenience Yields and Interest Rates," *Journal of Financial and Quantitative Analysis*, 33(1).

MITCHELL, WESLEY, C. (1915): "The Making and Using of Index Numbers," *Introduction to Index Numbers and Wholesale Prices in the United States and Foreign Countries* (published in 1915 as Bulletin No. 173 of the U.S. Bureau of Labor Statistics, reprinted in 1921 as Bulletin No. 284, and in 1938 as Bulletin No. 656).

MORO, B. (1995): "The Full Monte," *Risk Magazine*, February.

NELKEN, I. (1993): "Square Deals," *Risk Magazine*, 6(4).

NELSON, D. B. (1990): "ARCH Models as Diffusion Approximations," *Journal of Econometrics*, 45, 7–38.

NELSON, S. A. (1904): *The A B C of Options and Arbitrage*. (New York: The Wall Street Library).

NEUBERGER, A. (1994): "The Log Contract: A New Instrument to Hedge Volatility," *Journal of Portfolio Management*, Winter, 74–80.

—— (1996): "The Log Contract and Other Power Contracts," in *The Handbook of Exotic Options*, ed. I. Nelken, Winter, 200–212.

PARKINSON, M. (1980): "The Extreme Value Method for Estimating the Variance of the Rate of Return," *Journal of Business*, 53(1), 61–65.

PEARSON, N. D. (1995): "An Efficient Approach for Pricing Spread Options," *Journal of Derivatives*, 3, 76–91.

PECHTL, A. (1995): "Classified Information," *Risk Magazine*, 8.

PESKIR, G., AND A. N. SHIRYAEV (2001): "A Note on the Put-Call Parity and a Put-Call Duality," *Theory of Probability and its Applications*, 46, 181–183.

PIETERBARG, V. V., AND M. A. RENEDO (2004): "Eurodollar Futures Convexity Adjustments in Stochastic Volatility Models," Working paper.

PILIPOVIĆ, D. (1997): *Energy Risk* (New York: McGraw-Hill).

PRESS, W. H., S. TEUKOLSKY, W. VETTERLING, AND B. P. FLANNERY (1992): *Numerical Recipies in Fortran 77: The Art of Scientific Computing*, 2nd ed. (Cambridge, UK: Cambridge University Press).

RANDALL, C., AND D. TAVELLA (2000): *Pricing Financial Instruments the Finite Difference Method* (New York: John Wiley & Sons).

RAO, C. R. (1973): "Distributions of Correlation Coefficient," in *Linear Statistical Inference and Its Applications*, 2nd ed. (New York: John Wiley & Sons), pp. 206–208.

RAY, C. (1993): *The Bond Market: Trading and Risk Management* (New York: McGraw-Hill).

REBONATO, R. (1996): *Interest-Rate Option Models* (New York: John Wiley & Sons).

—— (2004): *Volatility and Correlation: The Perfect Hedger and the Fox* (New York: John Wiley & Sons).

REIDER, R. (1993): "An Efficient Monte Carlo Techniques for Pricing Options," Working paper, Wharton, University of Pennsylvania.

REIMER, M., AND M. SANDEMANN (1995): "A Discrete Time Approach for European and American Barrier Options," Working paper.

REINER, E. (1992): "Quanto Mechanics," *Risk Magazine*, 5, 59–63.

REINER, E., AND M. RUBINSTEIN (1991a): "Breaking Down the Barriers," *Risk Magazine*, 4(8).

—— (1991b): "Unscrambling the Binary Code," *Risk Magazine*, 4(9).

REISMANN, H. (1992): "Movements of the Term Structure of Commodity Futures and Pricing of Commodity Claims," Working paper, Faculty of I. E. and Management, Technion-Israel Institute of Technology, Israel.

RENDLEMAN, R. J., AND B. J. BARTTER (1979): "Two-State Option Pricing," *Journal of Finance*, 34, 1093–1110.

—— (1980): "The Pricing of Options on Debt Securities," *Journal of Financial and Quantitative Analysis*, 15, 11–24.

RICH, D. R. (1994): "The Mathematical Foundation of Barrier Option-Pricing Theory," *Advances in Futures and Options Research*, 7, 267–311.

RICH, D. R., AND D. M. CHANCE (1993): "An Alternative Approach to the Pricing of Options on Multiple Assets," *Journal of Financial Engineering*, 2(3), 271–285.

RITCHKEN, P. (1995): "On Pricing Barrier Options," *Journal of Derivatives*, 3, 19–28.

RITCHKEN, P., AND R. TREVOR (1997): "Pricing Options under Generalized GARCH and Stochastic Volatility Processes," *CMBF Papers*, 19, www.mafc.mq.edu.au/MAFCpapers/papers.htm.

ROGERS, L. C. G., AND S. E. SATCHELL (1991): "Estimating Variance from High, Low, and Closing Prices," *The Annals of Applied Probability*, 1, 504–512.

ROLL, R. (1977): "An Analytic Valuation Formula for Unprotected American Call Options on Stocks with Known Dividends," *Journal of Financial Economics*, 5, 251–58.

RUBINSTEIN, M. (1990): "The Super Trust," Working paper, www.in-the-money.com.

—— (1991a): "Double Trouble," *Risk Magazine*, 5(1).

—— (1991b): "One for Another," *Risk Magazine*, 4(7).

—— (1991c): "Options for the Undecided," *Risk Magazine*, 4(4).

—— (1991d): "Somewhere over the Rainbow," *Risk Magazine*, 4(10).

—— (1994a): "Implied Binomial Trees," *Journal of Finance*, 49, 771–818.

—— (1994b): "Return to OZ," *Risk Magazine*, 7(11).

—— (1995a): "As Simple as One, Two, Three," *Risk Magazine*, 8(1).

—— (1995b): "SuperShares," Chap. 19 in *The Handbook of Equity Derivatives*, ed. J.C. Francis (New York: John Wiley & Sons).

RUBINSTEIN, M. (1998): "Edgeworth Binomial Trees," *Journal of Derivatives*, XIX, 20–27.

SAMUELSON, P. (1965): "Rational Theory of Warrant Pricing," *Industrial Management Review*, 6, 13–31.

SCHAEFER, S., AND E. SCHWARTZ (1987): "Time Dependent Variance and the Pricing of Bond Options," *Journal of Finance*, 42, 1113–28.

SCHNABEL, J. A., AND J. Z. WEI (1994): "Valuing Takover-Contingent Foreign Exchange Call Options," *Advances in Futures and Options Research*, 7, 223–236.

SCHWARTZ, E. S. (1997): "The Stochastic Behavior of Commodity Prices: Implications for Valuation and Hedging," Finance working paper 1-97, The John E. Anderson Graduate School of Management at UCLA, Los Angeles, *Journal of Finance*, 52(3), 923–973 (July 1997).

SHARPE, W. (1966): "Mutual Fund Performance," *Journal of Business*, 119–138.

SHAW, W. (1998): *Modelling Financial Derivatives* (Cambridge, UK: Cambridge University Press).

SHILLING, H. (2001): "No-Arbitrage Bounds and Static Hedging of Compound Options," Working paper, Commerzbank.

SHIMKO, D. (1994): "Options on Futures Spreads: Hedging, Speculation, and Valuation," *The Journal of Futures Markets*, 14(2), 183–213.

SMITH, JR., C. W. (1976): "Option Pricing: A Review," *Journal of Financial Economics*, 3, 3–51.

SMITH, D. R. (1991): "A Simple Method for Pricing Interest Rate Swaptions," *Financial Analysts Journal*, May–June, 72–76.

SOBOL, I. M. (1967): "On the Distribution of Points in a Cube and the Approximate Evaluation of Integrales," *USSR Journal of Computational Mathematics and Mathematical Physics* (English Translation), 7, 784–802.

SORTINO, F., AND L. PRICE (1994): "Performance Measurment in a Downside Risk Framework," *The Journal of Investing*, 59–65.

SPRENKLE, C. (1964): "Warrant Prices as Indicators of Expectations and Preferences" in *The Random Character of Stock Market Prices*, ed. P. Cootner (Cambridge, MA: MIT Press).

STULZ, R. M. (1982): "Options on the Minimum or the Maximum of Two Risky Assets," *Journal of Financial Economics*, 10, 161–185.

SU, Y., AND M. FU (2000): "Optimal Importance Sampling in Securities Pricing," Working paper, Robert H. Smith School of Business.

TALEB, N. (1997): *Dynamic Hedging* (New York: John Wiley & Sons).

THORP, E. O. (1969): "Optimal Gambling Systems for Favorable Games," *Review of the International Statistics Institute*, 37(3).

THORP, E. O., AND S. T. KASSOUF (1967): *Beat the Market*. (New York: Random House).

TOPPER, J. (2005): "Financial Engineering with Finite Elements" (New York: John Wiley & Sons).

TREYNOR, J. (1965): "How to Rate Management of Investment Funds," *Harvard Business Review*, 43, 63–75.

TURNBULL, S. M., AND L. M. WAKEMAN (1991): "A Quick Algorithm for Pricing European Average Options," *Journal of Financial and Quantitative Analysis*, 26, 377–389.

VASICEK, O. (1977): "An Equilibrium Characterization of the Term Structure," *Journal of Financial Economics*, 5, 177–88.

VILLIGER, R. (2005): "Valuation of American Call Options," *Wilmott Magazine*.

WEBB, A. (1999): "The Sensitivity of Vega," *Derivatives Strategy*, November, 16–19.

WEST, G. (2005a): "Calibration of the SABR Model in Illiquid Markets," *Applied Mathematical Finance*, 12(4), 371–385.

—— (2005b): "Better Approximations to Cumulative Normal Functions," *Wilmott Magazine*.

WHALEY, R. E. (1981): "On the Valuation of American Call Options on Stocks with Known Dividends," *Journal of Financial Economics*, 9, 207–11.

WILCOX, D. (1991): "Spread Options Enchance Risk Managment Choices," *NYMEX Energy in the News*, Fall, 9–13.

WILMOTT, P. (1998): *Paul Wilmott Derivatives* (New York: John Wiley & Sons).

—— (2000): *Paul Wilmott on Quantitative Finance* (New York: John Wiley & Sons).

WILMOTT, P., J. DEWYNNE, AND S. HOWISON (1993): *Option Pricing: Mathematical Models and Computation* (Oxford, UK: Oxford Financial Press).

WU, X., AND J. E. ZHANG (1999): "Options on the Minimum or the Maximum of Two Average Prices," *Review of Derivatives Research*, (2), 183–204.

WYSTRUP, U. (1999): "Aspects of Symmetry and Duality of the Black-Scholes Pricing Formula for European Style Put and Call Options," Working paper, Sal. Oppenhim jr. & Cie.

—— (2002): "Vanilla Options," in *Foreign Exchange Risk*, J. Hakala and U. Wystrup (London: Risk Books).

ZHANG, P. (1994): "Flexible Asian Options," *Journal of Financial Engineering*, 3(1), 65–83.

—— (1995a): "Correlation Digital Options," *Journal of Financial Engineering*, 4, 75–96.

—— (1995b): "Flexible Arithmetic Asian Options," *Journal of Derivatives*, 2(3), 53–63.

—— (1998): *Exotic Options: A Guide to the Second Generation Options*, 2nd ed. (Singapore: World Scientific).

INDEX

Page numbers followed by n indicate footnotes.