

*Pricing  
Financial  
Instruments*

## **WILEY SERIES IN FINANCIAL ENGINEERING**

*Derivatives Demystified: Using Structured Financial Products*

John C. Braddock

*Option Pricing Models*

Les Clewlow and Chris Strickland

*Derivatives for Decision Makers: Strategic Management Issues*

George Crawford and Bidyut Sen

*Currency Derivatives: Pricing Theory, Exotic Options, and Hedging Applications*

David F. DeRosa

*Options on Foreign Exchange, Revised Edition*

David F. DeRosa

*The Handbook of Equity Derivatives, Revised Edition*

Jack Francis, William Toy, and J. Gregg Whittaker

*Dictionary of Financial Engineering*

John F. Marshall

*Interest-Rate Option Models: Understanding, Analyzing, and Using Models for Exotic Interest-Rate Options*

Ricardo Rebonato

*Derivatives Handbook: Risk Management and Control*

Robert J. Schwartz and Clifford W. Smith, Jr.

*Dynamic Hedging: Managing Vanilla and Exotic Options*

Nassim Taleb

*Credit Derivatives: A Guide to Instruments and Applications*

Janet Tavakoli

*Pricing Financial Instruments: The Finite Difference Method*

Domingo Tavella and Curt Randall

# *Pricing Financial Instruments*

**THE FINITE  
DIFFERENCE METHOD**

Domingo Tavella

Curt Randall



John Wiley & Sons, Inc.

New York • Chichester • Weinheim • Brisbane • Singapore • Toronto

This book is printed on acid-free paper. @

Copyright © 2000 by Domingo Tavella and Curt Randall. All rights reserved.

Published by John Wiley & Sons, Inc.

Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4744. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 605 Third Avenue, New York, NY 10158-0012, (212) 850-6011, fax (212) 850-6008, E-Mail: PERMREQ@WILEY.COM.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in rendering professional services. If professional advice or other expert assistance is required, the services of a competent professional person should be sought.

*Library of Congress Cataloging-in-Publication Data:*

Tavella, Domingo, 1948-

Pricing financial instruments : the finite difference method / Domingo Tavella, Curt Randall.

p. em. - (Wiley series in financial engineering)

Includes bibliographical references and index.

ISBN 0-471-19760-2 (cloth: alk. paper)

1. Options (Finance)-Prices. 2. Financial instruments-Prices. I. Randall,

Curt, 1951-. II. Series.

HG6024.A3 T38 2000

332.64'5-dc21

99-054917

Printed in the United States of America.

*To Susan and Nicole  
and  
To Montel and James*

# Preface

The pricing of financial instruments by numerical solution of their pricing equation has become an important component in the arsenal of techniques available to practitioners of modern quantitative finance. The price of financial derivatives follows, in many cases, from the solution of parabolic partial differential equations or from formulations involving parabolic partial differential equations. The demand for complex financial instruments and the availability of powerful desktop computers make the direct numerical solution of the governing pricing equation an increasingly appealing approach to pricing.

Numerical methods for the solution of parabolic partial differential equations have had broad application in several areas of science for many years. The advent of high-speed computational power at low cost made it possible to bring numerical techniques for the solution of partial differential equations to a high level of sophistication. The 1980s witnessed the emergence of numerical solutions of partial differential equations as a computational laboratory, where new concepts and designs could be explored by resolving their basic mathematical description, with little or no simplification. Computational aerodynamic design, numerical weather modeling, computational chemistry, and the modeling of nuclear devices are some significant examples.

This book presents a framework for applying the method of finite differences to the pricing of financial derivatives. This work is not meant to be a catalog of numerical techniques or to present a detailed listing of option pricing exercises. It is structured with an emphasis on understanding how numerical schemes work, and how the approximations involved affect the accuracy of the solution. We hope that the material presented here will help practitioners implement and interpret the output of standard numerical schemes sensibly. A basic approach to finite differences is adopted in this book. No attempt was made to discuss specialized

techniques, such as multigrids, upwinding, total variation diminishing schemes, or alternating direction schemes. Some of these solution strategies are currently the object of research.

This book is aimed primarily at practitioners, but can be used as a reference text in financial engineering coursework. Readers are expected to have some background in derivative pricing and linear algebra. Throughout the book, the attempt was made to reduce mathematical complexity to a minimum. This is especially the case with probability concepts, whose invocation is very limited. As a result, many of the financial concepts presented are not rigorously justified, and the reader interested in greater depth should refer to some of the outstanding sources on mathematical finance currently available.

The book is divided into six chapters. Chapter 1 is a summary introduction to stochastic processes and arbitrage pricing arguments. The discussion is at an informative level, and while it does not claim to be rigorous, it should be useful to analysts new to the field of quantitative finance.

Chapter 2 details the derivation of the pricing equations from a mathematical viewpoint, with no particular emphasis on specific types of instruments. This chapter offers a brief description of the linear complementary approach to pricing American derivatives. The chapter presents a discussion on dimensionality reduction using currency translated options as an example.

Chapter 3 presents a practical framework for analyzing numerical schemes and understanding the implications of discretization. The focal point of the chapter is the analysis of stability, which is used as a vehicle to discuss the properties of finite difference schemes.

Chapter 4 is dedicated to implementational issues that present significant challenges when considering the practical implementation of finite difference solutions. Issues such as jumps, boundary conditions, initial conditions, and path dependency are discussed in detail.

Chapter 5 deals with coordinate transformations in a general sense. This chapter provides the reader with the concrete information needed to enhance the flexibility of numerical schemes through

change of coordinates in a manner much more general than is typically treated in most finance textbooks.

Chapter 6 is a small collection of practical case studies. Although it is by no means comprehensive, the examples discussed are rich enough in features to demonstrate the capabilities of the methodology.

DOMINGO TAVELLA

CURT RANDALL

*San Francisco, California  
Austin, Texas  
March 2000*

# Acknowledgments

During the preparation of this work, we benefited greatly from discussions with colleagues.

Special thanks are due to Dr. Ervin Zhao, Dr. Eugene Wong, Dr. Ronald Lagnado, Dr. Raymond Hawkins, and Professor Francis Longstaff for their helpful comments on the manuscript.

D. T.  
c.R.

# Contents

<b>1 Introduction</b>	<b>1</b>
Stochastic Processes	3
~arkovProcesses	5
Stochastic Differential Equations	8
Ito's Formula	9
Ito's Formula for Processes with Jumps	10
Arbitrage Pricing Theory	13
Change of ~measure	16
References	21
<b>2 The Pricing Equations</b>	<b>23</b>
European Derivatives	24
Hedging Portfolio Approach	24
Feynman-Kac Approach	27
The Pricing Equation in the Presence of Jumps	30
An Application of Jump Processes:	
Credit Derivatives	34
Defaultable Bonds	37
Full Protection Credit Put	38
American Derivatives	39
Relationship between European and American Derivatives	40
American Options as Dynamic Optimization Problems	42
Conditions at Exercise Boundaries	43
Linear Complementarity Formulation of American Option Pricing	44
Path Dependency	45
Discrete Sampling of Path Dependency	47
Dimensionality Reduction	48

Reformulating the Underlying Processes in a Different Measure	49
Currency Translated Options	50
Equations for the Hedging Parameters	56
Computation of Greeks by Direct Discretization	57
Computation of Greeks through Their Governing Equations	57
References	60
<b>3 Analysis of Finite Difference Methods</b>	<b>61</b>
Motivation	61
Constructing Finite Difference Approximations	67
Stability Analysis: Matrix Approach	70
Space Discretization	71
Time Discretization	73
Analysis of Specific Algorithms	77
Eigenvalue Analysis of the Black-Scholes Equation	86
Stability Analysis: Fourier Approach	90
Implementation of the Time Advancement	93
Solving Sparse Systems of Linear Equations	94
Finite Difference Approach to American Options	100
The Linear Complementarity Problem	101
Distortions Induced by Discretization	105
Strategies for Complex Derivative Structures	107
References	108
<b>4 Special Issues</b>	<b>110</b>
Effect of Payoff Discontinuities on Convergence	110
Implementing Jump Conditions	114
Boundary Conditions	120
Boundary Conditions in One Dimension	121
Boundary Conditions in Multiple Dimensions	130
Continuous and Discrete Sampling Models for	
Path-Dependent Options	132
Continuous Sampling	132
Discrete Sampling	136

Performance of Solvers for Multidimensional Problems	141
Numerical Solution of PIDEs: Jump-Diffusion and Pure Jump Models	147
References	155
<b>5 Coordinate Transformations</b>	<b>156</b>
One- Dimensional, Time-Independent Transformations	157
Transformations Place Grid Points at Selected Positions	160
Transformations That Concentrate Grid Points	167
One-Dimensional, Time-Dependent Transformations	172
Multidimensional, Time-Independent Transformations	173
Factored Multidimensional, Time-Independent Transformations	174
General Multidimensional, Time-Independent Transformations	175
Multidimensional Linear Transformations	177
References	182
<b>6 Numerical Examples</b>	<b>183</b>
Barrier Options	183
Time-Dependent Barriers	183
Nonuniform Grids and Discrete Sampling	187
Discretely Sampled Parisian Options	196
A Leveraged Knockin Put	202
Discretely Sampled Asian Options	206
Stochastic Volatility	212
Convertible Bond	214
Simple Fixed Income Instruments: Forward Swap	218
Credit Derivatives	223
References	228

*Pricing  
Financial  
Instruments*

*Chapter*    **I**

# Introduction

This book presents a basic framework for pricing financial instruments through the numerical solution of partial differential equations (PDEs) using finite differences. The finite difference method has been applied to pricing financial contracts for many years. The pioneering work of Brennan and Schwartz (1977) is an early example of the application of finite differences to pricing financial instruments. Although the finite difference method is extremely powerful, its popularity has historically lagged that of trees and simulation. The main requirement to use the finite difference method effectively is a good understanding of its fundamentals. This understanding is essential to determine which problems can be solved with this method, as well as to properly assess the quality of the solution. Once this understanding of the fundamentals is achieved, the superior power and flexibility of this method can be realized in the valuation of many types of securities that would defy their analysis through either trees or simulation. Recently, as the financial community has become increasingly more aware of the potential of this methodology, the popularity of finite differences has been on the rise.

The material presented in this chapter is a brief synthesis of established results needed to derive the pricing equations. The basic principle underlying the pricing of financial instruments is the arbitrage pricing principle, introduced in the early 1970s by Black and Scholes (1973) and Merton (1973).

The arbitrage pricing principle applies to financial instruments whose cash flows are contractually related to the values of other securities or to economic factors that determine the value of other securities.

## 2 Introduction

For the purpose of pricing, financial instruments can be classified into two fundamental types. The first type consists of those whose cash flows cannot be influenced by the holder of the instrument. These are referred to as European instruments. The second type includes those whose cash flows can be influenced by the holder of the instrument. We will refer to these instruments as American. In general, however, the marketplace has agreed on a variety of names that reflect the nature of the holder's influence-on the cash flows of the instrument. For example, Bermudan options, which can be exercised at prescribed points in time, derive their name from the fact that Bermuda is situated between America and Europe. The holder of an American instrument influences the cash flows of the instrument through what we will refer to as an *exercise strategy*.

In the case of a European financial instrument, the application of the arbitrage pricing principle reduces the valuation of the instrument to the computation of an expectation over the space of the underlying factors that determine the instrument's price. This is the expectation of suitably discounted cash flows under risk-neutral probability distributions (Harrison & Kreps, 1979; Harrison & Pliska, 1981). Risk-neutral refers to the distribution of outcomes of processes in a world where investors are indifferent to risk. Under suitable conditions, computing this expectation reduces to the solution of a partial differential equation. Historically, the best-known form of these equations is the Black-Scholes equation.

For an American instrument, application of the arbitrage pricing principle reduces the pricing problem to the computation of the maximum, over all exercise strategies, of an expectation over the space of the underlying factors that determine the instrument's price. Under suitable conditions, computing this expectation reduces to the solution of a partial differential complementarity problem (Jaillet, Lamberton, & Lapeyre, 1990).

If the optimal exercise strategy were known, the pricing of an American instrument would be no different than the pricing of a European instrument. Because the optimal exercise strategy is not known in advance, the analytical pricing of American-style instruments is far more difficult than that of European ones. However, as will be seen through the remainder of this book, the pricing of American options via the numerical solution of PDE is not significantly harder than the pricing of European options.

In some instances (e.g., some European options), the expectations that result from applying the arbitrage pricing principle can be computed analytically. In other cases (e.g., American options), these expectations are more conveniently characterized as solutions to partial differential equations or linear complementarity problems. In yet other instances, the expectations are best computed through simulation. Simulation can be a winning strategy for certain path-dependent securities where it is not possible to eliminate the path-dependency by augmenting the dimensionality of the problem in a reasonable manner, as well as where the number of dimensions is high.

The numerical and computational challenges in pricing financial instruments arise in many contexts. Among them, a significant one is dimensionality. As the number of underlying processes increases, the computational requirements may favor one methodology over another. Simulation, for example, is more capable of dealing with a large number of dimensions than partial differential equations or trees. The underlying processes may pose another significant challenge. Some formulations of Asian options, for example, may require particularly robust numerical procedures if a PDE approach is used (Zvan, Forsyth, & Vetzal, 1998).

This book does not attempt to present a catalog of numerical solutions. Rather, it provides a framework for a better understanding of a particular methodology that is becoming increasingly popular; the solution by finite differences of the partial differential equations that describe the price of the security. The remainder of this chapter presents a succinct description of the main points of stochastic processes and arbitrage pricing theory. These are the fundamental principles on which the numerical methodologies are based. For a rigorous exposition and proofs, the reader is referred to more specialized sources in the various topics as they are discussed.

## STOCHASTIC PROCESSES

The subject of stochastic processes has been treated extensively by several authors. Three excellent sources are the works by Gillespie (1992), Van Kampen (1992), and Gardiner (1994).

We will consider a vector of random variables,  $X(t)$ , with realizations  $x_1, x_2, \dots$  at times  $t_1, t_2, \dots, t_n$ . We will define a stochastic

#### 4 Introduction

process to be a function that depends on  $X$  (including, of course,  $X$  itself). We will assume that the observation times are ordered as follows:  $t_1 \sim t_2 \dots t_n$

We are interested in stochastic processes that are completely defined by their joint probability density function (JPDF), and their conditional probability density function (CPDF):

$$JPDF = p(x_n, t_n; x_{n-1}, t_{n-1}; \dots; x_1, t_1) \quad (1.1)$$

$$CPDF = p(x_n, t_n; x_k, t_k; \dots; x_{k-1}, t_{k-1}; \dots; x_1, t_1) \quad (1.2)$$

In financial applications,  $X(t)$  will be the underlying processes that determine the price of a derivative. In the case of a derivative that depends on a single equity, for example,  $X(t)$  could be a one-dimensional process representing the underlying stock price returns. In a basket option (an option structured on the sum of several underlying processes),  $X(t)$  would be a multidimensional process representing the price returns of the underlying assets. In general,  $X(t)$  stands for a suitable description of processes such as interest rates, foreign exchange rates, forward rates, and the price process of other derivative securities.

We are interested in answering the three fundamental questions:

1. What is a suitable description of the array of underlying financial processes,  $X(t)$ ?
2. What is the description of a function of the underlying processes?
3. How do we use the description of the function of the underlying processes to price a derivative security?

The first question refers to the formulation of a mathematical model for the evolution of underlying processes. In many textbooks, this question is addressed by postulating a plausible stochastic differential equation (SDE) for the price processes (Hull, 1997). This leads to the postulation of the standard Wiener process for the description of continuous underlying processes. Discontinuous processes are

typically modeled as Poisson processes. Here, we address this question by exploring the evolution of  $X(t)$  suggested by the probability density functions. If the underlying processes are markovian and continuous, their conditional probability density function is described by the Fokker-Planck equation. The Fokker-Planck equation, in its turn, has solutions that correspond to the probability densities of generalized Wiener processes. This approach exposes the underlying assumptions (Gardiner, 1994).

The second question leads to the presentation of a fundamental tool for manipulating stochastic processes: Ito's formula. This formula gives an explicit expression for the time evolution of a function of a stochastic process. Since the price of a derivative is a function of the underlying processes, direct application of Ito's formula may lead to a description of the price process for the derivative we are trying to price. Use and applicability of Ito's formula require a specific interpretation of the SDE describing the underlying processes. Ito's formula is consistent with the evolution of probability density given by the Fokker-Planck equation.

The third question, in the context of this book, addresses the issue of obtaining a PDE for the derivative's price. Here is where arbitrage pricing enters the picture. As shown in Chapter 2, the fact that Ito's lemma allows us to obtain an expression for the process of the derivative security can be exploited to obtain a PDE for the derivative's price by invoking arbitrage pricing. This principle can be invoked by establishing hedging relationships or by using a fundamental property of expectations, as expressed in the Feynman-Kac theorem. In either case, under suitable conditions we obtain a partial differential equation (or partial differential complementarity) formulation of the instrument's price.

The following discussion of the Fokker-Planck equation suggests to us the appropriate SDE followed by the underlying processes  $X(t)$ .

## `~arkovProcesses`

In a general stochastic process, the conditional probability density functions are severely constrained. These constraints are the result of consistency relationships that must be satisfied by the conditional densities. For example, two such consistency relationships are the following:

## 6 Introduction

$$\begin{aligned} p(\mathbf{x}_n, t_n) &= \int p(\mathbf{x}_n, t_n; \mathbf{x}_{n-1}, t_{n-1}) d\mathbf{x}_{n-1} \\ &= \int p(\mathbf{x}_n, t_n | \mathbf{x}_{n-1}, t_{n-1}) p(\mathbf{x}_{n-1}, t_{n-1}) d\mathbf{x}_{n-1} \end{aligned} \quad (1.3)$$

$$\begin{aligned} p(\mathbf{x}_n, t_n | \mathbf{x}_{n-2}, t_{n-2}) &= \int p(\mathbf{x}_n, t_n; \mathbf{x}_{n-1}, t_{n-1} | \mathbf{x}_{n-2}, t_{n-2}) d\mathbf{x}_{n-1} \\ &= \int p(\mathbf{x}_n, t_n | \mathbf{x}_{n-1}, t_{n-1} | \mathbf{x}_{n-2}, t_{n-2}) \\ &\quad p(\mathbf{x}_{n-1}, t_{n-1} | \mathbf{x}_{n-2}, t_{n-2}) d\mathbf{x}_{n-1} \end{aligned} \quad (1.4)$$

These equations ensure the consistency of the conditional probabilities (Gillespie, 1992). In general, there are infinitely many such relationships. Evidently, the characterization and analysis of stochastic process by their probability distributions is a difficult task. This task is dramatically simplified if the future evolution of the process is entirely determined by its most recent state. Such a process is called *markovian*. In finance, the markovian assumption is connected with the notion that the market will evolve on the basis of the most recently revealed information.

If the process for X is markovian, the following relationships must hold:

$$p(\mathbf{x}_n, t_n; \dots \mathbf{x}_k, t_k | \mathbf{x}_{k-1}, t_{k-1}; \dots \mathbf{x}_1, t_1) = p(\mathbf{x}_n, t_n; \dots \mathbf{x}_k, t_k | \mathbf{x}_{k-1}, t_{k-1}) \quad (1.5)$$

With this relationship, the consistency relations to be satisfied by the conditional density functions simplify significantly and become what is known as the Chapman-Kolmogorov equation:

$$p(\mathbf{x}_{k+1}, t_{k+1} | \mathbf{x}_{k-1}, t_{k-1}) = \int p(\mathbf{x}_{k+1}, t_{k+1} | \mathbf{x}_k, t_k) p(\mathbf{x}_k, t_k | \mathbf{x}_{k-1}, t_{k-1}) d\mathbf{x}_k \quad (1.6)$$

Until now, we haven't said anything about the continuity of the stochastic process X. The Chapman-Kolmogorov equation is valid for continuous or discontinuous Markov processes. If we now require that the Markov process be continuous, the Chapman-Kolmogorov equation becomes the Fokker-Planck equation for the conditional density of the stochastic process. The requirement of

continuity of the Markov process  $X$  is relatively easy to see if we look at the process *path*. Here, however, we need to consider the impact of continuity on the process conditional *density*. It has been shown that if a Markov process is continuous, the density of a process conditional on an initial state tends to zero faster than the time distance from the initial state (Gihman & Skorohod, 1975). This is expressed as follows:

$$\lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} p(\mathbf{y}, t + \Delta t | \mathbf{x}, t) = 0 \text{ for all } |\mathbf{y} - \mathbf{x}| > 0 \quad (1.7)$$

We introduce the following definitions:

$$a_i(\mathbf{x}, t) = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \int (y_i - x_i) p(\mathbf{y}, t + \Delta t | \mathbf{x}, t) d\mathbf{y} \quad (1.8)$$

$$b_{ij}(\mathbf{x}, t) = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \int (y_i - x_i)(y_j - x_j) p(\mathbf{y}, t + \Delta t | \mathbf{x}, t) d\mathbf{y} \quad (1.9)$$

The forward version of the Fokker-Planck equation describes the future evolution of the probability density of a process conditional on an initial state. (For details of the derivation, please refer to Gardiner, 1994).

$$\begin{aligned} \frac{\partial p(\mathbf{x}, t | \hat{\mathbf{x}}, \hat{t})}{\partial t} = & - \sum_{i=1}^{i=n} \frac{\partial}{\partial x_i} (a_i(\mathbf{x}, t) p(\mathbf{x}, t | \hat{\mathbf{x}}, \hat{t})) \\ & + \frac{1}{2} \sum_{i,j=1}^{i,j=n} \frac{\partial^2}{\partial x_i \partial x_j} (b_{ij}(\mathbf{x}, t) p(\mathbf{x}, t | \hat{\mathbf{x}}, \hat{t})) \end{aligned} \quad (1.10)$$

The initial condition for the Forward Fokker-Planck equation is

$$p(\mathbf{x}, t | \hat{\mathbf{x}}, \hat{t}) = \delta(\mathbf{x} - \hat{\mathbf{x}}) \quad (1.11)$$

where  $\delta(\cdot)$  is the delta function. This initial condition reflects that  $\hat{\mathbf{x}}$  at time  $\hat{t}$  is known. A local solution of this equation suffices to tell

## 8 Introduction

us a plausible description of the underlying processes that are of interest in finance.

### Stochastic Differential Equations

To show how the Fokker-Planck equation suggests both the form and the appropriate interpretation of a continuous stochastic process, we will find an approximate solution of Equation 1.10 around an initial state  $p(\mathbf{x}, t | \hat{\mathbf{x}}, \hat{t}) = \delta(\mathbf{x} - \hat{\mathbf{x}})$ . This simple analysis, explained in greater detail in Gardiner (1994), leads to the conclusion that the probability density of  $\mathbf{x}$  at  $\hat{t} + \Delta t$  is to first order a normal distribution with mean  $\hat{\mathbf{x}} + \mathbf{a}(\hat{\mathbf{x}}, \hat{t})$  and covariance matrix  $\mathbf{b}(\hat{\mathbf{x}}, \hat{t})\Delta t$ . This means that the solution can be written as follows:

$$\mathbf{x}(\hat{t} + \Delta t) = \hat{\mathbf{x}}(\hat{t}) + \mathbf{a}(\hat{\mathbf{x}}, \hat{t})\Delta t + \mathbf{z}(\hat{\mathbf{x}}, \hat{t})\sqrt{\Delta t} \quad (1.12)$$

where

$$E[\mathbf{z}(\hat{\mathbf{x}}, \hat{t})] = 0 \quad (1.13)$$

and

$$E[\mathbf{z}(\hat{\mathbf{x}}, \hat{t})\mathbf{z}(\hat{\mathbf{x}}, \hat{t})^T] = \mathbf{b}(\hat{\mathbf{x}}, \hat{t}) \quad (1.14)$$

where  $E$  denotes the expectation operator.

This solution indicates that  $\mathbf{x}$  is continuous, but due to the square root term, it is not differentiable. The second term on the right of Equation 1.12 is a drift; the third represents random changes.

We will next explore the properties of the standard Wiener process,  $W(t)$ . For simplicity, consider the one-dimensional case. This process results from solving the Fokker-Planck equation with  $a = 0$  and  $b = 1$ :

$$\frac{\partial p(W(t) | 0)}{\partial t} = \frac{1}{2} \frac{\partial^2 p(W(t) | 0)}{\partial W^2} \quad (1.15)$$

subject to the initial condition

$$p(W(t=0)) = \delta(W) \quad (1.16)$$

The solution is

$$p(W(t) | 0) = \sqrt{\frac{1}{2\pi t}} \exp\left(-\frac{W(t)^2}{2t}\right) \quad (1.17)$$

This indicates that  $W(t)$  is Gaussian with zero mean and variance equal to  $t$ .

The local solution we obtained previously and the properties of the standard Wiener process suggest that we will be interested in processes of the form

$$dx = \mu(x, t) dt + \sigma(x, t) dW \quad (1.18)$$

where  $\sigma^2$  is the variance per unit time of  $dx$ , and  $W$  is the standard Wiener process.

This equation would not be meaningful unless a statement is made about the interpretation of the differentials. When the interpretation is as described by Equation 1.12, this equation is referred to as an Ito equation, and the corresponding forward Fokker-Planck equation is

$$\begin{aligned} \frac{\partial p(x, t | x(0), 0)}{\partial t} &= -\frac{\partial}{\partial x} (\mu(x, t) p(x, t | x(0), 0)) \\ &\quad + \frac{1}{2} \frac{\partial^2}{\partial x^2} (\sigma^2(x, t) p(x, t | x(0), 0)) \end{aligned} \quad (1.19)$$

## Ho's Formula

Ito's formula (also known as Ito's lemma) is the basic tool for determining the Ito process followed by a function of other Ito processes. For this reason, Ito's formula is also referred to as the change of variable formula. Derivations of Ito's formula at various levels of refinement can be found in many texts (e.g., Durrett, 1996; Karatzas & Shreve, 1988).

Given a twice-differentiable function  $G(x)$  of the process described by Equation 1.18, Ito's formula states:

$$dG = \frac{\partial G}{\partial x} dx + \frac{1}{2} \sigma^2 \frac{\partial^2 G}{\partial x^2} dt \quad (1.20)$$

## 10 Introduction

This equation is what would result if we expand  $G(x + dx)$  in Taylor's series and replace  $dW^2$  with  $dt$ . An alternative form of Ito's formula follows by replacing  $dx$  in Equation 1.20:

$$dG = \left( \mu \frac{\partial G}{\partial x} + \frac{1}{2} \sigma^2 \frac{\partial^2 G}{\partial x^2} \right) dt + \sigma \frac{\partial G}{\partial x} dW \quad (1.21)$$

Although this is a working form of Ito's lemma, keep in mind that this representation really refers to a relationship between the following two stochastic integrals:

$$x(t) = x(t_0) + \int_{t_0}^t \mu(x(s), s) ds + \int_{t_0}^t \sigma(x(s), s) dW(s) \quad (1.22)$$

$$G(t) = G(t_0) + \int_{t_0}^t \frac{\partial G}{\partial x} dx(s) + \frac{1}{2} \int_{t_0}^t \sigma^2 \frac{\partial^2 G}{\partial x^2} dW(s) \quad (1.23)$$

## Ito's Formula for Processes with Jumps

Ito's lemma can also be formulated for processes with jumps. A good source for this analysis is the book by Kushner and Dupuis (1992). In finance, a useful characterization of jumps is given by the Poisson process. Consider a process  $x(t)$  that may experience a jump in the interval  $t, t + \Delta t$ . Assume that if process  $x(t)$  does not jump, it remains constant. Evidently, for this process to be markovian, both the probability of a jump in  $t, t + \Delta t$  and the intensity of the jump must be functions of the process immediately before the jump. Let  $x(t^-)$  denote the limit of the process as it approaches the interval  $t, t + \Delta t$  from the left. The following notation characterizes the change of the process as a result of a jump:

$$dx(t) = \lim_{\Delta t \rightarrow 0} (x(t + \Delta t) - x(t^-)) \quad (1.24)$$

A process  $x(t)$  is said to be subject to Poisson jumps with intensity  $h(x(t))$  if  $x(t)$  will jump once in the interval  $t, t + \Delta t$  with probability

$h(x(t^-))\Delta t + O(\Delta t)$ . We can view  $h(x(t^-))$  as the instantaneous probability that one jump will occur. Given that a jump has occurred, the level of the jump is given by a probability distribution that is defined at time  $t^-$ . We denote the level of the jump of  $x(t^-)$  by  $z$ , and the probability density of the jump by  $\eta(x, z)$ . Immediately after the jump, the value of  $x$  will be  $x(t^-) + z$ .

It can be shown that we can derive a version of the Ito formula that accommodates Poisson jumps (Kushner & Dupuis, 1992). Since the purpose of Ito's formula is to derive the process for a function of a stochastic process, we must consider two jumps. One jump is the jump undergone by the underlying process  $x(t)$ . The other is the jump undergone by the function  $G(x(t))$  as a result of the jump in  $x(t)$ . We can denote the jump process for  $x(t)$  by  $J_x(t)$ , and the jump process for  $G(x(t))$  by  $J_G(t)$ . Assuming that jumps occur at random times  $\tau$ ,  $J(x(t))$  will be a martingale (a process is called a martingale if its expectation at time  $t > s$ , conditional on information available at  $s$ , is equal to the realization of the process at  $s$ ) if we define it as follows:

$$J_x(t) = \sum_{\tau \leq t} [x(\tau) - x(\tau^-)] - \int_0^t dsh(x(s)) \int z\eta(x(s), z) dz \quad (1.25)$$

where the second integral is over all possible values of  $z$ . The first term on the right is the sum total of the jump values that have occurred in  $0, t$ . Now assume that the process  $x(t)$  is the superposition of an Ito process and a Poisson jump process. Such a process is described as follows:

$$dx(t) = (\mu + h(x(t))) \int z\eta(x(t), z) dz dt + \sigma dW(t) + dJ_x(t) \quad (1.26)$$

where the jump increment is given by

$$dJ_x(t) = x(t) - x(t^-) - (h(x(t))) \int z\eta(x(t), z) dz dt \quad (1.27)$$

Notice that if there was a jump at time  $t$ , the first term on the right will be nonzero.

## 12 Introduction

We now obtain a version of Ito's formula that precisely parallels the conventional Ito's formula for continuous processes. For a function  $G(x(t), t)$ :

$$dG(x(t), t) = \left( \frac{\partial G}{\partial t} + \mu \frac{\partial G}{\partial x} + \frac{1}{2} \sigma^2 \frac{\partial^2 G}{\partial x^2} + h(x(t)) \int [G(x(t) + z, t) - G(x(t), t)] \eta(x(s), z) dz \right) dt + \frac{\partial G}{\partial x} \sigma dW(t) + dJ_G(t) \quad (1.28)$$

where the  $dJ_G$  is the induced change in  $G(x(t), t)$  as a result of the jump in  $x(t)$ . This change is given by:

$$dJ_G(t) = G(x_+(t), t) - G(x(t^-), t) - (h(x(t)) \int [G(x(t) + z, t) - G(x(t))] \eta(x(t), z) dz) dt \quad (1.29)$$

and the integrated jump process is

$$J_G(t) = \sum_{\tau \leq t} [G(x(\tau)) - G(x(\tau^-))] - \int_0^t h(x(s)) \int [G(x(t) + z) - G(x(t))] \eta(x(s), z) dz ds \quad (1.30)$$

As shown in Chapter 2, we will be particularly interested in the expected change of  $G(x(t))$  as a result of both diffusion and jumps undergone by  $x(t)$ . Because  $J_G(t)$  is a martingale, this expectation is given by

$$\mathbb{E}[dG(x(t))] = \left( \frac{\partial G}{\partial t} + \mu \frac{\partial G}{\partial x} + \frac{1}{2} \sigma^2 \frac{\partial^2 G}{\partial x^2} + h(x(t)) \int [G(x(t) + z, t) - G(x(t))] \eta(x(t), z) dz \right) dt \quad (1.31)$$

This equation will prove useful in determining the risk-neutral growth of the price of a derivative that depends on a jump diffusion process.

### ARBITRAGE PRICING THEORY

Arbitrage pricing theory is the subject of extensive discussion in specialized texts. For particularly clear expositions, the reader is referred to the excellent works by Duffie (1996), Dothan (1990), Karatzas and Shreve (1988), and Musiela and Rutkowski (1998). Our purpose here is to give the most succinct possible statements that will enable the reader to undertake the formulation of partial differential equations for pricing.

To formulate the basic statements of arbitrage-free pricing, we assume that trading of financial assets occurs in a given currency, within what we call a domestic market:

- Absence of arbitrage implies the existence of a probability measure, equivalent to the actual probability measure of the market, under which all traded assets that do not pay dividends have an expected instantaneous return equal to the risk-free rate of interest.
- If in addition, the market is complete, the probability measure is unique. That the market is complete means that any asset can be hedged (replicated) with a portfolio of other assets in the market.

The probability measure under which returns equal the risk-free rate is referred to as the risk-neutral measure. That the risk-neutral measure is equivalent to the probability measure of the market simply means that events that cannot happen under the market measure cannot happen under the risk-neutral measure either. A risk-neutral world then is one where the likelihood of events that can occur in the market is distorted in a manner that all non-dividend-paying securities yield the same instantaneous return. This distortion is accomplished by appropriate changes in the drift of the relevant stochastic processes. We will refer to the drift of

## 14 Introduction

a stochastic process in the risk-neutral measure as the risk-neutral drift. In many cases, finding this risk-neutral drift may involve significant mathematical and financial ingenuity.

Because an asset in the risk-neutral measure has an expected return equal to the risk-free rate, the ratio between the asset price and the price of a money market account (an investment that is continuously compounded at the instantaneous risk-free rate) is a process without drift. This follows from a simple application of Ito's lemma. The money market account,  $\beta(t)$ , has initial value 1 and is governed by the following process, where  $r(t)$  is the instantaneous risk-free rate at time  $t$ :

$$d\beta(t) = \beta(t)r(t)dt \quad (1.32)$$

Consider now a derivative security with the following price process:

$$\frac{dV}{V} = rdt + \sigma_v dW \quad (1.33)$$

where  $W$  is a Brownian motion in the risk-neutral world.

Ito's lemma gives the following process for the ratio  $\frac{v}{\beta}$ .

$$d\left(\frac{V}{\beta}\right) = \frac{V}{\beta} \sigma_v dW \quad (1.34)$$

This equation has the solution

$$\frac{V(t)}{\beta(t)} = \frac{V(0)}{\beta(0)} \exp\left(-\frac{1}{2} \int_0^t \sigma_v^2 ds + \int_0^t \sigma_v dW_s\right) \quad (1.35)$$

Taking expectations of both sides of this equation and using the expression

$$E_0 \left[ \exp \left( \int_0^t \sigma_v dW_s \right) \right] = \exp \left( \frac{1}{2} \int_0^t \sigma_v^2 ds \right) \quad (1.36)$$

we get

$$V(0) = E_0 \left[ \frac{V(t)}{\beta(t)} \right] \quad (1.37)$$

If the security matures at time  $T$  where it has a known payoff  $F(T)$ , then  $V(T) = F(T)$  and Equation 1.37 allows us to find the price of the security at time 0:

$$V(0) = E_0 \left[ \frac{F(T)}{\beta(T)} \right] \quad (1.38)$$

If the security has a dividend yield  $y(t)$  (this means that the dividend payment over  $dt$  is  $V(t)y(t)dt$ ), we can derive the risk-neutral process for  $V(t)$  by extending the idea expressed in Equation 1.37 by treating each infinitesimal of dividend payment as a payoff. We would then expect that the relative process for the expression

$$\frac{V(t)}{\beta(t)} + \int_0^t \frac{V(s)y(s)}{\beta(s)} ds \quad (1.39)$$

should be driftless. It is straightforward to see that this happens if the drift of  $V(t)$  is corrected by subtracting the dividend yield:

$$\frac{dV}{V} = (r - y) dt + \sigma_v dW \quad (1.40)$$

We can now price a security that pays a dividend yield and has a payoff  $F(T)$  as follows:

$$V(0) = E_0 \left[ \frac{F(T)}{\beta(T)} \right] + E_0 \left[ \int_0^T \frac{V(s)y(s)}{\beta(s)} ds \right] \quad (1.41)$$

These statements about arbitrage pricing are important for the following reasons.

We have been referring to a market where the risk of the financial assets is not compounded by currency conversions. When the assets in question are traded in foreign markets, the risk-neutral processes of their prices must reflect the risk imposed by foreign exchange. This notion is expanded in Chapter 2.

We now summarize the three reasons why statements about arbitrage pricing are important:

1. If all the relevant underlying processes that determine the price of a derivative are expressed in a risk-neutral world (or measure), the expectation of an infinitesimal change in the relative value of the derivative must grow at the risk-free rate. Combined with Ito's formula, this fact may immediately yield a partial differential equation for pricing the security.
2. The fact that the price of the security is given by an expectation allows us to invoke the Feynman-Kac theorem, which establishes that such expectation is given by the solution to a partial differential equation. Such a partial differential equation will be the pricing equation we seek.
3. In many instances, the invocation of the risk-free return of a security serves as an intermediate step in computing the risk-neutralized drift of other processes, which may not be traded assets. Once the risk-neutralization of a particular process has taken place, the process is available for immediate use in obtaining a pricing equation via Ito's lemma or the Feynman-Kac theorem.

Evidently, to effectively use the arbitrage pricing arguments to derive a pricing equation, it is essential to be able to express the relevant processes in the appropriate measure.

## Change of Measure

Here we briefly summarize the essential elements of measure transformation. The interested reader may consult Duffie (1996) or Karatzas and Shreve (1988) for additional details, including technical considerations. The central element in transforming the measure

of a stochastic process is Girsanov's theorem (Duffie, 1996). We assume that all processes are defined in  $0 \leq t \leq T$ , with  $T$  a finite, fixed time horizon. Before stating Girsanov's theorem, we consider a stochastic process  $\theta(t)$ , which we assume known given the information available at time  $t$ . Such a process is called an adapted process (we must also assume that  $\int_0^t \theta(s)^2 ds < \infty$ ). Girsanov's theorem states that if  $W(t)$  is a standard Wiener process, or Brownian motion, under the probability measure  $P$  (where we denote expectations by  $E_t[\cdot]$ ), then the process

$$W^*(t) = W(t) + \int_0^t \theta(s) ds \quad (1.42)$$

is also a Brownian motion under a different measure, denoted by  $P^*$  (where we denote expectations by  $E_t^*[\cdot]$ ), such that the following relationship holds:

$$E_t^*[X] = \frac{E_t[Z(T)X]}{Z(t)} \quad (1.43)$$

The function  $Z(t)$  is called the Radon-Nikodym derivative and is given by

$$Z(t) = e^{-\int_0^t \theta(s) dW(s) - \frac{1}{2} \int_0^t \theta^2(s) ds} \quad (1.44)$$

The function  $Z(T)$  assigns weights to the realizations of the original Brownian motion,  $W$ , such that the new process,  $W^*$ , is also a Brownian motion.

To see how Girsanov theorem relates to the arbitrage-free pricing statements of the previous section, consider the process for an asset  $S(t)$  with instantaneous return  $\mu$  and with log-return volatility  $\sigma$ :

$$dS = S\mu dt + S\sigma dW \quad (1.45)$$

This equation can be rewritten as follows:

$$dS = (S\mu - S\sigma\theta)dt + S\sigma(dW + \theta dt) \quad (1.46)$$

## 18 Introduction

According to Girsanov's theorem, the quantity  $dW(t) + \theta(t)dt$  is a differential of a Brownian motion under a new probability measure, and  $S\mu - S\sigma\theta(t)$  is the instantaneous return under that measure. From arbitrage-free pricing, we know that the return under the risk-neutral measure must be the risk-free rate. Furthermore, we know that if the market is complete, the risk-neutral measure is unique. This tells us the following:

- The process  $\theta = \frac{\mu - r}{\sigma}$  accomplishes the change to the risk-neutral measure, where the asset process is  $dS = S\mu dt + S\sigma dW^*$ .
- The volatility of the new Wiener process in the risk-neutral measure is the same as before.

This says that to go to a measure where our process has the desired expected returns, we alter the drift of the process and leave its volatility unchanged. We could have arrived at the same conclusion by requiring that the ratio of the asset price to the money market account be a martingale.

Clever measure transformations play an essential role in the analytical pricing of derivatives, where choosing the right measure may greatly facilitate the derivation of a closed-form or a particularly tractable solution. In such cases, it is not only the risk-neutral measure that is important, but other measures as well. The statement that there is a measure under which the ratio of the asset price to the money market account is a martingale can be interpreted as saying that there is a martingale measure associated with the money market account. If we now consider the ratio of the asset price to the price of another traded asset, we can generalize this statement by saying that there is a martingale measure (no longer the risk-neutral measure, of course) associated with this other asset. For practical purposes, practitioners often refer to this fact by saying that assets *induce* a probability measure, under which other traded assets have appropriate returns. For the purpose of deriving pricing equations in the form of partial differential equations, we will concentrate on the risk-neutral measure.

The derivation of the appropriate risk-neutral process may involve significant algebraic manipulation using Ito's lemma. As illustration, we will consider three simple examples. The first example is

the derivation of the risk-neutral process for the foreign exchange rate,  $X(t)$ , which represents the units of domestic currency needed to purchase one unit of foreign currency (this definition is the inverse of the typically quoted foreign exchange rates, with the exception of the British pound).

Assume that the process for  $X(t)$  is of the form

$$\frac{dX(t)}{X(t)} = \mu_x dt + \sigma_x dW \quad (1.47)$$

where  $W$  is a multidimensional Brownian motion in the domestic risk-neutral measure and  $\sigma_x$  is a vector. To derive the process for the foreign exchange, we must take into account that a Brownian motion in the risk-neutral measure in the foreign market is not in general a Brownian motion in the risk-neutral measure in the domestic market. Consider the price of a foreign asset,  $S_f(t)$ , translated into the domestic currency  $X(t)S_f(t)$ . The drift of the process for this quantity must be the domestic risk-free rate. For an arbitrary asset  $S_f$ , however, we would introduce the Wiener process of  $S_f$ , whose risk-neutral drift in the domestic economy is not known. If we select the foreign money market account as the foreign asset, we don't introduce any additional Brownian motions and we get

$$\frac{d(X\beta_f)}{X\beta_f} = (\mu_x + r_f)dt + \sigma_x dW \quad (1.48)$$

The drift of the foreign exchange rate is obtained by requiring that the drift of the translated foreign money market account should be the risk-free rate. This gives us

$$\mu_x = r - r_f \quad (1.49)$$

This says that in the domestic measure, the foreign exchange rate behaves like an asset that pays a dividend yield equal to the instantaneous foreign risk-free rate.

As a second example, consider the relationship between Brownian motions in the foreign risk-free measure and Brownian motions

## 20 Introduction

in the domestic risk-free measure. Consider a foreign asset that in the foreign risk-free measure has the process

$$\frac{dS_f}{S_f} = r_f dt + \sigma_f dW_f \quad (1.50)$$

where  $W_f$  is a multidimensional Brownian motion in the foreign risk-neutral measure and  $\sigma_f$  is a vector. The process for the foreign asset translated into the domestic currency is

$$\frac{d(XS_f)}{XS_f} = (r - r_f) dt + r_f dt + \sigma_x dW + \sigma_f dW_f + \sigma_x \sigma_f dt \quad (1.51)$$

where  $\sigma_x \sigma_f dt$  stands for  $\text{cov}(dX/X, dS_f/S_f)$ . This expression will produce the correct domestic risk-neutral drift if the following relationship is satisfied:

$$\sigma_f dW = \sigma_f dW_f - \sigma_x \sigma_f dt \quad (1.52)$$

This is the fundamental relationship that connects the two Brownian motions.

As an example of how we can use this relationship, consider the *domestic* risk-neutral process of the yield to maturity of a *foreign* bond. The value of the foreign bond is

$$B_f(t) = e^{-y(T-t)} \quad (1.53)$$

where  $y$  is the yield, and  $T$  is the maturity. We assume that in the foreign risk-neutral measure, this bond has the following process:

$$\frac{dB_f}{B_f} = r_f dt + \sigma_B dW_f \quad (1.54)$$

A straightforward application of Ito's formula gives us the following process for the yield in the foreign risk-neutral measure:

$$dy = \frac{1}{T-t} \left( y - r_f + \frac{1}{2} \sigma_B \sigma_B \right) dt - \frac{1}{T-t} \sigma_B dW_f \quad (1.55)$$

We can now convert to the domestic risk-neutral measure by adding and subtracting  $\frac{1}{T-t} \sigma_B \sigma_x$ .

$$dy = \frac{1}{T-t} (y - r_f + \sigma_B \sigma_B - \sigma_B \sigma_x) dt - \frac{1}{T-t} \sigma_B dW \quad (1.56)$$

More examples are discussed in Chapter 2, when we consider the change of measure as a strategy for reducing the dimensionality of the pricing equation.

#### REFERENCES

- Black, F., & Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81, 637-659.
- Brennan, M.J., & Schwartz, B.S. (1977). Convertible Bonds: Valuation and Optimal Strategies for Call and Conversion. *Journal of Finance*, 32, 1699-1715.
- Dothan, M.D. (1990). *Process in Financial Markets*. Oxford, England: Oxford Financial Press.
- Duffie, D. (1996). *Dynamic Asset Pricing Theory*. Princeton, NJ: Princeton University Press.
- Durrett, R. (1996). *Stochastic Calculus*. Boca Raton, FL: CRC Press.
- Gardiner, E.W. (1994). *Handbook of Stochastic Methods* (2nd ed.). Berlin, Germany: Springer-Verlag.
- Gihman, I.F., & Skorohod, A.V. (1975). *The Theory of Stochastic Processes* (Vols. 1,2,3). Berlin: Springer-Verlag.
- Gillespie, D.T. (1992). *Markov Processes*. San Diego: Academic Press.
- Harrison, J.M., & Kreps, D. (1979). Martingales and Arbitrage in Multi-period Securities Markets. *Journal of Economic Theory*, 20, 381-408.
- Harrison, J.M., & Pliska, S. (1981). Martingales and Stochastic Integrals in the Theory of Continuous Trading. *Stochastic Processes and Their Applications*, 11, 215-260.
- Hull, J.E. (1997). *Options, Futures, and Other Derivatives* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

- Jaillet, P., Lamberton, D., & Lapeyre, B. (1990). Variational Inequalities and the Pricing of American Options. *Acta Applicandae Mathematicae*, 21, 263-289.
- Karatzas, I., & Shreve, S. (1988). *Brownian Motion and Stochastic Calculus*. New York: Springer-Verlag.
- Karatzas, I., & Shreve, S. (1998). *Methods of Mathematical Finance*. New York: Springer-Verlag.
- Kushner, H.J., & Dupuis, P.C. (1992). *Numerical Methods for Stochastic Control Problems in Continuous Time*. Berlin: Springer-Verlag.
- Merton, R.C. (1976). Option Pricing When Underlying Stock Returns Are Discontinuous. *Journal of Financial Economics*, 3, 125-144.
- Musiela, M., & Rutkowski, M. (1998). *Martingale Methods in Financial Modelling*. New York: Springer-Verlag.
- Van Kampen, N.C. (1992). *Stochastic Processes in Physics and Chemistry* (2nd ed.). Amsterdam, The Netherlands: Elsevier.
- Zvan, R, Forsyth, P.A., & Vetzal, K.R (1998). Robust Numerical Methods for PDE Models of Asian Options. *The Journal of Computational Finance*, 1, 39-78.

## *Chapter* 2

# The Pricing Equations

The discussion in this chapter focuses on the derivation of the fundamental partial differential equations (PDEs) for pricing European and American derivatives. Since this book is centered on the pricing of derivatives through the numerical solution of partial differential equations, one can reasonably ask under what conditions the price of a derivative is described by a partial differential equation. This is a complex issue that we do not elaborate to any significant extent. The basic condition that allows us to obtain a partial differential equation for the option price is the markovian nature of the underlying processes that determine the option's price. Difficulties arise if the underlying process is not markovian, that is, if the underlying process depends on the history of the Brownian motion that describes the process. A well-known example is the case of the short rate in the Heath-Jarrow-Morton framework (1992). That such a prominent example is problematic as far as the PDE methodology goes does not seem to be a significant concern. Some of the well-known models for short-rate processes are ways of getting around the problem. Such models are designed with features that make the short-rate process markovian (Bhar & Chiarella, 1995).

This chapter focuses on methodological aspects of the derivation of the pricing equation, and not on the detailed analysis of particular types of derivatives (interest rate options as opposed to equity options, for example). The PDE method can be viewed as a particularly robust approach, quite insensitive (within reasonable bounds) to the underlying processes. This is in stark opposition to analytical methods or trees, where the properties of the underlying process are crucial to the success of the method. For this reason, we will not devote separate sections to interest rates versus equity options. Any distinction between them will appear in the context of

implementational issues. This chapter does not provide a catalog of the pricing equations for a large variety of options. Instead, the emphasis is on presenting methodologies that readers can apply on their own. A few examples are used to illustrate the derivation of the pricing equations. The chapter starts with the derivation of the pricing equation for European options following two approaches. One approach is based on constructing a portfolio that replicates the cash flows of the derivative, the other approach uses the differential equation that is obeyed by conditional expectations: We then consider derivatives influenced by jumps and use credit derivatives as an illustrative example. We introduce the American option pricing problem as a dynamic optimization problem. This allows us to address the issue of free boundaries in an intuitive and straightforward fashion. Next, we discuss the linear complementarity framework for American option pricing. We then introduce path dependency and elaborate on discrete sampling as a practical implementational approach, noticing that in practice, it is continuous sampling that must be regarded as an approximation. We dedicate a detailed section to dimensionality reduction, using currency-linked options as an illustration. The chapter concludes with comments on the computation of hedging parameters.

## EUROPEAN DERIVATIVES

In the absence of jumps in the underlying processes, there are two fundamental approaches for deriving the pricing equations for European derivatives. The first approach consists in constructing a hedging portfolio whose value tracks the value of the derivative as a function of time. The second approach is based on the Feynman-Kac theorem, which states that the conditional expectation of a stochastic process obeys a partial differential equation. If there are jumps in the underlying processes, we need to introduce additional assumptions about the effect of the jumps. We devote a separate section to the derivation of the pricing equation in the presence of jumps.

### Hedging Portfolio Approach

Consider an option on an underlying process  $S(t)$ ,  $t \in [0, T]$ , with a payoff  $V(S(T), T) = F(S(T))$ . We want to obtain the equation describing

the value of the option at time  $t \in [0, T]$ ,  $V(S(t), t)$ . Assume that the underlying process is as follows, where  $\mu(S(t), t)$  and  $\sigma(S(t), t)$  are known given the information available at time  $t$ :

$$\frac{dS(t)}{S(t)} = \mu(S(t), t)dt + \sigma(S(t), t)dW(t) \quad (2.1)$$

For notational convenience, we will from now on omit the arguments in  $V$  and its derivatives,  $\mu$ ,  $\sigma$ , and  $W$  unless there is need for additional clarity. Applying Ito's lemma to  $V(S(t), t)$ , we get

$$dV(S(t), t) = \left( \frac{\partial V}{\partial t} + \mu S \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} \right) dt + \sigma S \frac{\partial V}{\partial S} dW \quad (2.2)$$

Assume now a hedging portfolio,  $P(t)$ , designed to track the evolution of the value of the derivative from time 0 to  $T$ . The portfolio will consist of  $n(S(t), t)$  units of  $S(t)$ , plus borrowing or lending. Over a small interval of time,  $\Delta t$ , the portfolio value will change for two reasons. The position in the underlying,  $nS$ , will cause capital gains or losses; and the difference between the value of the portfolio and the underlying position,  $P - nS$ , will accrue interest over the interval  $\Delta t$ . Denoting the instantaneous borrowing or lending rate by  $r(t)$ , the change in the replicating portfolio value is

$$dP = ndS + (P - nS)r\Delta t \quad (2.3)$$

where the first term on the right represents the capital gain or loss, and the second term represents the change in value due to either borrowing or lending. There is no term  $dndS$  in this equation because during the interval of time the portfolio composition remains unchanged. Substituting  $dS$  from Equation 2.1 into Equation 2.3, we find that the portfolio process must be:

$$dP = [Pr + nS(\mu - r)]dt + nS\sigma dW \quad (2.4)$$

For the portfolio to hedge the option,  $V(S(t), t)$  and  $P(t)$  must have the same drift and the same volatility. Equating the drifts in

## 26 The Pricing Equations

Equations 2.2 and 2.4, we find the following equation for the value of the option:

$$\frac{\partial V}{\partial t} + \mu S \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} = rP + n(\mu - r)S \quad (2.5)$$

Equating the volatilities, we find the number of units of  $S$  must satisfy the relationship:

$$n = \frac{\partial V}{\partial S} \quad (2.6)$$

Since we require that the hedging portfolio should replicate the option for any  $t \in [0, T]$ , we set  $P(t) = V(S(t), t)$  in Equation 2.5. Replacing for  $n$  in Equation 2.5 we get the following pricing equation:

$$\frac{\partial V}{\partial t} + rS \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} = rV \quad (2.7)$$

with the end condition

$$V(T, S(T)) = F(S(T)) \quad (2.8)$$

This is the Black-Scholes partial differential equation of option pricing. The drift  $\mu(S(t), t)$  of the underlying process does not appear in this expression.

Equation 2.7 applies if  $r$  is a deterministic function of time. If  $r$  itself were governed by a stochastic process, this should be reflected in the derivation and additional terms would appear. Notice also that if we set  $\mu = r$  and  $P = V$  in Equation 2.5, we obtain the Black and Scholes equation. However, Equation 2.5 only enforces the drifts of  $dV$  and  $dP$  to be the same, not their unanticipated changes. In other words, assuming that  $\mu = r$  has the same effect as requiring that the volatility component of the option price process and the volatility component of the replicating portfolio process should be the same. If  $r(t)$  is the instantaneous risk-free rate, a situation where we assume that the rate of growth of any asset is equal to  $r(t)$  is referred to as risk

neutral. As a result, the invocation of risk neutrality allows us, in the absence of cash flows associated with the underlying processes, to determine the pricing equation by simply enforcing the equality  $E_t[dV(S(t),t)] = r(t)V(S(t),t)$ , where  $dV(S(t),t)$  is the increment of  $V$  obtained through the application of Ito's lemma. Invoking risk neutrality means that the drifts of the underlying processes must be consistent with risk-neutral returns of traded assets. If the underlying process is the price of a stock that does not pay dividends, this consistency with risk neutrality simply means that  $E_t[dS(t)] = r(t)S(t)$ . In more complex cases, obtaining the appropriate risk-neutral drift may require significant elaboration.

### Feynman-Kac Approach

The Feynman-Kac theorem establishes a relationship between stochastic differential equations and partial differential equations. Given the stochastic differential equation

$$dy(t) = \mu(y(t), t)dt + \sigma(y(t), t)dW(t) \quad (2.9)$$

the Feynman-Kac theorem states that the expectation

$$f(y, t) = E_{y,t}[F(y(T))] \quad (2.10)$$

is the solution to the following partial differential equation:

$$\frac{\partial f}{\partial t} + \mu \frac{\partial f}{\partial y} + \frac{1}{2} \sigma^2 \frac{\partial^2 f}{\partial y^2} = 0 \quad (2.11)$$

subject to the end condition

$$f(y, T) = F(y) \quad (2.12)$$

In the case of several underlying processes  $y_1(t), y_2(t), \dots, y_n(t)$  following the stochastic differential equations

$$dy_i = \mu_i(y_1, y_2, \dots, y_n, t)dt + \sigma_i(y_1, y_2, \dots, y_n, t)dW_i \quad (2.13)$$

## 28 The Pricing Equations

the function

$$f(y_1, y_2, \dots, y_n, t) = E_{y_1, y_2, \dots, y_n, t} [F(y_1(T), \dots, y_n(T))] \quad (2.14)$$

is given by the solution of the differential equation:

$$\frac{\partial f}{\partial t} + \sum_{i=0}^{i=n} \mu_i \frac{\partial f}{\partial y_i} + \frac{1}{2} \sum_{i,j=0}^{i,j=n} \rho_{ij} \sigma_i \sigma_j \frac{\partial^2 f}{\partial y_i \partial y_j} = 0 \quad (2.15)$$

subject to

$$f(y_1, y_2, \dots, y_n, T) = F(y_1, y_2, \dots, y_n) \quad (2.16)$$

where  $\rho_{ij} = \text{cov}(dW_i, dW_j) / dt$ .

As an example of the Feynman-Kac approach, consider the derivation of the pricing equation for a claim in the case where the interest rate is described by the Hull and White model (1990). In the risk-neutral measure, the price of a claim on  $S(t)$  paying  $F(S(T))$  at maturity is given by

$$V(r(t), S(t), t) = E_{r, S, t} \left( F(T) e^{-\int_t^T r(s) ds} \right) \quad (2.17)$$

Assume that the processes for  $S(t)$  and  $r(t)$  are given by

$$dS = rSdt + \sigma_s S dW_s \quad (2.18)$$

and

$$dr = (\theta(t) - ar) dt + \sigma_r dW_r, \quad (2.19)$$

where  $\sigma_s$  and  $\sigma_r$  may be functions of the state variables  $r$  and  $S$  and time. Notice that the argument in the expectation in Equation 2.17 is not simply a function of  $T$ , but it depends on the trajectories of  $r$ . To apply the Feynman-Kac formula, we need to transform the

argument of the expectation so that it depends on the values of processes at time  $T$  only. We can accomplish this at the expense of temporarily increasing the dimensionality of the problem by defining an auxiliary process of the form

$$dz = -r(t)dt \quad (2.20)$$

With this, the value of the claim is (notice the conditions required of the expectation)

$$V(r, S, t) = e^{-z} E_{r, S, z, t}[F(T)e^{z(T)}] \quad (2.21)$$

We now define

$$V(r, S, t) = e^{-z} U(r, S, z, t) \quad (2.22)$$

and apply the three-dimensional Feynman-Kac formula to  $U(r, S, z, t)$

$$\begin{aligned} \frac{\partial U}{\partial t} + rS \frac{\partial U}{\partial S} + (\theta - ar) \frac{\partial U}{\partial r} - r \frac{\partial U}{\partial z} + \frac{1}{2} \sigma_s^2 \frac{\partial^2 U}{\partial S^2} \\ + S \sigma_s \sigma_r \rho_{S,r} \frac{\partial^2 U}{\partial S \partial r} + \frac{1}{2} \sigma_r^2 \frac{\partial^2 U}{\partial r^2} = 0 \end{aligned} \quad (2.23)$$

We can now eliminate the additional dimension,  $z$ , by replacing  $U(r, S, z, t) = e^z V(r, S, t)$  in Equation 2.24. This gives the two-dimensional pricing equation

$$\begin{aligned} \frac{\partial V}{\partial t} + rS \frac{\partial V}{\partial S} + (\theta - ar) \frac{\partial V}{\partial r} - rV + \frac{1}{2} \sigma_s^2 S^2 \frac{\partial^2 V}{\partial S^2} \\ + \rho_{S,r} r \sigma_s \sigma_r S \frac{\partial^2 V}{\partial S \partial r} + \frac{1}{2} \sigma_r^2 \frac{\partial^2 V}{\partial r^2} = 0 \end{aligned} \quad (2.24)$$

with the end condition

$$V(S, t, T) = F(S(T)), r \in [-\infty, +\infty], S \in [0, +\infty] \quad (2.25)$$

THE PRICING EQUATION IN THE  
PRESENCE OF JUMPS

Jump processes are frequently used to produce a more realistic description of the underlying processes, such as in the case of jump diffusion models for stock price movements (Merton, 1976). In other instances, the introduction of jumps is an essential component of a pricing framework, such as credit derivatives (Duffie & Singleton, 1999). In the next section, we will use credit puts to illustrate pricing with jumps (Schönbucher, 1997).

If one or more of the underlying process are discontinuous, the derivation of a pricing equation becomes somewhat more involved. As shown in the previous section, the argument that we can construct a hedging portfolio containing a given amount of the underlying security leads to the pricing equation and to the conclusion that the derivative should grow at the risk-free rate. If, instead of invoking the argument of a hedging portfolio, we take the expectation of the derivative process in a risk neutral world, we arrive at the same pricing equation. In the case of jumps, the pricing equation that we can get by invoking the hedging portfolio argument may depend on a criterion for the risk of the hedged portfolio. A common criterion is to state that the hedged portfolio should have minimum variance (Wilmott, 1998). This criterion leads to a particular pricing equation. You may justifiably ask, however, whether the equation obtained in this way is that which is followed by market prices of the derivative. An alternative approach to derive a pricing equation is to invoke the idea that changes in prices due to jumps are diversified away in the marketplace. This argument allows us to obtain a pricing equation by requiring that the risk-neutral expectation of the derivative price change should be proportional to the risk-free rate.

We first illustrate the derivation of a minimum variance pricing equation. Consider a single underlying process of the form,

$$dS = \hat{\mu}S(t^-)dt + \sigma S(t^-)dW + d\hat{J}s \quad (2.26)$$

as described in Chapter 1,

$$d\hat{J}s = S(t) - S(t^-) - (h(t) \int z n(S, z) dz) dt \quad (2.27)$$

where  $h$  is the jump probability per unit time and  $\hat{\eta}$  is the jump distribution density. We now assume a more particular definition of the jump by stating that if there is a jump the underlying price process is reduced to the fraction  $a$  of its value immediately before the jump. This is not a restriction in the derivation that follows. In this case,

$$d\hat{J}_s = S(t) - S(t^-) - (h(t)S(t^-) \int \alpha \hat{\eta}(S(t^-), \alpha) d\alpha) dt \quad (2.28)$$

where the “hat” on  $\eta$  distinguishes between the density of absolute jumps and the density of relative jumps. We can now define

$$d\hat{J}_s = S dJ_s \quad (2.29)$$

where

$$dJ_s = \frac{S(t) - S(t^-)}{S(t)} - (h(t) \int \alpha \hat{\eta}(S, \alpha) d\alpha) dt \quad (2.30)$$

We can now write the underlying process in the more familiar form

$$\frac{dS}{S} = \mu dt + \sigma dW + dJ_s \quad (2.31)$$

with

$$\mu = \hat{\mu} + h(t) \int \alpha \hat{\eta}(S(t^-), \alpha) d\alpha \quad (2.32)$$

As we did with the continuous case, assume that we can construct a portfolio  $P(t)$ , consisting of  $n(t)$  units of  $S(t)$ , which we will use to “hedge” our derivative with price process  $V(t)$ . The change in value of the portfolio is given by

$$dP = ndS + (P - nS) r dt \quad (2.33)$$

Replacing  $dS$  from Equation 2.31, we get

$$dP = (Pr + nS(\mu - r)) dt + nS\sigma dW + nSdJ_s \quad (2.34)$$

## 32 The Pricing Equations

As we saw in Chapter 1, the value of the derivative in the presence of jumps follows the process

$$dV = \left( \frac{\partial V}{\partial t} + \mu S \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + h \int [V(S+z) - V(S)] \eta(S, z) dz \right) dt + \frac{\partial V}{\partial S} \sigma S dW + dJ_V \quad (2.35)$$

The procedure that we used previously to obtain a pricing equation was to equate the drifts and volatilities of  $dV$  and  $dP$ , and then set  $V(t) = P(t)$ . This guaranteed that the replicating portfolio  $P(t)$  exactly tracked the derivative value  $V(t)$ . If we do the same now, we get the pricing equation

$$\frac{\partial V}{\partial t} + \mu S \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + h \int [V(S+z) - V(S)] \eta(S, z) dz = V_r + nS(\mu - r) \quad (2.36)$$

and the following two conditions

$$\frac{\partial V}{\partial S} \sigma S dW = nS \sigma dW \quad (2.37)$$

$$dJ_V = nS dJ_S \quad (2.38)$$

To derive a minimum variance pricing equation, we determine the value of  $n$  that minimizes the resulting variance from Equations 2.37 and 2.38. Denoting this variance by  $\sigma_T^2$ , we have

$$\sigma_T^2 dt = \text{Var} \left( \frac{\partial V}{\partial S} S \sigma dW - nS \sigma dW \right) + \text{Var} (dJ_V - nS dJ_S) \quad (2.39)$$

where we invoke that the jump processes are independent from the continuous process. Since  $dJ_V$  and  $dJ_S$  are martingales (see Chapter 1), the total variance is

$$\sigma_T^2 dt = \sigma^2 S^2 \left( \frac{\partial V}{\partial S} - n \right)^2 dt + E(dJ_V - n dJ_S)^2 \quad (2.40)$$

Setting the derivative of  $\sigma_T^2$  with respect to  $n$  equal to zero, we get the amount of  $S$  that minimizes the hedge variance. We use the notation

$$E_{VS} = \frac{E(dJ_V dJ_S)}{dt} \quad (2.41)$$

$$E_{SS} = \frac{E[(dJ_S)^2]}{dt} \quad (2.42)$$

$$n = \frac{\frac{\partial V}{\partial S} S^2 \sigma^2 + E_{VS}}{S^2 \sigma^2 + E_{SS}} \quad (2.43)$$

Replacing this expression in Equation 2.36, we get the following partial differential equation for pricing a derivative under a minimum variance hedge,

$$\begin{aligned} & \frac{\partial V}{\partial t} + \left[ \mu \left( 1 - \frac{S\sigma^2}{S\sigma^2 + E_{SS}} \right) + r \frac{S\sigma^2}{S\sigma^2 + E_{SS}} \right] S \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 s^2 \frac{\partial^2 V}{\partial S^2} \\ &= V_r - h \int [V(S+z) - V(S)] \eta(S, z) dz + \frac{E_{VS}}{S\sigma^2 + E_{SS}} S(\mu - r) \end{aligned} \quad (2.44)$$

Here we can make the following observations. In the absence of jumps, this equation becomes the standard Black and Scholes equation, as expected. The presence of jumps affects the convective term and adds to the source term on the right-hand side. The right-hand side now contains a convolution term. As a result, this pricing equation is an integra-partial differential equation. As explained in Chapter 4, the numerical solution of this equation does not pose a particularly difficult problem because the integral term on the right can be easily taken into account by iteration. This is the case even when the kernel of the convolution term is singular, as discussed in Chapter 4.

Another alternative to derive a pricing equation is to assume that the expectation of the derivative price change grows in proportion to

the risk-free rate. This assumption is consistent with the statement that jumps are diversified and do not impose a risk premium. The next subsection develops this approach in full detail and uses credit derivatives as a working example.

### An Application of Jump Processes: Credit Derivatives

The purpose of this section is to illustrate issues in the derivation of pricing equations that involve jumps. The reader interested in gaining a broader understanding of credit derivatives may consult some of the numerous articles in the literature (e.g., Duffie & Singleton, 1999; Schonbacher, 1997).

Credit derivatives are instruments sensitive either to changes in the credit quality of the issuer of an underlying security or to so-called credit events affecting the issuer of the underlying security. Among the credit events that affect the quality of the issuer are defaults and changes in credit rating. In principle, a corporate bond can be viewed as a credit derivative much the same way a nondefaultable bond can be viewed as an interest rate derivative. The market has agreed to call credit derivatives instruments specifically designed to manage credit risk, either by mitigating the risk or by gaining exposure to credit risk.

The two main approaches for pricing credit derivatives are structural methods, which attempt to establish relationships between the capital structure of the issuer of securities underlying the credit derivative; and reduced methods, where pricing is done by postulating models for the stochastic processes involved, without particular regard for capital structure considerations.

Although finite difference techniques can be used in the context of both of these approaches, we will concentrate on the applications to reduced models. We will illustrate the derivation of the pricing equations by considering a derivative that depends on a credit event such as default or credit change. We will assume that the credit event can be characterized as a pure jump process. If the credit event is a default, the value of the jump process at time  $t$  is an integer number that indicates the number of defaults that have occurred up to time  $t$ .

The process for the value of the derivative will be a function of the stochastic process that represents the credit event, as well as a function of any other relevant diffusion processes, such as the short rate.

In Chapter 1, we discussed the application of Ito's lemma to the case of processes with jumps. Consider the combined jump-diffusion process  $x$  (for additional details, see Chapter 1):

$$dx(t) = (\mu + h(x(t), t)) \int z \eta(x(t), z) dz dt + \sigma dW(t) + dJ_x \quad (2.45)$$

If we now set  $\mu = 0$  and  $\sigma = 0$ , we obtain a pure jump process with jumps at time  $t$  of magnitude  $z$ , distributed according to  $\eta(x(t), z)$ :

$$dx(t) = (h(x(t), t)) \int z \eta(x(t), z) dz dt + dJ_x(t) \quad (2.46)$$

If in addition, we assume that  $\eta(x(s), z) = \delta(z - 1)$ , we obtain a jump process that can jump by one at time  $t$ . We will assume that this is a useful representation of the process of credit events we are interested in. At this stage, we are not precisely specifying what the credit event is. The event intensity,  $h$ , can depend on both the event count,  $x(t)$ , and time. If the credit event were a default and we restricted our attention to the first event of default, the intensity would be only a function of time. The process for the credit event becomes

$$dx = h dt + dJ_x \quad (2.47)$$

Additionally, we may assume that the intensity of occurrence of the credit event results follows a diffusion process

$$dh = \mu_h dt + \sigma_h dW_h \quad (2.48)$$

where  $\mu_h$  and  $\sigma_h$  can be functions of time and the other continuous processes in the problem. To illustrate the derivation of the pricing equation for credit derivatives, let's also assume that the short rate is described by a diffusion process

$$dr = \mu_r dt + \sigma_r dW_r \quad (2.49)$$

where the drift and the volatility can be functions of time and the other continuous processes in the problem. Using Ito's lemma with

### 36 The Pricing Equations

jumps, the value of the derivative,  $V(t)$ , is governed by the following jump diffusion process:

$$dV(t) = \left( \frac{\partial V}{\partial t} + \mu_r \frac{\partial V}{\partial r} + \mu_h \frac{\partial V}{\partial h} + \frac{\sigma_r^2}{2} \frac{\partial^2 V}{\partial r^2} + \sigma_r \sigma_h \rho_{r,h} \frac{\partial^2 V}{\partial r \partial h} + \frac{\sigma_h^2}{2} \frac{\partial^2 V}{\partial h^2} \right) dt \\ + (h(x(t), t) \int [V(x(t) + z, r, t) - V(x(t), r, t)] \delta(z-1) dz) dt \quad (2.50) \\ + \frac{\partial V}{\partial r} \sigma_r dW_r + \frac{\partial V}{\partial h} \sigma_h dW_h + dJ_V$$

where the last term represents the random shocks undergone by the derivative's price as a result of the jumps in default process  $x$ . Carrying out the integral in Equation 2.51,

$$dV(t) = \left( \frac{\partial V}{\partial t} + \mu_r \frac{\partial V}{\partial r} + \mu_h \frac{\partial V}{\partial h} + \frac{\sigma_r^2}{2} \frac{\partial^2 V}{\partial r^2} + \sigma_r \sigma_h \rho_{r,h} \frac{\partial^2 V}{\partial r \partial h} + \frac{\sigma_h^2}{2} \frac{\partial^2 V}{\partial h^2} \right) dt \\ + (h(V(x(t) + 1, r, t) - V(x(t), r, t)) dt + \frac{\partial V}{\partial r} \sigma_r dW_r, \quad (2.51) \\ + \frac{\partial V}{\partial h} \sigma_h dW_h + dJ_V$$

Assume now that the market fully diversifies the shocks due to default. This means that the risk-neutral expectation of  $dV(t)$  must be equal to  $rV(t)$ . If we assume that the processes for  $h$ ,  $r$ , and  $x$  are in the risk-neutral measure, this expectation gives us the partial differential equation governing the derivative price. Taking the expectation of Equation 2.52

$$\frac{\partial V}{\partial t} + \mu_r \frac{\partial V}{\partial r} + \mu_h \frac{\partial V}{\partial h} + \frac{\sigma_r^2}{2} \frac{\partial^2 V}{\partial r^2} + \sigma_r \sigma_h \rho_{r,h} \frac{\partial^2 V}{\partial r \partial h} + \frac{\sigma_h^2}{2} \frac{\partial^2 V}{\partial h^2} \quad (2.52) \\ = V(x(t)) r - h(V(x(t) + 1) - V(x(t)))$$

where we used the fact that  $W_r$ ,  $W_h$ , and  $J_V$  are martingales.

For clarity of the discussion, assume now that the credit event characterized by  $x(t)$  is the default process. Notice that  $x$  does not contribute to the dimensionality of the problem because it does not

appear on the left-hand side of the pricing equation. It is convenient to rewrite the pricing equation as follows:

$$\frac{\partial V}{\partial t} + \mu_r \frac{\partial V}{\partial r} + \mu_h \frac{\partial V}{\partial h} + \frac{\sigma_r^2}{2} \frac{\partial^2 V}{\partial r^2} + \sigma_r \sigma_h \rho_{r,h} \frac{\partial^2 V}{\partial r \partial h} + \frac{\sigma_h^2}{2} \frac{\partial^2 V}{\partial h^2} = Vr - h\delta V \quad (2.53)$$

where  $\delta V$  is the known change in value of the derivative if default occurs. We next discuss two simple examples of application of the pricing equation.

### Defaultable Bonds

In the event of default of the issuer of a bond, the bond drops in value to a level called the recovery value of the bond. The way the recovery of the bond is characterized can significantly influence the pricing of the bond. Assume that upon default, the holder of the bond receives a given fraction,  $R(t)$ , of the contemporaneous market value of the bond. In that case, the right-hand side of Equation 2.53 can be written as follows:

$$Vr - h\delta V = Vr - h(RV - V) = Vr + hLV = V(r + Lh) \quad (2.54)$$

where  $L$  is the loss fraction in the event of default. Of course,  $R(t)$  can be a stochastic process. This assumption about recovery is known as recovery of market value (Duffie & Singleton, 1999). The pricing equation is now

$$\frac{\partial V}{\partial t} + \mu_r \frac{\partial V}{\partial r} + \mu_h \frac{\partial V}{\partial h} + \frac{\sigma_r^2}{2} \frac{\partial^2 V}{\partial r^2} + \sigma_r \sigma_h \rho_{r,h} \frac{\partial^2 V}{\partial r \partial h} + \frac{\sigma_h^2}{2} \frac{\partial^2 V}{\partial h^2} = V(r + Lh) \quad (2.55)$$

In addition, we need an end condition at maturity, and boundary condition for  $r = 0$ ,  $r = \infty$ ,  $h = 0$ ,  $h = \infty$ . If we view the bond as a contract that terminates on default, the end condition will be equal to a known payment equal to the notional amount. Although the boundary conditions are discussed in greater detail in Chapter 4, here we can make an observation that brings up the significance of assuming the recovery to be a given fraction of market value as opposed to a given amount. If we ask ourselves, what is the value of

the bond as  $h \rightarrow \infty$ , it appears intuitively clear that if the recovery is equal to a given amount to be paid in case of default, the value of the bond should be precisely this amount. The reason for this is that if the intensity of default is infinitely large, the bond will default immediately and the holder will receive the known amount immediately. If, on the other hand, recovery is given in terms of market value, the pricing equation contains a source term of the form  $LVh$ , which cannot be balanced by the finite terms on the left-hand side if  $V \neq 0$ . The only way for the equation to make sense as  $h \rightarrow \infty$  (assuming  $V$  remains smooth) is for  $V \rightarrow 0$  as  $h \rightarrow \infty$ .

The suitable boundary condition for  $h = 0$  is the solution to the PDE describing a default-free bond.

### Full Protection Credit Put

A credit put compensates the holder for either the loss of value of defaultable bonds as a result of default, or as a result of credit degradation, or both. Here we formulate a simple example as follows. Consider a derivative that pays the holder the following amount if default occurs at time  $t$ , where  $0 < t < T$

$$KB(t, T_B) - R(t)B_d(t, T_D) \quad (2.56)$$

where  $R(t)$  is the recovery rate,  $B(t, T_B)$  is a risk-free bond maturing at time  $T_B$ ,  $B_d(t, T_D)$  is a defaultable bond maturing at time  $T_D$ , and  $K$  is a constant. We will assume that the occurrence of a default is enough to trigger the payment. In such a case, the contract terminates. If no payments triggered by default have occurred until maturity, at maturity the derivative pays the following:

$$\max[0, KB(T, T_B) - B_d(T, T_D)] \quad (2.57)$$

This is a full-protection credit put because it protects both against defaults and drop of value as a result of increased yields.

The right-hand side of Equation 2.53 in this case is

$$\begin{aligned} Vr - h\delta V &= Vr - h(KB(t, T_B) - R(t)B_d(t, T_D) - V) \\ &= V(r + h) - h(KB(t, T_B) - R(t)B(t, T_D)) \end{aligned} \quad (2.58)$$

The pricing equation of the credit put can be written as

$$\begin{aligned} \frac{\partial V}{\partial t} + \mu_r \frac{\partial V}{\partial r} + \mu_h \frac{\partial V}{\partial h} + \frac{\sigma_r^2}{2} \frac{\partial^2 V}{\partial r^2} + \sigma_r \sigma_h \rho_{r,h} \frac{\partial^2 V}{\partial r \partial h} + \frac{\sigma_h^2}{2} \frac{\partial^2 V}{\partial h^2} \\ = V(r+h) - h[KB(t, T_B) - B_d(t, T_D)] \end{aligned} \quad (2.59)$$

with the end condition

$$V(T) = \max[0, KB(T, T_B) - B_d(T, T_D)] \quad (2.60)$$

and suitable boundary conditions. Boundary conditions for the credit put are fully described in the implementation example discussed in Chapter 5.

To solve Equation 2.59, we need to know  $B(t, T_B)$  and  $B_d(t, T_D)$  at every point in the solution space. This means that in addition to this PDE, we need to solve the two additional ones representing the values of the riskless and defaultable bonds. For the riskless bond, we have

$$\frac{\partial B}{\partial t} + \mu_r \frac{\partial B}{\partial r} + \frac{\sigma_r^2}{2} \frac{\partial^2 B}{\partial r^2} = Br \quad (2.61)$$

and for the defaultable bond,

$$\begin{aligned} \frac{\partial B_d}{\partial t} + \mu_r \frac{\partial B_d}{\partial r} + \mu_h \frac{\partial B_d}{\partial h} + \frac{\sigma_r^2}{2} \frac{\partial^2 B_d}{\partial r^2} + \sigma_r \sigma_h \rho_{r,h} \frac{\partial^2 B_d}{\partial r \partial h} \\ + \frac{\sigma_h^2}{2} \frac{\partial^2 B_d}{\partial h^2} = B_d(r + Lh) \end{aligned} \quad (2.62)$$

In addition, we need suitable boundary conditions. These are discussed fully in Chapter 5. The pricing problem is then the solution of a system of the three partial differential Equations 2.59, 2.61, and 2.62. Although this task may appear daunting, it is not particularly difficult, as shown in Chapter 6.

## AMERICAN DERIVATIVES

An American style derivative is a contract whose cash flows can be influenced by the holder. The holder affects the cash flows of the

contract through an exercise strategy. In its simplest form, the exercise strategy may simply consist of the decision to exercise or not exercise at any given time between the inception of the contract and its maturity date. In general, however, the exercise policy can consist of complex rules stipulated in the contract. The market has evolved specialized names for specific types of exercise policies. Here, however, we call American derivatives any derivative whose cash flows can be affected by the holder in a nontrivial manner (the trivial manner to affect the cash flows is to sell the contract) ..

### Relationship between European and American Derivatives

For simplicity of exposition, we will consider the case of a single underlying process,  $S(t)$ . We denote the set of exercise strategies by  $c(\cdot)$ , whose arguments may include items such as the underlying process, properties of the past history of the underlying processes, and time. In the simple case of an American put,  $c(S,t)$  is a binary variable representing the decision to exercise or not.

If markets are complete, there is a straightforward conceptual relationship between European and American derivatives. A market is complete if the cash flows generated by any security can be replicated through a dynamic trading strategy with other securities. Fixing an exercise strategy means that the holder of the security will not be able to influence the cash flows. This means that for every exercise strategy of an American security there is a corresponding European security. Consider a derivative whose price,  $\sim$  depends on an exercise strategy,  $c$ . Consider now a particular exercise strategy  $\hat{c}$ , such that

$$V(\hat{c}) = \max_c V(c) \quad (2.63)$$

The exercise strategy  $\hat{c}$  is called an optimal exercise strategy. It is easy to see that to prevent arbitrage, the value of the American security must equal  $V(\hat{c})$ . Assume that the American security's price is less than  $V(\hat{c})$ . In this case, we sell the portfolio that replicates  $V(\hat{c})$  and we purchase the American derivative. We now follow an optimal exercise strategy, thereby matching the cash flows between the American security and  $V(\hat{c})$ , and keep the initial risk-free profit.

If, on the other hand, the price of the American security were greater than  $V(\hat{c})$ , we could buy the portfolio that replicates  $V(\hat{c})$  and sell the American derivative. The holder of the American security would follow an optimal exercise strategy, thus causing the cash flows between the American security and  $V(\hat{c})$  to be matched. Again, we keep the initial risk-free profit. Because we assume that markets are complete, there is no problem in replicating  $V(\hat{c})$ .

We see that the valuation of American options is no different than the valuation of European options, *provided we know the optimal exercise strategy  $\hat{c}$* . Of course, we don't know ahead of time what the optimal exercise strategy is. In practice, the optimal exercise strategy is found simultaneously with the price.

In derivatives pricing, optimal exercise strategies are associated with the concept of free boundaries. For illustration, consider the simple *exercise-don't exercise* version of exercise strategy. The free boundary separates the region in  $(S,t)$  where it is optimal to exercise the option from the region where it is optimal to hold the option. The free boundary is also referred to as the exercise boundary. The argument in the previous paragraph indicates that the price of the American security will have the same description as a European security on the side of the free boundary where it is optimal to hold the security, and it will be equal to the exercise value on the side where it is optimal to exercise. Since the European pricing problem is described by a partial differential equation, the presence of the exercise boundary poses the question of suitable conditions at those boundaries. In the approach followed in this book, it is not necessary to be concerned about the details of what happens at the exercise boundaries. The reason is that in the linear complementarity implementation of finite differences, the exercise boundary is resolved as part of the solution. To better understand the behavior of the solution, however, the next two sections discuss the conditions that must be satisfied at the exercise boundaries. It is possible to pose the problem such that the exercise boundary is also a boundary of the computational domain. This is easy to do if the problem has one space variable. The resolution of the boundary as an integral part of the solution, however, is a much more practical approach, especially in several dimensions.

For the purpose of clarifying the conditions at the exercise boundaries, the next two sections address the American pricing problem in the context of dynamic optimization.

## American Options as Dynamic Optimization Problems

This discussion is not intended as a real solution strategy, but as a device to derive the continuity conditions at the exercise boundaries. For the purpose of illustration, consider an American derivative with an *exercise-don't exercise* strategy that applies in the interval  $0 \leq t \leq T$ . Here, at every time  $t$  the decision must be made whether to exercise the option or to continue to hold it. Direct optimization of the option value over the space of exercise strategies will in general lead to a problem with a very large number of dimensions, since the number of possible exercise strategies is very large. We can, in principle, "parameterize" the problem by a suitable description of the free boundaries and solve an optimization problem over a reduced number of dimensions. Although this is often done in practice (Ingersoll, 1998), it requires a priori knowledge of the features of the free boundaries.

When the optimal exercise strategy depends only on the current value of the underlying processes and time, there is a powerful alternative for determining the optimal strategy and thereby the option value, known as the Bellman principle of dynamic programming (Dixit & Pindyck, 1994).

The Bellman principle leads to a recursive argument that states the optimal strategy in terms of two components. In the case of an American option with an *exercise-don't exercise* strategy, the Bellman principle in continuous time can be phrased as follows: *At a given time, the optimal strategy corresponds to the maximum of either the exercise value, or the value associated with selecting an optimal strategy an instant later.* This idea can be expressed in the Bellman equation of dynamic programming. For simplicity, consider only one underlying price process,  $S(t)$ , and an exercise value,  $F(S(t))$  that depends only on  $S(t)$ . Notice that we assume the exercise value itself does not depend on the exercise strategy. The Bellman equation is

$$V(S(t)) = \max\{F(S(t)), PV_t[V(S(t) + dS(t), t + dt)]\} \quad (2.64)$$

where  $PV_t$  stands for present value at time  $t$ .

Notice that the recursive structure of Equation 2.64 allows us to solve for both the optimal strategy and the value of the security if we know end conditions and work backward in time. *The commonly*

used backward induction techniques implemented through lattices for American option pricing are particular implementations of solutions of the Bellman equation. We will use the Bellman equation to discuss the boundary conditions at exercise boundaries.

## Conditions at Exercise Boundaries

The purpose of this section is to better understand the way the solution behaves near exercise boundaries. In practice, we don't need to concern ourselves with the properties of the exercise boundary, since the exercise boundary is automatically captured by the linear complementarity formulation. Two conditions must be satisfied at exercise boundaries. The first condition is that the exercise value and the continuation value of the option must be the same at the exercise boundary. This is a way of characterizing the exercise boundary as a region of indifference between exercising and not exercising the option.

The second condition is that the gradient of the option value with respect to the underlying variables must be continuous at the exercise boundary. This condition is known as *smooth pasting*, and it states that the gradient of the exercise value of the option must be equal to the gradient of the continuation value.

To prove the condition of smooth pasting, consider the implications of Equation 2.64 when we are at the exercise boundary. If we are at the exercise boundary, we must have  $V(S,t) = F(S)$ . Assume that an upward movement in  $S$  would place us in the exercise region, while a downward movement would place us in the continuation region (this only requires continuity of  $V(S,t)$  at the exercise boundary). If there is an upward movement in  $S$ , the option payoff will be

$$F(S + dS) \quad (2.65)$$

If there is a downward movement, the option value will be

$$V(S + dS, t + dt) + E_{t+dt} dV \quad (2.66)$$

Assume now that the probability of an upward movement is equal to  $p$ . The option value can be written as follows:

## 44 The Pricing Equations

$$V(S, t) = \text{PV}[F(S + dS)p + (V(S + dS, t + dt) + E_{t+dt} dV)(1 - p)] \quad (2.67)$$

where  $\text{PV}(\cdot)$  denotes present value. Since at the exercise boundary we are indifferent between exercising and waiting,  $V(S, t)$  must equal the discounted expectation of  $V(S + dS, t + dt)$ . Replacing this in Equation 2.67, we get:

$$\begin{aligned} \text{PV}[V(S + dS, t + dt) - F(S + dS)p - (V(S + dS, t + dt)(1 - p) \\ - E_{t+dt} dV)(1 - p)] = 0 \end{aligned} \quad (2.68)$$

Expanding in Taylor series and replacing  $F = V$ ,

$$\text{PV}\left[\left(\frac{\partial F}{\partial S}dS - \frac{\partial V}{\partial S}dS - \frac{\partial V}{\partial t}dt\right)p + (E_{t+dt} dV)(1 - p)\right] = 0 \quad (2.69)$$

If the underlying process is a diffusion process of the form  $dS = \mu dt + \sigma dW$ , then infinitesimal changes in  $S$  are proportional to  $\sqrt{dt}$ . In addition, the probability of upward or downward movements deviates from one also proportionally to  $\sqrt{dt}$ . This means that Equation 2.69 can be written as follows:

$$\text{PV}\left[\left(\frac{\partial F}{\partial S}O(\sqrt{dt}) - \frac{\partial V}{\partial S}O(\sqrt{dt}) - \frac{\partial V}{\partial t}dt\right)p + (E_{t+dt} dV)O(\sqrt{dt})\right] = 0 \quad (2.70)$$

Neglecting higher order terms, we get the following relationship that must be satisfied at the exercise boundary:

$$\frac{\partial V}{\partial S} = \frac{\partial F}{\partial S} \quad (2.71)$$

This condition is known as “smooth pasting.”

## Linear Complementarity Formulation of American Option Pricing

This section derives the valuation strategy for American options that we follow throughout the book. The value at time  $t$  of an option that can be exercised at times  $\tau$  is given by the following expectation:

$$V(r, S, t) = \sup_{\tau} E_{r, S, t} \left[ e^{-\int_t^{\tau} r(s) ds} f(S(\tau)) \right] \quad (2.72)$$

where  $\tau$  are stopping times conditional on information at time  $t$ , and  $f(\cdot)$  is the payoff if the option is exercised at time equal to the stopping time  $\tau$  (Lamberton & Lapeyre, 1996).

It can be shown that Equation 2.72 will hold if the following system of partial differential inequalities is satisfied (Lamberton & Lapeyre, 1996).

$$\begin{aligned} V &\geq f \\ \frac{\partial V}{\partial t} + rS \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} &\leq rV \\ \left( \frac{\partial V}{\partial t} + rS \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV \right) (V - f) &= 0 \\ V(T, S) &= f(S) \end{aligned} \quad (2.73)$$

Intuitively, this system can be understood in the following manner. The first inequality expresses the fact that at all times the value of the option cannot fall below its intrinsic value. The second inequality reflects that if the value of the option grows more slowly than that of a riskless bond, the option is exercised. The third condition enforces the fact that if the value of the option is above its intrinsic value, the price of the option is described by the same partial differential equation that describes the corresponding European option. The last condition indicates that the option value at maturity equals its intrinsic value. In Chapter 3, we describe in detail a standard method for solving system (2.73).

## PATH DEPENDENCY

In this section, path dependency refers to the dependency of the payoff of the derivative on the trajectory followed by one or more of the underlying processes. Another notion of path dependency refers to the underlying processes themselves being a function of the trajectory. There is a distinction between these two notions of path dependency. If the underlying processes themselves are path

dependent, we may not be able to obtain, without further assumptions (Bhar & Chiarella, 1995), a pricing equation in the form of a partial differential equation. Our case of interest is when the value of the derivative depends on the trajectory but the underlying processes are markovian.

Path dependency is a concern in the context of numerical solutions for two primary reasons. First, if the cash flows of the derivative depend on some function of the trajectory of the underlying processes, this dependency will result in a larger number of dimensions in the pricing equation. Second, it may cause the resulting pricing equation to be significantly more difficult to solve because of the absence of diffusion in the additional dimension. To illustrate the second point, consider a simple case of a single under lying process,  $S(t)$ , where the derivative price depends on the time integral of some function of the underlying process,  $g(t)$ , defined as follows:

$$g(t) = \int_0^t f(S(t)) dt \quad (2.74)$$

The derivative price will be a function of  $S$ ,  $g$ , and  $t$ . Assume that  $S$  follows the log-normal process  $dS = S\mu dt + S\sigma dW$ . The application of Ito's lemma to  $g(t)$  gives us

$$dg(t) = f(S(t)) dt \quad (2.75)$$

This means that the increment of  $g$  at time  $t$  is known if the information set at time  $t$  is known. The fact that  $g$  was given by an integral of a function of the underlying process means that  $dg$  does not have a diffusion component. We can now apply Ito's lemma to  $dV(S,g,t)$  and use the procedures described earlier to derive the pricing equation:

$$\frac{\partial V}{\partial t} + rS \frac{\partial V}{\partial S} + f(S) \frac{\partial V}{\partial g} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV = 0 \quad (2.76)$$

We can make two observations about the effect of the  $g$  dimension. The most obvious one is that there is no corresponding diffusion in the  $g$  dimension. This alone can be a reason for concern,

since, as shown in Chapter 3, it is the diffusion term that contributes to the stability of the numerical schemes used to solve the pricing equation. A more subtle observation, however, is that the coefficient in front of  $\frac{CIV}{Clg}$  can in principle be of a very different magnitude than the other coefficients in the equation, depending on the definition of  $f(5)$ . This may also significantly add to the numerical difficulty (Zvan, Forsyth, & Vetzal, 1998).

In practice, the additional convective term turns out not to be a significant difficulty if one considers the case of discrete sampling. Discrete sampling is the main approach in this book. The interested reader will find Zvan et al. (1998) is a good example of work on the development of robust algorithms for the continuous case. An alternative approach to getting around the problem of absent diffusion in a particular coordinate is to introduce artificial diffusion in that direction and then obtain the limit as the artificial diffusion vanishes. This can be done numerically quite efficiently.

### Discrete Sampling of Path Dependency

Discrete sampling of path dependency is a better approximation to what really happens, since the movement of the underlying processes can only be observed at discrete points in time. Discrete sampling, however, can also be viewed as a means of dealing with the problem introduced by the additional convective term and the absence of corresponding diffusion. The key observation is that the value of the option immediately before the sampling time and immediately after the sampling time must be the same. This will be the case as long as sampling itself does not trigger cash flows. Denoting the sampling times by  $t_i$ ,  $i = 1, 2, \dots$ , and the times immediately preceding the sampling time by  $t_i^-$ ,  $i = 1, 2, \dots$ , this continuity condition is expressed as follows:

$$V(S(t_i), g(t_i), t_i) = V(S(t_i), g(t_i^-), t_i^-), i = 1, 2, \dots \quad (2.77)$$

The pricing equation between sampling times is obtained by applying Ito's lemma to the relevant underlying processes. Since between sampling times  $g(t)$  is constant, it does not appear in the pricing equation. Between sampling times, we must solve

$$\frac{\partial V}{\partial t} + rS \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} = rV \quad (2.78)$$

where the term  $f(S) \frac{\partial V}{\partial g}$  does not enter. This equation must be solved within each sampling interval subject to initial conditions derived from the preceding continuity condition.

The initial condition at the beginning of each sampling interval must be extracted from the solution at the end of the previous sampling interval such that the continuity conditions are satisfied. Notice that although the  $g$  dimension has dropped from the PDE, the solution itself still has the  $g$  dimension. What has happened is that the changes in the  $g$  dimension, which in the continuous sampling case occur through the differential equation, here occur through the initial conditions in each sampling interval. The application of the continuity condition given by Equation 2.77 has the effect of concentrating the convection at the boundaries of sampling intervals.

The practical implementation of the continuity condition is discussed in greater detail in Chapter 4. The continuity condition is also referred to as a jump condition in the literature (Wilmott, DeWynne, & Howison, 1993).

### DIMENSIONALITY REDUCTION

Although the previous section shows the general form of the pricing equations, without further manipulation these equations may be more complex than they need to be. It is often possible to reduce the number of dimensions involved by invoking a change of measure or self-similarity argument. It is important to realize that a reduction in the number of dimensions can contribute greatly to the efficiency of the finite difference implementation. This procedure amounts to what is called a change of numeraire. The possibility of making this transformation is suggested by the payoff function. The next section summarizes the reformulation of a process in the measure induced by another process. We then show a detailed application of this technique to the derivation of one-dimensional pricing equations for options on FX-linked processes.

A particularly fine example of change of numeraire has been contributed in pricing Asian options (Andreasen, 1998).

### Reformulating the Underlying Processes in a Different Measure

A powerful tool for exploring this issue is the reformulation of the problem in a different measure (see discussion of this concept in Chapter 1). Here, we implement this concept in a straightforward fashion with the objective of eliminating a dimension in the payoff function of a European derivative. Consider two assets,  $A(t)$  and  $B(t)$ , with constant dividend yields  $qa$  and  $qb$  respectively, and whose risk-neutral processes are given by

$$\frac{dA}{A} = (r - q_a)dt + \sigma_a dW \quad (2.79)$$

$$\frac{dB}{B} = (r - q_b)dt + \sigma_b dW \quad (2.80)$$

where  $W$  is multidimensional Brownian motion under the risk-neutral measure,  $Q$  and  $\sigma_a$ ,  $\sigma_b$  are vectors. Application of Ito's lemma leads to

$$\frac{d\frac{A}{B}}{\frac{A}{B}} = (q_b - q_a - (\sigma_a - \sigma_b) \sigma_b)dt + (\sigma_a - \sigma_b)dW \quad (2.81)$$

The process

$$dW^* = dW - \sigma_b dt \quad (2.82)$$

is a Brownian motion in a particular measure,  $Q^*$ , which we will refer to as the measure induced by or associated with asset  $B(t)$ . In this measure, the process for  $\frac{A}{B}$  becomes

$$d\frac{A}{B} = \frac{A}{B} (q_b - q_a)dt + \frac{A}{B} (\sigma_a - \sigma_b)dW \quad (2.83)$$

## 50 The Pricing Equations

Since we have assumed that  $q_a$  and  $q_b$  are constant, we can write

$$A(0) = B(0) \mathbb{E}^{Q^*} \left[ e^{(q_a - q_b)t} \frac{A(t)}{B(t)} \right] \quad (2.84)$$

where  $\mathbb{E}^{Q^*}$  indicates expectation under  $Q^*$ . This tells us that when we change from the risk-neutral measure, where the discounting is at the instantaneous risk-free rate minus the dividend yield, to the measure associated with the process  $B(t)$ , the discounting of  $\frac{A}{B}$  becomes the difference between the two dividend yields. Another way of saying this is that under the measure associated with  $B(t)$ , the discounting of  $\frac{A}{B}$  is the difference between their risk-neutral drifts.

Under the measure  $Q^*$ , the processes for  $A(t)$  and  $B(t)$  become

$$\frac{dA}{A} = (r - q_a + \sigma_a \sigma_b) dt + \sigma_a dW^* \quad (2.85)$$

$$\frac{dB}{B} = (r - q_b + \sigma_b \sigma_b) dt + \sigma_b dW^* \quad (2.86)$$

As an example, we will now see how this notion can be applied to the case of currency translated options. This technique has been used successfully in other types of options as well. The interested reader may consult Andreason (1998) for an example of the potential offered by this approach in pricing path-dependent options.

### Currency Translated Options

Currency translated options involve payoffs in a currency other than that of the reference asset, or payoffs that are proportional to the value of an asset denominated in a currency other than that of the settlement. European currency translated options were studied in depth in Reiner (1992). Currency translated options can be classified in four types, (Shaw, 1998) whose payoffs are characterized as follows:

$$\text{Type 1: Payoff in domestic currency} = XTF(ST) \quad (2.87)$$

$$\text{Type 2: Payoff in domestic currency} = F(X_T ST) \quad (2.88)$$

$$\text{Type 3: Payoff in domestic currency} = X_0 F(S_T) \quad (2.89)$$

$$\text{Type 4: Payoff in domestic currency} = S_T F(X_T) \quad (2.90)$$

where  $X$  is the foreign exchange rate in number of domestic currency units per foreign currency unit,  $S$  is the foreign asset, valued in foreign currency, and  $F$  denotes the payoff function.

The domestic risk-neutral processes for  $S$  and  $X$  (see Chapter 1) are

$$\frac{dS}{S} = (r - \rho \sigma_x \sigma_s) dt + \sigma_s dW_s \quad (2.91)$$

and

$$\frac{dX}{X} = (r_f - r) dt + \sigma_x dW_x \quad (2.92)$$

where  $r$  and  $r_f$  are the domestic and foreign short interest rates, respectively. We assume that  $W_s$  and  $W_x$  are one-dimensional Brownian motions in the domestic risk-neutral measure and  $\rho$  is the correlation coefficient between  $dW_s$  and  $dW_x$ .

The pricing equation is

$$\begin{aligned} \frac{\partial V}{\partial t} + (r - r_f) X \frac{\partial V}{\partial X} + (r_f - \rho \sigma_x \sigma_s) S \frac{\partial V}{\partial S} + \frac{1}{2} \sigma_x^2 X^2 \frac{\partial^2 V}{\partial X^2} \\ + \rho \sigma_x \sigma_s X S \frac{\partial^2 V}{\partial X \partial S} + \frac{1}{2} \sigma_s^2 S^2 \frac{\partial^2 V}{\partial S^2} = rV \end{aligned} \quad (2.93)$$

We could solve this two-dimensional equation, together with initial conditions given by the payoff function and suitable boundary conditions. However, by applying a simple measure transformation, we can reduce the pricing of all four types of currency translated options to one-dimensional problems.

**Type 1.** In this case, we would like to get rid of  $X(T)$  in the payoff function. According to the results in the previous section, the following expression holds, where  $E\sim$  indicates expectation in the

## 52 The Pricing Equations

measure induced by  $X(t)$  conditional on information available at time 0.

$$\frac{V(0)}{X(0)} = E_0^x \left[ e^{-(\mu_v - \mu_x)T} \frac{V(T)}{X(T)} \right] \quad (2.94)$$

Where  $\mu_v$  and  $\mu_x$  are the risk-neutral drifts of  $V(t)$  and  $X(t)$ , respectively. Since  $V(t)$  is the value of an asset that does not pay dividends,  $\mu_v = r$ . We also know that  $\mu_x = r - r_f$  hence

$$\begin{aligned} \frac{V(0)}{X(0)} &= E_0^x \left[ e^{-r_f T} \frac{V(T)}{X(T)} \right] \\ &= E_0^x \left[ e^{-r_f T} \frac{X(T)F(S(T))}{X(T)} \right] \\ &= E_0^x [e^{-r_f T} F(S(T))] \end{aligned} \quad (2.95)$$

The value of the option at time 0 is then,

$$V(0) = X(0) E_x [e^{-r_f T} F(S(T))] \quad (2.96)$$

We can now apply the Feynman-Kac theorem to the expectation on the right. To do this, the process for  $S(t)$  must be expressed in the measure induced by  $X(t)$ . We start out with the risk-neutral process in the foreign measure

$$dS = S r_f dt + S \sigma_s dW_f \quad (2.97)$$

where  $W_f$  is a Brownian motion in the foreign measure. Next, we express the process in the domestic risk-neutral measure,

$$dS = S(r_f - \rho \sigma_x \sigma_s) dt + S \sigma_s dW \quad (2.98)$$

where  $W$  is a Brownian motion in the domestic measure. Finally, we convert this process to the measure induced by  $X(t)$ ,

## 52 The Pricing Equations

measure induced by  $X(t)$  conditional on information available at time 0.

$$\frac{V(0)}{X(0)} = E_0^x \left[ e^{-(\mu_v - \mu_x)T} \frac{V(T)}{X(T)} \right] \quad (2.94)$$

Where  $\mu_v$  and  $\mu_x$  are the risk-neutral drifts of  $V(t)$  and  $X(t)$ , respectively. Since  $V(t)$  is the value of an asset that does not pay dividends,  $\mu_v = r$ . We also know that  $\mu_x = r - r_f$  hence

$$\begin{aligned} \frac{V(0)}{X(0)} &= E_0^x \left[ e^{-r_f T} \frac{V(T)}{X(T)} \right] \\ &= E_0^x \left[ e^{-r_f T} \frac{X(T)F(S(T))}{X(T)} \right] \\ &= E_0^x [e^{-r_f T} F(S(T))] \end{aligned} \quad (2.95)$$

The value of the option at time 0 is then,

$$V(0) = X(0) E_x [e^{-r_f T} F(S(T))] \quad (2.96)$$

We can now apply the Feynman-Kac theorem to the expectation on the right. To do this, the process for  $S(t)$  must be expressed in the measure induced by  $X(t)$ . We start out with the risk-neutral process in the foreign measure

$$dS = S r_f dt + S \sigma_s dW_f \quad (2.97)$$

where  $W_f$  is a Brownian motion in the foreign measure. Next, we express the process in the domestic risk-neutral measure,

$$dS = S (r_f - \rho \sigma_x \sigma_s) dt + S \sigma_s dW \quad (2.98)$$

where  $W$  is a Brownian motion in the domestic measure. Finally, we convert this process to the measure induced by  $X(t)$ ,

$$dS = S(r_f - \rho\sigma_x\sigma_s + \rho\sigma_x\sigma_s) dt + S\sigma_s dW_x \quad (2.99)$$

equivalently,

$$dS = S r_f dt + S \sigma_s dW_x \quad (2.100)$$

We see that in the measure induced by the foreign exchange rate, the drift of a foreign-denominated asset is the foreign risk-free rate. It is now straightforward to apply the Feynman-Kac theorem to the expectation in Equation 2.96. Define

$$V(0) = X(0)U(0) \quad (2.101)$$

where

$$U(0) = E_x[e^{-r_f T} F(S(T))] \quad (2.102)$$

We obtain  $U(S,t)$  by solving the PDE

$$\frac{\partial U}{\partial t} + r_f S \frac{\partial U}{\partial S} + \frac{1}{2} S^2 \sigma_s^2 \frac{\partial^2 U}{\partial S^2} = r_f U \quad (2.103)$$

with end conditions  $U(S,T) = F(S(T))$ . This shows that the Type 1 currency translated options is truly a one-dimensional problem.

**Type 2.** The payoff function of this type depends only on the product  $S(t)X(t) = G(t)$ . The value of the option is then

$$V(0) = E[e^{-rT} F(G(T))] \quad (2.104)$$

To evaluate the expectation, we need the process for  $G(t)$  in the domestic risk-neutral measure. To do this, we apply Ito's lemma to  $S(t)X(t)$ , with  $S(t)$  and  $X(t)$  given by Equations 2.91 and 2.92. This gives

$$dG = Grdt + G(\sigma_x^2 + 2\rho\sigma_x\sigma_s + \sigma_s^2) dW \quad (2.105)$$

## 54 The Pricing Equations

Our pricing equation is now the one-dimensional problem

$$\frac{\partial V}{\partial t} + rG \frac{\partial V}{\partial G} + \frac{1}{2} (\sigma_x^2 + 2\rho\sigma_x\sigma_s + \sigma_s^2) G^2 \frac{\partial^2 V}{\partial G^2} = rV \quad (2.106)$$

with end condition  $V(T) = F(G(T))$ .

**Type 3.** This is obviously a one-dimensional problem, since the foreign exchange rate is evaluated at  $t = 0$  and does not enter into the payoff function as a stochastic process. The purpose of the exchange rate in the payoff function can be viewed as simply a constant with the appropriate units to convert from the foreign currency to the domestic currency. The value of the option is

$$V(0) = X(0) E[e^{-rT} F(S(T))] \quad (2.107)$$

We can derive the pricing equation by simply expressing  $S(t)$  in the domestic measure. Using Equation 2.98, the pricing equation is

$$\frac{\partial V}{\partial t} + S(r_f - \rho\sigma_x\sigma_s) \frac{\partial V}{\partial S} + \frac{1}{2} \sigma_s^2 S^2 \frac{\partial^2 V}{\partial S^2} = rV \quad (2.108)$$

subject to the end condition  $V(T) = X(0)F(S(T))$ .

**Type 4.** This case is very similar to Type 1. We can get rid of  $S(t)$  by selecting  $S(t)$  as the reference price process, in a similar way as we did in Type 1 with the foreign exchange rate.

Following the same line of analysis as in Type 1, we have the following expression, where  $E^S$  indicates expectation in the measure induced by  $S(t)$ :

$$\frac{V(0)}{S(0)} = E^S \left[ e^{-(\mu_v - \mu_s)T} \frac{V(T)}{S(T)} \right] \quad (2.109)$$

Here,  $\mu_v$  and  $\mu_s$  are the domestic risk-neutral drifts of  $V(t)$  and  $S(t)$ , respectively. Since  $V(t)$  is the value of an asset,  $\mu_v = r$ . From Equation 2.98,  $\mu_s = r_f - \rho\sigma_x\sigma_s$ :

$$\begin{aligned}
\frac{V(0)}{S(0)} &= E_s \left[ e^{-(r-r_f+\rho\sigma_x\sigma_s)T} \frac{V(T)}{S(T)} \right] \\
&= E_s \left[ e^{-(r-r_f+\rho\sigma_x\sigma_s)T} \frac{S(T)F(X(T))}{S(T)} \right] \\
&= E_s [e^{-(r-r_f+\rho\sigma_x\sigma_s)T} F(X(T))]
\end{aligned} \tag{2.110}$$

Finally,

$$V(0) = S(0) E_s [e^{-(r-r_f+\rho\sigma_x\sigma_s)T} F(X(T))] \tag{2.111}$$

To apply the Feynman-Kac theorem to the expectation on the right, we need to express the process for  $X(t)$  in the measure induced by  $S(t)$ . We start out with the risk-neutral process for  $X(t)$  in the domestic measure:

$$dX = X(r - r_f) dt + X\sigma_x dW \tag{2.112}$$

When expressed in the measure induced by  $S(t)$ , this process becomes

$$dX = X(r - r_f + \rho\sigma_x\sigma_s) dt + X\sigma_x dW_s \tag{2.113}$$

where  $W_s$  is a Brownian motion in the  $S(t)$ -measure.

Now we define,

$$V(0) = S(0)U(0) \tag{2.114}$$

where

$$U(0) = E_s [e^{-(r-r_f+\rho\sigma_x\sigma_s)T} F(X(T))] \tag{2.115}$$

We obtain  $U(S,t)$  by solving the PDE

$$\frac{\partial U}{\partial t} + S(r - r_f + \rho\sigma_x\sigma_s)X \frac{\partial U}{\partial X} + \frac{1}{2}\sigma_x^2 X^2 \frac{\partial^2 U}{\partial X^2} = (r - r_f + \rho\sigma_x\sigma_s)U \tag{2.116}$$

with end conditions  $U(S,T) = F(X(T))$ .

### EQUATIONS FOR THE HEDGING PARAMETERS

The sensitivity of the option price to the parameters appearing in the governing pricing equation are known collectively as the Greeks. They are defined as follows:

$$\Delta = \frac{\partial V}{\partial S} \quad (2.117)$$

$$\Gamma = \frac{\partial^2 V}{\partial S^2} \quad (2.118)$$

$$\rho = \frac{\partial V}{\partial r} \quad (2.119)$$

$$\nu = \frac{\partial V}{\partial \sigma} \quad (2.120)$$

$$\Theta = \frac{\partial V}{\partial t} \quad (2.121)$$

where  $\sigma$  is the volatility, and  $r$  is the risk-free rate. Other hedging parameters can be defined. For example, if the asset pays dividends and we model them with a continuous dividend rate  $D_0$ , then we may also define the sensitivity to dividend rate.

$$\rho_d = \frac{\partial V}{\partial D_0} \quad (2.122)$$

Higher order hedging parameters have also been defined, for example, charm and color, which are the time derivatives of  $\Delta$  and  $\Gamma$  respectively.

There are two fundamental ways of computing the Greeks. The simplest way is to use a finite difference discretization to approximate the partial derivatives that constitute the Greeks. A different approach is to formulate PDEs for the Greeks and solve those equations numerically.

## Computation of Greeks by Direct Discretization

A simple use of finite differences for approximating the Greeks is as follows (Chapter 3 considers in depth the issue of accuracy of discretizations of this sort):

$$\Delta(S, t) \approx \frac{V(S + \Delta S, t) - V(S - \Delta S, t)}{2\Delta S} \quad (2.123)$$

$$\Gamma(S, t) \approx \frac{V(S + \Delta S, t) - 2V(S, t) + V(S - \Delta S, t)}{\Delta S^2} \quad (2.124)$$

$$\rho(S, t) \approx \frac{V_{r+\Delta r}(S, t) - V_{r-\Delta r}(S, t)}{2\Delta r} \quad (2.125)$$

$$\nu(S, t) \approx \frac{V_{\sigma+\Delta\sigma}(S, t) - V_{\sigma-\Delta\sigma}(S, t)}{2\Delta\sigma} \quad (2.126)$$

$$\Theta(S, t) \approx \frac{V_{t+\Delta t}(S, t) - V_{t-\Delta t}(S, t)}{2\Delta t} \quad (2.127)$$

where  $V_r(S, t)$  means  $V(S, t)$  evaluated with a risk-free rate  $r$ . Because finite difference calculations generate option value on an entire grid of  $S$  values, the calculation of  $\Delta$ ,  $\Gamma$ , and  $\Theta$  in this manner come at negligible additional computational cost. Evaluation of the other hedging parameters in this manner requires repeated solution of the equation for option value.

## Computation of Greeks through Their Governing Equations

Alternatively, we may evaluate hedging parameters through numerical solution of partial differential equations that can be obtained by differentiating the pricing equation with respect to the relevant market parameter. To illustrate, consider the single-asset Black-Scholes equation:

$$L_S V = 0 \quad (2.128)$$

where

$$L_S = \frac{\partial}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2}{\partial S^2} + (r - d)S \frac{\partial}{\partial S} - r \quad (2.129)$$

and  $\sigma$ ,  $r$ , and  $d$  (the dividend yield) are assumed constant. Then, the Greeks satisfy

$$L_S \Delta + S \sigma^2 \frac{\partial \Delta}{\partial S} + (r - d) \Delta = 0 \quad (2.130)$$

$$L_S \Gamma + 2S \sigma^2 \frac{\partial \Gamma}{\partial S} + \sigma^2 \Gamma + 2(r - d) \Gamma = 0 \quad (2.131)$$

$$L_S \rho + S \frac{\partial V}{\partial S} - V = 0 \quad (2.132)$$

$$L_S \rho_D + S \frac{\partial V}{\partial S} = 0 \quad (2.133)$$

$$L_S \nu + \sigma S^2 \frac{\partial^2 V}{\partial S^2} = 0 \quad (2.134)$$

$$L_S(\Theta) = 0 \quad (2.135)$$

Equations 2.130, 2.131, and 2.135 are included only for completeness. In practice, Equations 2.123, 2.124, and 2.127 are much preferred since they are much more economical to compute. The latter equations may also have theoretically discontinuous initial conditions or initial conditions involving delta functions (at option expiration). If any market parameters are functions of  $S$  or  $t$ —such as  $\sigma(S,t)$  or  $r(t)$ —then Equations 2.130, 2.131, and 2.135 must be modified with additional terms.

In general, if an arbitrary pricing equation is written in terms of an operator  $L_p$

$$L_p V = 0 \quad (2.136)$$

where  $\mathbf{p}$  is a vector of market parameters, and a hedging parameter is defined  $\alpha_i = \frac{\partial V}{\partial p_i}$ , where  $p_i$  is the  $i^{th}$  parameter of the operator  $L_p$ , then the equation obeyed by  $\alpha_i$  is

$$L_p \alpha_i + \frac{\partial L_p}{\partial p_i} = 0 \quad (2.137)$$

In the case of American options, the option value is subject to the constraint  $V \geq P$ , where  $P$  is the payoff value of the option. Similarly, the foregoing equations for Black-Scholes hedge parameters are also subject to constraints:

$$L_S \rho + S \frac{\partial V}{\partial S} - V = 0 \text{ if } V \leq P \quad (2.138)$$

$$\rho = 0 \text{ if } V = P \quad (2.139)$$

$$L_S \rho_D - S \frac{\partial V}{\partial S} = 0 \text{ if } V \leq P \quad (2.140)$$

$$\rho_D = 0 \text{ if } V = P \quad (2.141)$$

$$L_S v + \sigma S^2 \frac{\partial^2 V}{\partial S^2} = 0 \text{ if } V \leq P \quad (2.142)$$

$$v = 0 \text{ if } V = P \quad (2.143)$$

The hedging parameters obey the partial differential equations when the American constraint is not operating, that is, outside the exercise region. However, if the constraint is operating, then the option value is simply equal to the payoff condition. In this case, if the payoff is independent of the parameters of the pricing equation, the hedging parameters must be equal to zero.

## REFERENCES

- Andreasen, J. (1998). The Pricing of Discretely Sampled Asian Options and Lookback Options: A Change of Numeraire Approach. *Journal Compo Fin*, 2, No.1, 5-30.
- Bhar, R, & Chiarella, C. (1995). *Transformation of Heath-Jarrow-Morton Models to Markovian Systems*. Working Paper No. 53, Sydney, Australia: University of Technology.
- Dixit, AK., & Pindyck, RS. (1994). *Investment under Uncertainty*. Princeton, NJ: Princeton University Press.
- Duffie, D., & Singleton, K. (1999). Modeling Term Structures of Defaultable Bonds. *Review of Financial Studios*, 12,687-720.
- Heath, D., Jarrow, R, & Morton, A (1992). Bond Pricing and the Term Structure of Interest Rates: A New Methodology for Contingent Claims Valuation. *Econometrica*, 60, 77-105.
- Hull, J., & White, A (1990). Pricing Interest Derivative Securities. *Review of Financial Studies*, 3(4), 573-592.
- Ingersoll, J.E., Jr. (1998, Fall). Approximating American Options and Other Financial Contracts Using Barrier Derivatives. *Journal Compo Fin*, 1, 85-112.
- Lamberton, D., & Lapeyre, B. (1996). *Introduction to Stochastic Calculus Applied to Finance*. New York: Chapman and Hall.
- Merton, RC. (1976). Option Pricing When Underlying Stock Returns Are Discontinuous. *Journal of Financial Economics*, 3, 125-144.
- Reiner, E. (1992, March). Quanto Mechanics. *Risk*, 5, 59-63.
- SchOnbucher, P.J. (1997). *Pricing Credit Risk Derivatives*. Working Paper, London: London School of Economics.
- Shaw, W. (1998). *Pricing Derivatives with Mathematica*. Cambridge, England: Cambridge University Press.
- Wilmott, P. (1998). *Derivatives*. Chichester, England: John Wiley & Sons.
- Wilmott, P., Dewynne, J., & Howison, S. (1993). *Option Pricing: Mathematical Models and Computation*. Oxford, England: Oxford Financial Press.
- Zvan, R, Forsyth, P.A, & Vetzal, K. (1997-1998, Winter). Robust Numerical Methods for PDE Models of Asian Options. *Journal Compo Fin*, 1, 39-78.

## *Chapter* 3

# Analysis of Finite Difference Methods

This chapter presents a general framework for analyzing the properties of finite difference discretization schemes for solving the pricing equation. We present a theoretical development of this framework, together with detailed practical examples for the analysis of discretization schemes. This chapter emphasizes the fundamentals and gives the reader sufficient information to understand, analyze, and design finite difference schemes with a fair degree of generality. We do not attempt, however, to present an exhaustive treatment of finite difference theory. For a more extensive analysis, the interested reader may refer to the classical works by Richtmeyer and Morton (1967) and Smith (1985). The treatment in this chapter provides a background on standard difference schemes. More specialized topics, such as up-winding, multigrid, and total variation diminishing (TVD) methods, are current areas of research in finance and are not specifically covered in this book. However, as demonstrated in Chapters 4 and 6, standard difference schemes, when properly understood and implemented, are sufficient for solving a broad range of problems efficiently and accurately.

### MOTIVATION

To set the stage for our analysis, consider the pricing equation for a European option on an asset with price process  $dx = \mu(x,t)dt + \sigma(x,t)dW$ ,

$$\frac{\partial u}{\partial t} + \mu \frac{\partial u}{\partial x} + \frac{1}{2} \sigma^2 \frac{\partial^2 u}{\partial x^2} - ru = 0 \quad (3.1)$$

with end conditions  $u(x, t = T) = F(x)$ . To integrate this equation, we must start with the end conditions and solve the equation backward in time. Numerically, we can accomplish this in two ways. One way is to begin the solution at  $t = T$  and step backward with negative time steps. Alternatively, we can introduce the transformation to backward time  $\hat{t} = T - t$ , and integrate forward in  $\hat{t}$ . In this chapter we choose the latter to make the analysis consistent with standard texts (from a practical point of view; however, the first approach is more convenient because contractual dates affecting cash flows are specified in calendar time,  $t$ ). We rewrite Equation 3.1 as follows:

$$\frac{\partial u}{\partial \hat{t}} = \mu_x \frac{\partial u}{\partial x} + \frac{1}{2} \sigma_x^2 \frac{\partial^2 u}{\partial x^2} - ru \quad (3.2)$$

For the remainder of this chapter, we dispense with the  $\hat{t}$  notation, understanding that  $t$  will refer to backward time.

The core of the finite difference method is the replacement of partial derivatives by finite difference expressions. For example, first and second derivatives may be written

$$\frac{\partial u}{\partial x}(x, t) = \frac{u(x + \Delta x, t) - u(x - \Delta x, t)}{2\Delta x} - TE_1 \quad (3.3)$$

$$\frac{\partial^2 u}{\partial x^2}(x, t) = \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{2\Delta x^2} - TE_2 \quad (3.4)$$

where  $\Delta x$  is an increment of  $x$ .  $TE_1$  and  $TE_2$  are the terms in the Taylor series expansion of the finite difference expressions that must be *subtracted* to recover the derivatives exactly. The finite difference expression in Equations 3.3 and 3.4 are second order central differences, meaning that the leading order terms in  $TE_1$  and  $TE_2$  are proportional to  $\Delta x^2$  and that the finite difference formulas evaluated at  $x$  involve samples of  $u$  symmetrically arranged about  $x$ .

We define an array of equally spaced grid points  $x_i = \{x_0, x_1, \dots, x_l\}$  with  $x_{i+1} - x_i = \Delta x$ , and introduce the notation  $u_i = u(x_i)$ . Then

$$\frac{\partial u}{\partial x}(x_i, t) = \frac{u_{i+1}(t) - u_{i-1}(t)}{2\Delta x} - TE_1, i = 1, \dots, I-1 \quad (3.5)$$

$$\frac{\partial^2 u}{\partial x^2}(x_i, t) = \frac{u_{i+1}(t) - 2u_i(t) + u_{i-1}(t)}{\Delta x^2} - TE_2, i = 1, \dots, I-1 \quad (3.6)$$

We refer to these discretizations as *space discretizations*. The  $u_i, i = 1, \dots, I-1$  are referred to as interior points or values. The quantities  $u_0$  and  $u_I$  are referred to as boundary values. The boundary values may or may not be known ahead of time. If they are not known, they must be found as part of the solution. The most common case is that the boundary values must be found along with the rest of the solution. If the boundary values must be found together with the solution, we must link the boundary values with the rest of the solution. How this is done depends on the problem at hand and is discussed extensively in Chapter 4. Suppose that the solution near the boundaries in the  $x$ -direction is such that we can linearly extrapolate the values  $u_0$  and  $u_I$  using two adjacent points. This gives us the relationships

$$u_0 = 2u_1 - u_2 \quad (3.7)$$

$$u_I = 2u_{I-1} - u_{I-2} \quad (3.8)$$

Our next task is to transform the pricing equation in backward time, Equation 3.2, into a system of ordinary differential equations where the unknowns are the  $u_i(t), i = 0, \dots, I$ .

Substituting the discretized partial derivatives from Equations 3.5 and 3.6, and the boundary relationships given by Equations 3.7 and 3.8 into the pricing equation in backward time, Equation 3.2 we obtain the following system of ordinary differential equations:

$$\frac{d\mathbf{u}}{dt} = \mathbf{A}\mathbf{u} + \boldsymbol{\epsilon} \quad (3.9)$$

where

$$\mathbf{u} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{I-1} \\ u_I \end{bmatrix} \quad (3.10)$$

$$\mathbf{A} = \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 & 0 \\ \beta_1 & -\gamma_1 & \alpha_1 & 0 & \cdots & 0 & 0 \\ 0 & \beta_2 & -\gamma_2 & \alpha_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta_{I-1} & -\gamma_{I-1} & \alpha_{I-1} \\ 0 & 0 & 0 & 0 & 1 & -2 & 1 \end{bmatrix} \quad (3.11)$$

with

$$\alpha_i = \frac{1}{2} \left( \frac{\sigma_i^2}{\Delta x^2} + \frac{u_i}{\Delta x} \right) \quad (3.12)$$

$$\beta_i = \frac{1}{2} \left( \frac{\sigma_i^2}{\Delta x^2} - \frac{u_i}{\Delta x} \right) \quad (3.13)$$

$$\gamma_i = \frac{\sigma_i^2}{\Delta x^2} + r \quad (3.14)$$

and where

$$\epsilon = \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_{I-1} \\ \epsilon_I \end{bmatrix} \quad (3.15)$$

are truncation terms that arise from  $TE_1$  and  $TE_2$ , and the assumptions about the values at the boundaries.

If we now *ignore* these error terms in Equation 3.9, we can regard the solution to the system of ordinary differential equations:

$$\frac{d\mathbf{u}}{dt} = \mathbf{A}\mathbf{u} \quad (3.16)$$

as an approximation to the solution of our partial differential equation (PDE) at the grid points  $x_1, x_2, \dots, x_{l-1}$ .

Our next step is to solve this system of ordinary differential equations numerically. To do this, we discretize the time derivative in Equation 3.16 through a *time discretization scheme*. The approximate solution of Equation 3.16 is yet another approximation to the solution of our PDE, Equation 3.2. We accomplish the numerical integration of Equation 3.16 by implementing a time discretization scheme. To indicate time dependency, we use the more economical notation:

$$u_i^n = u_i(t_n) \quad (3.17)$$

where the time step,  $\Delta t$ , is defined

$$\Delta t = t_n - t_{n-1} \quad (3.18)$$

The objective of the time discretization is to obtain the solution at time  $t_n$  as a function of information available up to and including  $t_{n-1}$ . As an illustration, consider the simplest way to advance Equation 3.16:

$$\mathbf{u}^n = \mathbf{u}^{n-1} + \Delta t \left[ \frac{d\mathbf{u}}{dt} \right]^{n-1} + TE_{\Delta t} \quad (3.19)$$

where  $TE_{\Delta t}$  is a truncation error associated with our time discretization. If we now use this scheme to advance the solution by combining Equations 3.16 and 3.19, ignoring the error term, we obtain the simple scheme

$$\mathbf{u}^n = \mathbf{u}^{n-1} + \Delta t \mathbf{A}^{n-1} \mathbf{u}^{n-1} \quad (3.20)$$

This constitutes the finite difference scheme. By advancing the solution of Equation 3.20 in time, we obtain a solution that *approximates* the exact solution of Equation 3.16, which in turn is also an approximation of the exact solution of Equation 3.2, our pricing equation. In this simple example, we see that there are two fundamental sources of error. One source is the truncation error in the space discretization; the other is the truncation error in the time discretization.

The implication of truncation error is that the numerical scheme solves a problem that is not exactly the same as the problem we are trying to solve. The *approximate* solution of our PDE obtained with the numerical scheme can be viewed as the *exact* solution to a different problem. To characterize what the numerical scheme does, we need to address three fundamental issues:

1. *Consistency.* A numerical scheme is said to be consistent if the finite difference representation converges to the PDE we are trying to solve as the space and time steps tend to zero.
2. *Stability.* A numerical scheme is said to be stable if the difference between the numerical solution and the exact solution remains bounded as the number of time steps tends to infinity.
3. *Convergence.* A scheme is said to converge if the difference between the numerical solution and the exact solution *at a fixed point* in the domain of interest tends to zero uniformly as the space and time discretizations tend to zero (not necessarily independently from each other).

A powerful statement links these issues together. This statement is the Lax Equivalence Theorem: *Given a properly posed linear initial value problem and a consistent finite difference scheme, stability is the only requirement for convergence.* Formal proof of this theorem is beyond the scope of this book (Richtmeyer & Morton, 1967). This theorem illustrates why we will dedicate much effort to analyzing and understanding stability.

The careful reader may wonder about the relevance of the concept of consistency. In the simple example we discussed, the truncation errors  $TE_l'$ ,  $TE_z'$  and  $TE_M$  can obviously be driven to zero by taking  $ilx$  and  $M$  sufficiently small. This means that the scheme

will indeed describe the problem we are trying to solve as  $\Delta x \rightarrow 0$  and  $\Delta t \rightarrow 0$ . Consistency, in this case, does not appear to be a relevant notion because in this simple example the space and time discretizations were kept neatly separate. This does not have to be the case. To illustrate, consider a discretization of the second derivative used in the DuFort and Frankel (1953) method for diffusion equations:

$$\frac{\partial^2 u}{\partial x^2}(x_i, t) \approx \frac{u(x_{i+1}, t) - u(x_i, t + \Delta t) - u(x_i, t - \Delta t) + u(x_{i-1}, t)}{\Delta x^2} \quad (3.21)$$

In this case, the truncation error of the space discretization and the truncation error of the time discretization are mixed, and it is not clear that the resulting total truncation error will tend to zero as the space and time steps tend to zero. Lax's theorem comes to the rescue, stating that the scheme will be convergent if consistency can be obtained by requiring that the space and time discretization steps tend to zero in a *related* way. In what follows, we will consider consistent schemes only.

The vague issue of the "accuracy" of a scheme is not very relevant in its own right. A consistent scheme can be made increasingly accurate by decreasing the time and spatial steps. What matters is the cost of accuracy. Here, cost must be interpreted broadly and must include coding effort, memory requirements, computational speed, and robustness.

## CONSTRUCTING FINITE DIFFERENCE APPROXIMATIONS

We can construct finite difference approximations of the derivatives of a function by expressing the derivative as a linear combination of the function evaluated at a number of adjacent points. By expanding the function evaluations in Taylor series, we can determine the coefficients of the linear combination such that the derivative we seek is expressed with any desired order of accuracy.

We will illustrate a systematic way to obtain finite difference approximations in one dimension. The extension to multiple dimensions is trivial. The Taylor series expansion of  $u(x + m\Delta)$  is

$$\begin{aligned} u(x + m\Delta x) &= u(x) + m\Delta x \frac{\partial u}{\partial x}(x) + m^2 \frac{1}{2} \Delta x^2 \frac{\partial^2 u}{\partial x^2}(x) \\ &\quad + m^3 \frac{1}{6} \Delta x^3 \frac{\partial^3 u}{\partial x^3}(x) + \dots \end{aligned} \quad (3.22)$$

Suppose we want to construct a second-order accurate finite difference scheme for  $u_x(x)$  using function values at  $x, x - \Delta x$ , and  $x - 2\Delta x$ . This particular arrangement of finite differences is called one-sided differences. In this case,  $m = \{-2, -1, 0\}$ . We first postulate the following expression:

$$\Delta x \frac{\partial u}{\partial x}(x) = \alpha_{-2} u(x - 2\Delta x) + \alpha_{-1} u(x - \Delta x) + \alpha_0 u(x) + TE \quad (3.23)$$

where  $\alpha_k$  are unknown coefficients to be determined such that the accuracy we require of the scheme is satisfied. Substituting the Taylor series given by Equation 3.22 into Equation 3.23 and equating powers of  $\Delta x$ , we find that  $TE = O(\Delta x^3)$  if the coefficients  $\alpha_k$  satisfy the linear system of equations

$$\begin{bmatrix} 1 & 1 & 1 \\ -2 & -1 & 0 \\ 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha_{-2} \\ \alpha_{-1} \\ \alpha_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (3.24)$$

Solving Equation 3.24 and replacing  $\alpha_k$  in Equation 3.23, we get the following finite difference discretization:

$$\frac{\partial u}{\partial x}(x) = \frac{3u(x) - 4u(x - \Delta x) + u(x - 2\Delta x)}{2\Delta x} + \vartheta(\Delta x^2) \quad (3.25)$$

As an additional example, assume we want a second order scheme for  $\frac{\partial^2 u}{\partial x^2}$  using values at  $x, x + \Delta x$ , and  $x - \Delta x$ . This arrangement is called central differencing. For the expression

$$\frac{1}{2} \Delta x^2 \frac{\partial^2 u}{\partial x^2}(x) = \alpha_{-1} u(x - \Delta x) + \alpha_0 u(x) + \alpha_1 u(x + \Delta x) + TE \quad (3.26)$$

to be second order accurate, we require that  $TE = O(\Delta x^4)$  and that the  $\alpha_k$  must satisfy the system of equations:

$$\begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.27)$$

**Table 3.1 Commonly Used Finite Difference Discretization Schemes**

Derivative	FD Scheme	Accuracy
$\frac{\partial u}{\partial x}$	$\frac{u(x + \Delta x, y) - u(x, y)}{\Delta x}$	$O(\Delta x)$
$\frac{\partial u}{\partial x}$	$\frac{u(x, y) - u(x - \Delta x, y)}{\Delta x}$	$O(\Delta x)$
$\frac{\partial u}{\partial x}$	$\frac{u(x + \Delta x, y) - u(x - \Delta x, y)}{2\Delta x}$	$O(\Delta x^2)$
$\frac{\partial u}{\partial x}$	$\frac{-u(x + 2\Delta x, y) + 4u(x + \Delta x, y) - 3u(x, y)}{2\Delta x}$	$O(\Delta x^2)$
$\frac{\partial u}{\partial x}$	$\frac{3u(x, y) - 4u(x + \Delta x, y) + u(x - 2\Delta x, y)}{2\Delta x}$	$O(\Delta x^2)$
$\frac{\partial^2 u}{\partial x^2}$	$\frac{u(x + 2\Delta x, y) - 2u(x + \Delta x, y) + u(x, y)}{\Delta x^2}$	$O(\Delta x)$
$\frac{\partial^2 u}{\partial x^2}$	$\frac{u(x, y) - 2u(x - \Delta x, y) + u(x - 2\Delta x, y)}{\Delta x^2}$	$O(\Delta x)$
$\frac{\partial^2 u}{\partial x^2}$	$\frac{u(x + \Delta x, y) - 2u(x, y) + u(x - \Delta x, y)}{\Delta x^2}$	$O(\Delta x^2)$
$\frac{\partial^2 u}{\partial x^2}$	$\frac{-u(x + 3\Delta x, y) + 4u(x + 2\Delta x, y) - 5u(x + \Delta x, y) + 2u(x, y)}{\Delta x^2}$	$O(\Delta x^2)$
$\frac{\partial^2 u}{\partial x^2}$	$\frac{2u(x, y) - 5u(x - \Delta x, y) + 4u(x - 2\Delta x, y) - u(x - 3\Delta x, y)}{\Delta x^2}$	$O(\Delta x^2)$
$\frac{\partial^2 u}{\partial x \partial y}$	$\frac{u(x + \Delta x, y) - u(x + \Delta x, y - \Delta y) - u(x, y) + u(x, y - \Delta y)}{\Delta x \Delta y}$	$O(\Delta x) + O(\Delta y)$
$\frac{\partial^2 u}{\partial x \partial y}$	$\frac{u(x, y + \Delta y) - u(x, y) - u(x - \Delta x, y + \Delta y) + u(x - \Delta x, y)}{\Delta x \Delta y}$	$O(\Delta x) + O(\Delta y)$
$\frac{\partial^2 u}{\partial x \partial y}$	$\frac{u(x, y) - u(x, y - \Delta y) - u(x - \Delta x, y) + u(x - \Delta x, y - \Delta y)}{\Delta x \Delta y}$	$O(\Delta x) + O(\Delta y)$
$\frac{\partial^2 u}{\partial x \partial y}$	$\frac{u(x + \Delta x, y + \Delta y) - u(x + \Delta x, y) - u(x, y + \Delta y) + u(x, y)}{\Delta x \Delta y}$	$O(\Delta x) + O(\Delta y)$
$\frac{\partial^2 u}{\partial x \partial y}$	$\frac{u(x + \Delta x, y + \Delta y) - u(x + \Delta x, y - \Delta y) - u(x - \Delta y, y + \Delta y) + u(x - \Delta x, y - \Delta y)}{4\Delta x \Delta y}$	$O(\Delta x^2) + O(\Delta y^2)$

The solution gives the following expression, which we used earlier in the chapter.

$$\frac{\partial^2 u}{\partial x^2}(x) = \frac{u(x + \Delta x) - 2u(x) + u(x - \Delta x)}{\Delta x^2} + O(\Delta x^2) \quad (3.28)$$

Table 3.1 on page 69 contains some commonly used difference schemes and their accuracy. Central difference schemes are typically used in the interior of the solution field, while one-sided schemes tend to be used at boundaries.

### STABILITY ANALYSIS: MATRIX APPROACH

The analysis of stability is of fundamental importance for the proper implementation of finite difference schemes. The process of analyzing the stability of a scheme is not simply concerned with determining whether the numerical solution may "blow up" under certain conditions. This analysis also sheds light on the ability of the scheme to properly handle features of the instrument's price, such as the capacity to capture large gamma exposures. The methodology also helps us understand the natural limitations of certain schemes and the ways to work with such schemes without being concerned that their limitations may impact the quality of the results.

There are two fundamental ways of analyzing the stability of finite difference schemes; the matrix approach and the Fourier analysis approach. The matrix approach is more comprehensive than the Fourier approach in that it fully captures the effect of boundary conditions. It also provides an excellent framework for understanding the nuances of schemes. The Fourier approach, discussed later in this chapter, is much more straightforward. Although it does not address the effect of boundary conditions, it is extremely popular. The fact that the Fourier approach does not include the effect of the boundaries is not a significant drawback in many financial applications. The Fourier analysis is more appropriate for a quick assessment of the stability features of a scheme.

As illustrated by the simple example in the previous section, solving the pricing equation through finite differences involves the following two steps:

1. Application of a space discretization scheme that transforms the PDE into a set of coupled ordinary differential equations (ODEs).
2. Application of a time discretization scheme to solve the coupled ODEs.

After applying the time discretization scheme to the space-discretized equations, we obtain a large, sparse system of linear algebraic equations that must be solved at each time step.

The matrix framework for analyzing the properties of the solution of the finite difference equations (FDEs) consists of the following three steps.

1. Obtain a local analytic solution to the ODE system that results from discretizing the space variables. This solution will reveal the connection between the characteristics of the space discretization scheme and the properties of the numerical solution assuming no time discretization.
2. Obtain a local analytic solution of the FDE system that results from time discretization.
3. Compare the two local analytic solutions. This leads to a relationship between the time step and the space discretization scheme. This relationship indicates whether the numerical algorithm is stable and allows us to assess the truncation error.

## Space Discretization

Consider a pricing equation of the form

$$\frac{\partial u}{\partial t} = Lu \quad (3.29)$$

where  $L$  is a partial differential operator containing no time derivatives. Define the vector  $u$  as the function  $u$  evaluated at gridpoints defined by a space discretization scheme. The application of space discretization to  $Lu$  will result in the following system of equations:

$$\frac{d\mathbf{u}}{dt} = \mathbf{A}\mathbf{u} \quad (3.30)$$

where  $A$  is a matrix, whose elements will typically be functions of the state variables and time. Equation 3.30 may also contain an additional term, in the form of a forcing function, which may arise as a result of the way boundary conditions are treated, or as a result of the financial instrument itself. Assume that the boundary conditions are incorporated into matrix  $A$ . The presence of a forcing function does not alter the main conclusions regarding stability.

Assume that  $A$  is nonsingular of rank  $M$  with a set of linearly independent eigenvectors  $x_m$ ,  $m = 1, 2, \dots, M$ . Associated with each eigenvector there is an eigenvalue  $\lambda_m$  which satisfies the relationship:

$$Ax_m = \lambda_m x_m \quad (3.31)$$

Let  $X$  denote a matrix whose columns are the eigenvectors  $x_m$  and let  $\lambda$  denote a diagonal matrix whose diagonal elements are the eigenvalues  $\lambda_m$ . The following relationship holds:

$$X^{-1}AX = \lambda \quad (3.32)$$

To obtain a local analytic solution of Equation 3.30, we first multiply Equation 3.30 by  $X^{-1}$  on the left, taking onto account that  $XX^{-1} = I$ , where  $I$  is the identity matrix,

$$X^{-1} \frac{du}{dt} = X^{-1} A X X^{-1} u \quad (3.33)$$

We now introduce a local time linearization and assume that the elements of  $X$  are independent of time. Substituting from Equation 3.32, Equation 3.33 becomes

$$\frac{dX^{-1}u}{dt} = \lambda X^{-1}u \quad (3.34)$$

Introducing the definition

$$v = X^{-1}u \quad (3.35)$$

the system of ODEs becomes

$$\frac{d\mathbf{v}}{dt} = \lambda \mathbf{v} \quad (3.36)$$

This is an uncoupled system of ODEs whose solution vector  $\mathbf{v}$  has elements

$$v_m = c_m e^{\lambda_m t} \quad (3.37)$$

where  $c_m$  are constants that depend on initial conditions. Equation 3.37 will be the basis for our analysis. Of course, we can retrieve the solution  $u$  from Equation 3.37. This, however, is not necessary for our analysis.

The next step is to introduce the time discretization. The combination of the space and time discretizations constitutes our numerical scheme.

## Time Discretization

Stability and other properties of finite difference algorithms result from the combined effects of the space and time discretization schemes used to construct the algorithm.

To analyze the effect of time discretization, we first introduce the shift operator,  $E^i$ .

Given the definition

$$v^n = v(n\Delta t) \quad (3.38)$$

the shift operator is defined through the relationship

$$E^i v^n = v^{n+i} \quad (3.39)$$

The shift operator should not be confused with the expectation operator. Since these operators are used in different contexts, no confusion should arise. We are particularly interested in polynomials of

the shift operator, because the time-discretized finite differences equations of interest can be expressed as polynomials in the shift operator. Define a polynomial in the shift operator as

$$P(E) = \sum_{i=0}^n a_i E^i \quad (3.40)$$

where  $a_i$  are constant coefficients. An equation of the form

$$P(E)v^n = 0 \quad (3.41)$$

is known as a homogeneous difference equation and has the solution (Mickens, 1990)

$$v^n = \sum_{k=1}^K b_k (\Lambda_k)^n \quad (3.42)$$

where the  $b_k$  are constants and the  $\Lambda_k$  are the roots of the polynomial  $P(\Lambda)$ .

We can now analyze the time discretization of Equation 3.36. We consider the  $j^{th}$  component of  $v$ :

$$\frac{dv_j}{dt} = \lambda_j v_j \quad (3.43)$$

This equation has the analytic solution

$$v_j(t) = c_j e^{\lambda_j t} \quad (3.44)$$

Now we use the notation  $v_j^n$  to represent the time-discretized approximation to  $v_j(n\Delta t)$ . We replace the time derivative in Equation 3.43 by the expression

$$\frac{dv_j}{dt}(n\Delta t) = \frac{1}{\Delta t} \hat{P}(E) v_j^n \quad (3.45)$$

where  $\hat{P}(E)$  is defined as

$$\hat{P}(E) = \sum_{i=1}^M c_i E^i \quad (3.46)$$

where  $M$  represents the number of time levels in the time discretization. As shown in the analysis of multilevel schemes later in the chapter, we can arrange the scheme so that  $i$  is always positive.

Substituting for the time derivative in Equation 3.43, we get the following expression:

$$(\hat{P}(E) - \lambda_j \Delta t) v_j^n = 0 \quad (3.47)$$

or, equivalently

$$P(E) v_j^n = 0 \quad (3.48)$$

with

$$P(E) = \hat{P}(E) - \lambda_j \Delta t \quad (3.49)$$

Using the results of the previous section, the solution to this equation is

$$v_j^n = \sum_{k=1}^K c_{jk} (\Lambda_{jk})^n \quad (3.50)$$

where  $c_{jk}$  are constants and  $\Lambda_{jk}$  are solutions of

$$\hat{P}(\Lambda_{jk}) - \lambda_j \Delta t = 0 \quad (3.51)$$

In general, each  $\lambda_j$  is related to  $\Lambda_{jk}$ ,  $k = 1, 2, \dots, K$ , which themselves are determined by the space discretization scheme. Therefore, the  $\Lambda_{jk}$  arise from a particular combination of space-time discretizations. Evidently, a particular space-time discretization that gave rise to

## 76 Analysis of Finite Difference Methods

any  $\Lambda_{jk}$  with absolute value greater than one would be unstable. The  $\Lambda_{jk}$  are called amplification factors.

The relationship between time discretization and space discretization is embodied in the relationship between  $\lambda_j$  and the  $\Lambda_{jk}$ . To analyze this relationship explicitly, we set  $t = n\Delta t$  in Equation 3.44:

$$v_j(t) = c_j e^{(\lambda_j \Delta t)n} \quad (3.52)$$

and expand the right-hand side in Taylor series:

$$v_j(t) = c_j [1 + \lambda_j \Delta t + \frac{1}{2!} (\lambda_j \Delta t)^2 + \dots]^n \quad (3.53)$$

From Equation 3.51, we see that the  $\Lambda_{jk}$  depend on  $\lambda_j$  and the time step  $\Delta t$  only through the product  $\lambda_j \Delta t$ . Furthermore, if  $v_j^n$  from Equation 3.50 is to converge to  $v_j(n\Delta t)$ , from Equation 3.53, as  $\Delta t \rightarrow 0$ , at least one of the  $\Lambda_{jk}$  must converge to the expansion on the right-hand side of Equation 3.53 as  $\Delta t \rightarrow 0$ . Formally, this means that at least one  $\Lambda_{jk}$  for some  $k$  must be expressible as follows:

$$\Lambda_{jk} = 1 + \lambda_j \Delta t + \frac{1}{2!} (\lambda_j \Delta t)^2 + \dots + O(\Delta t^{p+1}) \quad (3.54)$$

where the  $p^{\text{th}}$  term is the first that deviates from the Taylor expansion of  $e^{\lambda_j \Delta t}$ .  $p$  is called the order of the scheme. For each  $j$ , the remaining  $\Lambda_{jk}$  are called spurious roots or spurious amplification factors. In general, the spurious roots have two primary effects on the numerical solution. They contribute to the error of the scheme, and they may limit its stability bounds.

The  $\lambda$  eigenvalues are determined by the space discretization scheme. Some of these eigenvalues are fundamentally relevant to the solution we are trying to obtain. The more extreme eigenvalues, however, may be the result of the discretization itself, and do not necessarily correspond to a feature of the problem that we are interested in resolving. These are known as parasitic eigenvalues, and although they do not represent relevant information, they must be dealt with along with the more relevant ones. A desirable finite difference scheme is capable of resolving the relevant features of the problem without being encumbered by parasitic eigenvalues.

## Analysis of Specific Algorithms

The purpose of this section is to illustrate how the framework developed in this chapter can be used to analyze individual algorithms. Following these steps, the reader can easily scrutinize alternative algorithms (including their own).

The construction of a finite difference scheme consists of two parts. One part is the space discretization, the other is the time discretization. In the matrix approach, the analysis of a particular algorithm starts out by mapping the relationship between the  $A_{jk}$  and the  $A_j$ . In other words, the analysis starts with the time discretization part. This leads to the definition of regions in the  $A_j$  complex plane where the resulting scheme will be either stable or unstable. Space and time discretizations must be such that the resulting scheme is stable and has suitable convergence properties.

**Example 1. The Explicit Euler Scheme.** Consider one of the ordinary differential equations in eigenvector space resulting from space discretization:

$$\frac{dv}{dt} = \lambda v \quad (3.55)$$

Introducing the explicit approximation

$$\frac{dv}{dt} \Big|_n = \frac{v^{n+1} - v^n}{\Delta t} \quad (3.56)$$

we get the following representation of the finite difference problem

$$v^{n+1} = v^n + \lambda \Delta t v^n \quad (3.57)$$

Equivalently,

$$P(E)v^n = 0 \quad (3.58)$$

where the shift polynomial is defined as follows:

$$P(E) = E - 1 - \lambda \Delta t \quad (3.59)$$

## 78 Analysis of Finite Difference Methods

This method has only one amplification factor:

$$\Lambda = 1 + \lambda \Delta t \quad (3.60)$$

To compute the accuracy of this method, we compare Equation 3.60 with the expansion of  $e^{\lambda \Delta t}$ :

$$e^{(\lambda \Delta t)} = 1 + \lambda \Delta t + \frac{1}{2}(\lambda \Delta t)^2 + \dots \quad (3.61)$$

The discrepancy between the right-hand sides in Equation 3.61 and Equation 3.60 is

$$e^{(\lambda \Delta t)} - \Lambda = \frac{1}{2}(\lambda \Delta t)^2 + O(\lambda \Delta t)^3 \quad (3.62)$$

Because the leading order term is  $\vartheta(\Delta t^2)$ , this is a first order accurate method. The stability region is determined by the condition  $|\Lambda| \leq 1$ .

For the diffusion equation

$$\frac{\partial u}{\partial t} = v \frac{\partial^2 u}{\partial S^2} \quad (3.63)$$

discretized with central space differences, it can be shown that the most extreme eigenvalue (Smith, 1985) is  $-\frac{4v}{\Delta s^2}$  (under the assumption that  $u(S, t)$  is zero at the boundaries). It follows that for stability we require

$$\left| 1 - \frac{4v \Delta t}{\Delta s^2} \right| \leq 1 \quad (3.64)$$

This means that the time step must be restricted as

$$\Delta t \leq \frac{\Delta s^2}{2v} \quad (3.65)$$

Notice that if we double the number of spatial grid points, we obtain four times more spatial accuracy at the expense of four times more time steps. This, however, requires eight times more computational work. In many cases, this requirement from stability causes the time step to become much smaller than would be needed to control time discretization accuracy. We can postulate an alternative interpretation of this restriction. The inverse of each of the  $|\lambda_j|$  can be viewed as a characteristic diffusion time. The smallest of such times is proportional to the square of the spatial grid spacing. Denoting this diffusion time by  $\tau_d$ , the stability constraint becomes simply

$$\Delta t \leq 2\tau_d \quad (3.66)$$

This means that the time step is determined by a characteristic time that is typically irrelevant to the problem under consideration. This is an illustration of the serious limitations of explicit schemes.

**Example 2. The Implicit Euler Scheme.** Now consider the following implicit approximation to the derivative in Equation 3.55:

$$\frac{dv}{dt} \Big|^{n+1} = \frac{v^{n+1} - v^n}{\Delta t} \quad (3.67)$$

$$v^{n+1}(1 - \lambda\Delta t) = v^n \quad (3.68)$$

Equivalently,

$$P(E)v^n = 0 \quad (3.69)$$

where the shift polynomial is

$$P(E) = E(1 - \lambda\Delta t) - 1 \quad (3.70)$$

This method also has only one  $\Lambda$  root

$$\Lambda = \frac{1}{1 - \lambda\Delta t} \quad (3.71)$$

where  $\lambda$  is negative. We can expand this expression in terms of  $\lambda\Delta t$ :

$$\Lambda = 1 + \lambda\Delta t + (\lambda\Delta t)^2 + O(\lambda\Delta t)^3 \quad (3.72)$$

We compute the accuracy of this method by comparing this expansion with the expansion of  $e^{\lambda\Delta t}$ . The discrepancy between the two expansions is

$$e^{\lambda\Delta t} - \Lambda = -\frac{1}{2}(\lambda\Delta t)^2 + O(\lambda\Delta t)^3 \quad (3.73)$$

This indicates that this is also only a first order method. Since the requirement for stability,  $|\Lambda| \leq 1$ , is satisfied for all values of  $\lambda\Delta t$  from Equation 3.71, this is an unconditionally stable method.

This method is free from the constraint imposed by Equation 3.65. That constraint was imposed entirely by the discretization of the space dimension and was not connected with the phenomenon described by the partial differential equation. Notice that the relationship between accuracy and computational effort is less clear than in the previous case. Now, advancing the solution requires solving a linear system at each time step. If we assume that the solution of the linear system requires effort linearly proportional to the number of spatial grid points, then we can quadruple the spatial accuracy by doubling the computational effort. This happens in the particularly simple case when we have a tridiagonal matrix to invert at each time step. The relationship between accuracy and computational effort is more complex (and typically less favorable) if the linear system must be solved by more general direct or iterative solvers, as is the case in multiple dimensions.

The fact that the method is unconditionally stable does not necessarily mean that the solution will make sense for large time steps. It simply means that the numerical solution will not blow up.

**Example 3. The Crank-Nicolson Scheme.** This method incorporates both explicit and implicit features. It is unconditionally stable, but may exhibit undesirable qualities if the time step is very large. Despite this, the Crank-Nicolson scheme has been extremely popular for numerical solutions in finance. The main appeal of the method is its

second order accuracy and stability, which are achieved with minor increase in computational cost compared with the implicit method.

Consider an approximation to the derivative in Equation 3.55 that combines explicit and implicit components:

$$\frac{1}{2} \left( \frac{dv}{dt} \Big|_n + \frac{dv}{dt} \Big|^{n+1} \right) = \frac{v^{n+1} - v^n}{\Delta t} \quad (3.74)$$

This gives the representation

$$v^{n+1} = v^n + \frac{1}{2} \lambda \Delta t (v^{n+1} + v^n) \quad (3.75)$$

Equivalently,

$$P(E)v^n = 0 \quad (3.76)$$

where the shift polynomial is

$$P(E) = (1 - \frac{1}{2} \lambda \Delta t) - (1 + \frac{1}{2} \lambda \Delta t) \quad (3.77)$$

This method also has only one  $\Lambda$  root:

$$\Lambda = \frac{1 + \frac{\lambda \Delta t}{2}}{1 - \frac{\lambda \Delta t}{2}} \quad (3.78)$$

We can expand this expression in terms of  $\lambda \Delta t$ :

$$\Lambda = 1 + \lambda \Delta t + \frac{1}{2} (\lambda \Delta t)^2 + \frac{1}{4} (\lambda \Delta t)^3 + O(\lambda \Delta t)^4 \quad (3.79)$$

The discrepancy between this expansion and the expansion of  $e^{\lambda \Delta t}$  is

$$e^{\lambda \Delta t} - \Lambda = \frac{1}{12} (\lambda \Delta t)^3 + O(\lambda \Delta t)^4 \quad (3.80)$$

## 82 Analysis of Finite Difference Methods

This indicates that this is a second order method. As was the case with the implicit Euler method, the requirement for stability,  $|\Lambda| \leq 1$ , is satisfied for all values of  $\lambda\Delta t$  from Equation 3.78. This is also an unconditionally stable method.

The Crank-Nicolson scheme has been shown to have undesirable properties under some circumstances (Shaw, 1998). To understand this, consider what happens for  $\Delta t > > \tau_d$ . In this case, the amplification factor has the limit:

$$\Lambda \rightarrow -1 + \frac{\tau_d}{\Delta t} \quad (3.81)$$

This means that the solution components for which  $\Delta t > > \tau_d$  may not decay appreciably, but simply oscillate in time. These components originate in discontinuities in initial conditions or in price-dependent shocks, such as those induced by discreetly sampled barriers (these topics are discussed in greater detail in Chapter 6). Although these components decay, we are not interested in an accurate description of their time evolution. For a satisfactory solution, these components should disappear after a relatively small number of time steps. After  $q$  time steps, Equation 3.81 implies that the amplitude of these components will have decayed by the factor

$$\Lambda^q \approx (-1)^q \exp\left(-\frac{q\tau_d}{\Delta t}\right) \quad (3.82)$$

This means that we can get a reduction of these components by a factor of about 2.7 after  $q$  time steps if we select  $\Delta t$  as follows:

$$\Delta t = q\tau_d \quad (3.83)$$

We can select the time step to ensure that such components are damped significantly within a given number of time steps. Usually, but not always, it is easy to find a time step that causes the quick disappearance of these unimportant oscillations.

**Example 4. Predictor-Corrector Methods.** A simple predictor-corrector method advances the solution by combining the current

value of the derivative in Equation 3.55 with an approximation to the derivative at the end of the time step. This idea is illustrated best by considering an auxiliary variable,  $\tilde{v}$ , which is used to approximate the derivative at the end of the time step. The structure of a simple predictor corrector method is formulated in two steps. The first step is the predictor part, the second is the corrector part. As an example, consider the following approximation, where we use the explicit Euler method to predict and correct the solution advancement:

$$\frac{\tilde{v}^{n+1} - v^n}{\Delta t} = \frac{dv}{dt}\Big|_n \quad (3.84a)$$

$$\frac{v^{n+1} - v^n}{\Delta t} = \frac{1}{2} \left( \frac{dv}{dt}\Big|_n + \frac{d\tilde{v}}{dt}\Big|^{n+1} \right) \quad (3.84b)$$

Substituting for the derivatives,

$$\tilde{v}^{n+1} = v^n + \lambda \Delta t v^n \quad (3.85a)$$

$$\tilde{v}^{n+1} = v^n + \frac{1}{2} \lambda \Delta t (v^n + \tilde{v}^{n+1}) \quad (3.85b)$$

In this simple example, we can obtain the shift polynomial by first solving for  $\tilde{v}^{n+1}$ . Alternatively, we can represent these two equations in matrix form:

$$\begin{bmatrix} E & -(1+\lambda\Delta t) \\ -\frac{1}{2}\lambda\Delta t E & E - (1 + \frac{1}{2}\lambda\Delta t) \end{bmatrix} \begin{bmatrix} \tilde{v}^n \\ v^n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.86)$$

The shift polynomial is given by

$$P(E) = E - (1 + \lambda\Delta t + \frac{1}{2}(\lambda\Delta t)^2) \quad (3.87)$$

This method also has only one  $\Lambda$  root

$$\Lambda = 1 + \lambda \Delta t + \frac{1}{2}(\lambda \Delta t)^2 \quad (3.88)$$

We now expand this expression in terms of  $\lambda \Delta t$ ; the discrepancy between this expansion and the expansion of  $e^{\lambda \Delta t}$  is

$$e^{\lambda \Delta t} - \Lambda = \frac{1}{6}(\lambda \Delta t)^3 + O(\lambda \Delta t)^4 \quad (3.89)$$

This shows that this method is second order accurate. The condition for stability requires that  $1 + \lambda \Delta t(1 + \frac{1}{2}\lambda \Delta t) \leq 1$ . Since the  $\lambda$  is negative, this requirement translates to  $1 + \frac{1}{2}\lambda \Delta t \geq 0$ . The condition for stability is then

$$\frac{2v\Delta t}{\Delta S^2} \leq 1 \quad (3.90)$$

Compared with the explicit Euler scheme, in this predictor corrector scheme the time step constraint has been relaxed by a factor of two, while at the same time increasing the order of the time accuracy.

**Example 5. Multiple-Level Methods.** Multiple-level schemes use information from two or more points in time. At the start of the numerical integration, information is available only at the current time. Because these methods require information at more than one point in time to be able to work, to start the computation it is typically necessary to obtain the necessary information by using a different numerical scheme that only requires current information and is of comparable accuracy. We will refer to this auxiliary scheme needed to start the multiple-level method as the starting scheme.

Multiple-level methods are methods with spurious roots and parasitic eigenvalues. The behavior of the spurious solutions will be affected by the choice of starting scheme.

The following simple scheme is an example of a multiple-level method:

$$\frac{3}{2} \frac{v^{n+1} - v^n}{\Delta t} - \frac{1}{2} \frac{v^n - v^{n-1}}{\Delta t} = \frac{dv}{dt}|^{n+1} \quad (3.91)$$

Substituting for the time derivative, we can write this expression either as

$$\frac{3}{2} (v^{n+1} - v^n) - \frac{1}{2} (v^n - v^{n-1}) = \lambda \Delta t v^{n+1} \quad (3.92)$$

or

$$\frac{3}{2} (v^{n+2} - v^{n+1}) - \frac{1}{2} (v^{n+1} - v^n) - \lambda \Delta t v^{n+2} = 0 \quad (3.93)$$

The shift polynomial is given by

$$P(E) = \left( \frac{3}{2} - \lambda \Delta t \right) E^2 - 2E + \frac{1}{2} \quad (3.94)$$

The amplification factors are the solutions to

$$(3 - 2\lambda \Delta t) \Lambda^2 - 4\Lambda + 1 = 0 \quad (3.95)$$

Here we have two amplification factors, as follows (here we use the notation  $\Lambda_{1,2}$  to mean  $\Lambda_{jk}$ ,  $k = 1$  and  $\Lambda_{jk}$ ,  $k = 2$ ).

$$\Lambda_{1,2} = \frac{2 \pm \sqrt{1 + 2\lambda \Delta t}}{3 - 2\lambda \Delta t} \quad (3.96)$$

where  $\Lambda_1$  is identified with the + sign. Expanding the right-hand side of this equation in powers of  $\lambda \Delta t$ , we get the following expressions for each root:

$$\Lambda_1 = 1 + \lambda \Delta t + \frac{1}{2} (\lambda \Delta t)^2 + \frac{1}{2} (\lambda \Delta t)^3 + O(\lambda \Delta t)^4 \quad (3.97)$$

$$\Lambda_2 = \frac{1}{3} - \frac{1}{9}\lambda\Delta t + \frac{5}{54}(\lambda\Delta t)^2 + \frac{17}{162}(\lambda\Delta t)^3 + O(\lambda\Delta t)^4 \quad (3.98)$$

This makes it clear that only  $\Lambda_1$  is a valid approximation to  $e^{\lambda\Delta t}$ . The truncation error of the scheme is

$$e^{\lambda\Delta t} - \Lambda_1 = -\frac{1}{3}(\lambda\Delta t)^3 + O(\lambda\Delta t)^4 \quad (3.99)$$

This shows that this scheme is second order accurate. Stability requires that the absolute values of both  $\Lambda_1$  and  $\Lambda_2$  be less than or equal to 1. This is always the case if  $\lambda < 0$ . As a result, this scheme is also unconditionally stable.

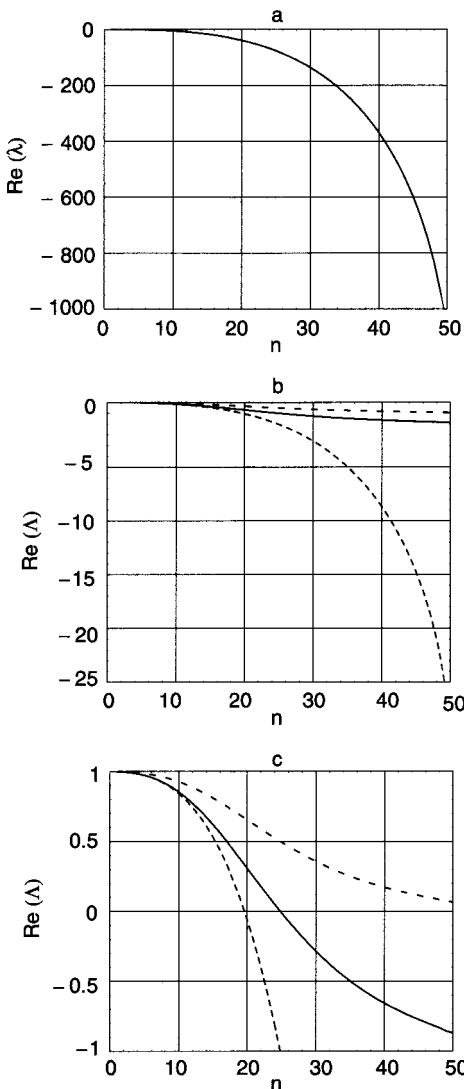
Notice that both amplification factors in this scheme tend to zero for large  $\lambda$ . This means that this scheme is free from the oscillations that may occur with Crank-Nicolson. Furthermore, the spurious amplification factor never exceeds  $\frac{1}{3}$ . The components associated with high values of  $|\lambda|$  decay rapidly through both the principal and spurious amplification factors.

### Eigenvalue Analysis of the Black-Scholes Equation

In this section, we specialize the analysis to the one-dimensional Black-Scholes equation:

$$\frac{\partial V}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} + (r - D_0)S \frac{\partial V}{\partial S} - rV = 0 \quad (3.100)$$

In Figure 3.1(a) we display the real part of eigenvalues  $\lambda_i$  and corresponding amplification factors  $\Lambda_i$  for a second order central difference discretization of the Black-Scholes equation in the native spot price  $S$  coordinates. In the cases discussed here, the imaginary components caused by the drift terms of both the eigenvalues and the amplification factors are quite small. A grid of 51 points is used, with  $S_{\min} = 0$  and  $S_{\max} = 250$  ( $\Delta S = 5$ ). The interest rate, dividend yield, and volatility are respectively,  $r = 0.05$ ,  $D_0 = 0.025$ ,  $\sigma = 0.5$ . For simplicity, the boundary conditions at both boundaries of the spot price grid are of Neumann type,  $\frac{\partial V}{\partial S}$ . This type of boundary condition would apply to the case of digital options (in Chapter 4 we



**Figure 3.1** Real part of eigenvalues  $\lambda_i$  and corresponding amplification factors  $\Lambda_i$  for a second-order central difference discretization of the Black-Scholes equation in the native spot price  $S$  coordinates. A grid of 51 grid points is used. (a) The 51 eigenvalues. (b) The corresponding 51 amplification factors  $\Lambda_i$  for three time discretizations, implicit Euler (dashed curve), Crank-Nicolson (solid curve), and explicit Euler (dotted curve). The time step size is  $1/40$  year ( $T = 1$  and  $N = 40$ ). (c) Same plot as (b), but on an expanded scale. Parameters used:  $T = 1$ ,  $D_0 = 0.025$ ,  $r = 0.05$ ,  $\sigma = 0.5$ ,  $S_{\min} = 0$ ,  $S_{\max} = 250$ ,  $(\Delta S = 5)$ .

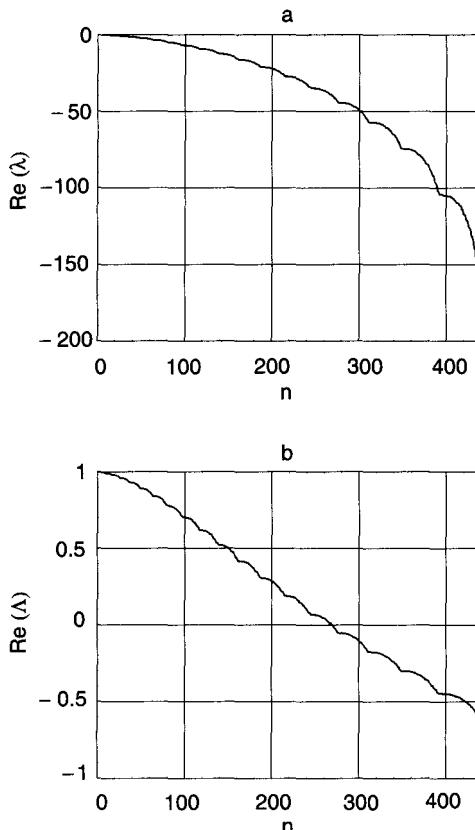
explore the effect of various boundary conditions on these results). The largest magnitude eigenvalue is 1Asolz 1000.

When these eigenvalues are substituted into Equations 3.61, 3.71, and 3.78, with  $ilt = 14.0$  (i.e., 40 time steps per year), we obtain the corresponding amplification factors  $A_i$  for the implicit Euler (dashed curve), Crank-Nicolson (solid curve), and explicit Euler (dotted curve) time discretizations shown in Figure 3.1(b) and (c). The explicit Euler scheme is unstable for these parameters, with the most unstable mode increasing in magnitude by a factor of more than 25 every time step! To stabilize this scheme, the number of time steps would have to be increased to more than 500 per year, even for this very coarse spot price grid. Since the largest-magnitude  $A$  scales with the square of the number of price grid points, we see that a more typical price grid of several hundred points increases the required time steps per year well into the thousands for the explicit Euler scheme. On the other hand, both the Crank-Nicolson and implicit Euler time discretizations are unconditionally stable, with the magnitude of all  $A$  bounded by 1. As discussed earlier, the Crank-Nicolson scheme is accurate to  $O(ilt^2)$ , while the implicit Euler scheme is only accurate to  $O(ilt)$ . However, the implicit scheme does have the property that  $A_{..0}$  for largest magnitude eigenvalues. As explained earlier, this means that the highest spatial frequency modes on the grid decay to 0 rapidly. We should not suppose that we are *accurately* describing this decay in time. Those modes simply go away rapidly.

As has been observed, in the Crank-Nicolson scheme  $A_{..-1}$  for the largest negative eigenvalues. Therefore, these modes decay slowly while alternating in sign each time step. On a very fine price grid,  $A$  may become so large in magnitude that  $A$  may approach -1 for reasonable  $ilt$ . The result is that discontinuities in the payoff condition are manifested as localized oscillations in option value on the grid. The oscillations are worse for the Greeks delta and gamma, which involve derivatives in price. Seldom are price grids required that are fine enough to result in this phenomenon, but when they are, a multilevel scheme can be used, retaining second order accuracy in  $f1t$ . The implicit Euler scheme can also be used, with a loss of accuracy.

Figure 3.2 shows the real part of the eigenvalues for a central difference discretization and the real part of amplification factors for the Crank-Nicolson scheme of the two-dimensional Black-Scholes equation,

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + \frac{1}{2} \sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} + \rho_{12} \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} (r - D_1) S_1 \frac{\partial V}{\partial S_1} \\ + (r - D_2) S_2 \frac{\partial V}{\partial S_2} - rV = 0$$



**Figure 3.2** Real part of eigenvalues  $\lambda_i$  and corresponding amplification factors  $\Lambda_i$  for a second-order central difference discretization of the two-dimensional Black-Scholes equation in the native spot price  $S_1$  and  $S_2$  coordinates. A grid of  $21 \times 21$  points is used. (a) The  $441 = 21^2$  eigenvalues. (b) The corresponding amplification factors  $\Lambda_i$  for a Crank-Nicolson time discretization. The time step size is  $1/20$  year ( $T = 1$  and  $N = 20$ ). Parameters used:  $T = 1$ ,  $D_1 = D_2 = 0.025$ ,  $r = 0.05$ ,  $\sigma = 0.5$ ,  $S_{\min} = 0$ ,  $S_{\max} = 250$ ,  $(\Delta S = 5)$ .

In multiple dimensions, the distribution of eigenvalues and amplification factors looks qualitatively similar to the one-dimensional case.

The distribution of eigenvalues and corresponding amplification factors can be dramatically altered by transforming the coordinates of the pricing equations. For example, by transforming the Black-Scholes equation to log S space, the spectrum of eigenvalues becomes much flatter. This is definitely advantageous if an explicit scheme is used, and can also be helpful if an implicit scheme is used to the extent that spurious oscillations caused by high-magnitude eigenvalues are reduced. This is discussed in Chapter 5.

#### STABILITY ANALYSIS: FOURIER APPROACH

This method, which can be found in many textbooks on elementary numerical analysis, is based on decomposing the numerical solution into Fourier harmonics on the spatial grid. While the Fourier approach does not capture the influence of boundary conditions on stability like the matrix method, it is quite easy to formulate and usually accurate enough to provide practical stability criteria. Associated with each harmonic is an amplification factor, the factor by which the harmonic amplitude changes from time step to time step. Stability conditions are obtained by analyzing the behavior of amplification factors. These amplification factors are closely related to the As we considered in the matrix approach. The difference is that here the amplification factors do not reflect the presence of boundaries. The Fourier method operates directly on the discretized equation.

We illustrate the Fourier approach with two additional examples with the simple diffusion equation:

$$\frac{\partial u}{\partial t} = v \frac{\partial^2 u}{\partial x^2} \quad (3.101)$$

where the diffusion coefficient  $v$  is a constant. The extension of the analysis to a more complicated equation is straightforward.

**Example 6. The Theta Method.** We choose second order central space differencing for the  $x$  derivative and use a time discretization with adjustable “implicitness” parameter  $\theta$ :

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \theta v \frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{\Delta x^2} + (1-\theta) v \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} \quad (3.102)$$

This includes the explicit Euler ( $\theta = 0$ ), the implicit Euler ( $\theta = 1$ ), and the Crank-Nicolson schemes ( $\theta = \frac{1}{2}$ ) as special cases.

We can decompose the solution into Fourier modes on the mesh:

$$u_j^n = \sum_m u_m(t_n) e^{ik_m(j\Delta x)} \quad (3.103)$$

where  $u_m(t_n)$  is the amplitude and  $k_m$  is the wavenumber of mode  $m$ . A real function sampled at a finite number of points on a grid is completely defined by a like number of Fourier modes, whose (real) wavenumbers  $k_m$  are not arbitrary, but satisfy periodicity conditions. These details are not particularly important for the stability analysis, however, and we choose not to clutter the notation by enforcing them. A further simplification is that for linear equations with constant coefficients like this one, the Fourier modes are uncoupled, so that we may consider them individually. Hence we drop the subscript  $m$ . Writing the time dependence of the amplitude in terms of an amplification factor  $\Lambda$ :

$$u_j^n = \Lambda^n e^{ik(j\Delta x)} \quad (3.104)$$

Substitution into Equation 3.102 yields

$$\frac{\Lambda - 1}{\Delta t} = \lambda_k (\theta \Lambda + 1 - \theta) \quad (3.105)$$

where

$$\lambda_k = -\frac{4v}{\Delta x^2} \sin^2\left(\frac{k\Delta x}{2}\right) \quad (3.106)$$

## 92 Analysis of Finite Difference Methods

Solving for  $\Lambda$  we find

$$\Lambda = \frac{1 + (1 - \theta)\lambda_k \Delta t}{1 - \theta\lambda_k \Delta t} \quad (3.107)$$

The stability criterion for this discretization is:

$$|\Lambda| \leq 1 \quad \forall k \quad (3.108)$$

Evaluating Equation 3.107 over the range  $-\infty \leq \lambda_k \leq 0$ , we find that  $-(1 - \theta)/\theta \leq \Lambda \leq 1$ . Therefore, the stability criterion becomes  $\Lambda >= -1$ , which leads to (remember that in our case the  $\lambda$ s are negative)

$$|\lambda_k| \Delta t \leq \frac{2}{2\theta - 1} \quad (3.109)$$

If  $\theta \geq 1/2$ , then the right-hand side of Equation 3.109 is positive and since  $\lambda_k \leq 0$ , the discretization is stable for any  $\Delta t$ . However, if  $\theta < 1/2$ , there exists an upper limit on the allowable  $\Delta t$ . In particular, if  $\theta = 0$ , Equation 3.109 becomes

$$\Delta t \leq \frac{2}{|\lambda_k|} \quad (3.110)$$

This can also be written

$$\Delta t \leq \frac{\Delta x^2}{2v} = 2\tau_d \quad (3.111)$$

where we have used the fact that  $\sin^2(\cdot) \leq 1$ .

**Example 7. Multilevel Scheme.** Here we repeat the analysis of Example 5 provided earlier in this chapter. The multiple-level discretization applied to the diffusion equation is

$$\frac{3}{2} \frac{u_j^{n+1} - u_j^n}{\Delta t} - \frac{1}{2} \frac{u_j^n - u_j^{n-1}}{\Delta t} = v \frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{\Delta x^2} \quad (3.112)$$

The equation for the amplification factor is

$$\frac{3}{2}(\Lambda^2 - \Lambda) - \frac{1}{2}(\Lambda - 1) = \lambda_k \Delta t \Lambda^2 \quad (3.113)$$

which can be written

$$\left( \frac{3}{2} - \lambda_k \Delta t \right) \Lambda^2 - 2\Lambda + \frac{1}{2} = 0 \quad (3.114)$$

This is exactly the same as Equation 3.95.

#### IMPLEMENTATION OF THE TIME ADVANCEMENT

In the matrix analysis of finite difference schemes, we considered the time discretization as if it acted on the decoupled ordinary differential equations that resulted from space discretization. It is important to realize that this decoupling only occurs in eigenvector space. Although there are methods based on solving the decoupled equations (so-called spectral methods), we will not consider such approaches here. How well such methods can deal with contingent cash flows, early exercise, and so forth, is a subject for research.

We will consider time advancement implementations that solve the discretized equations directly, without decoupling them first. Except in the case of purely explicit schemes, this leads to the need to solve a large, sparse linear system at each time step. To illustrate, consider the case of the Crank-Nicolson scheme.

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \frac{1}{2} \left( \frac{d\mathbf{u}}{dt} \Big|_n + \frac{d\mathbf{u}}{dt} \Big|^{n+1} \right) \quad (3.115)$$

Replacing the value of the derivatives from Equation 3.30, we get

$$(\mathbf{I} - \frac{1}{2} \mathbf{A}) \mathbf{u}^{n+1} = (\mathbf{I} + \frac{1}{2} \mathbf{A}) \mathbf{u}^n + \frac{\Delta t}{2} (\mathbf{f}^n + \mathbf{f}^{n+1}) \quad (3.116)$$

This is a system of equations that we can express as follows:

$$\hat{\mathbf{A}}\mathbf{u}^{n+1} = \mathbf{b} \quad (3.117)$$

where the right-hand side is known. Matrix  $\hat{\mathbf{A}}$  can be extremely large (in the case of a two-dimensional problem with a  $100 \times 100$  grid,  $\hat{\mathbf{A}}$  would contain  $10^8$  elements!), and its elements consist mostly of zeroes. Furthermore, the nonzero entries are arranged in diagonal bands. As a result, these systems are called banded systems, and have been the object of extensive research over the years.

The solution of Equation 3.117 calls for extremely efficient algorithms that exploit the sparseness of  $\hat{\mathbf{A}}$ . In problems of practical size, it is essentially impossible to deal with matrix  $\hat{\mathbf{A}}$  as if it were dense. The next section gives a brief description of the issues involved in solving large sparse systems. The literature on banded systems and sparse matrices in general is enormous. For a comprehensive survey, the reader is referred to the survey by Duff (1977).

### Solving Sparse Systems of Linear Equations

In solving a sparse system such as Equation 3.117, we can consider direct solvers or iterative solvers. A direct solver is one that achieves the solution within a finite number of steps. An iterative solver achieves a solution on the basis of satisfying an accuracy criterion. In a direct solver, accuracy of the solution is not a controllable parameter; it depends on the particulars of the implementation and on the characteristics of the algorithm itself. A direct solver is not simply one that would theoretically reach the solution within a finite number of steps, but one that will do so *infinite precision arithmetic*. Notice the special role of accuracy in iterative solvers. The use of accuracy as a termination criterion gives iterative solvers a dimension of flexibility and efficiency. We describe next the strategies followed by direct and iterative solvers in practical implementations.

**Direct Solvers.** In derivatives pricing, direct solvers offer less flexibility than iterative solvers. This will become clear when we discuss American derivatives pricing. The most popular direct solver is the tridiagonal solver, which is applicable to one-dimensional problems. The tridiagonal solver is also relevant to multidimensional problems

when they are treated as a succession of one-dimensional problems, such as in the case of alternating directions implicit (ADI) algorithms (Richtmeyer & Morton, 1967).

The tridiagonal solver is the Gaussian elimination method applied to tridiagonal equations. Due to its simplicity and popularity, we explain it in detail here. The purpose is to solve the tridiagonal system:

$$\begin{bmatrix} b_1 & c_1 & 0 & 0 & \cdots & 0 \\ a_2 & b_2 & c_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & a_{n-1} & b_{n-1} & c_{n-1} & \\ 0 & 0 & 0 & 0 & a_n & b_n \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix} \quad (3.118)$$

This is accomplished in two steps. The first step is a downward (upward) sweep of normalization and elimination, the second step is an upward (downward) sweep that yields the solution. The following version of the algorithm shows the downward sweep first.

Normalization:

$$\begin{bmatrix} 1 & \frac{c_1}{b_1} & 0 & 0 & \cdots & 0 \\ a_2 & b_2 & c_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & a_{n-1} & b_{n-1} & c_{n-1} & \\ 0 & 0 & 0 & 0 & a_n & b_n \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} = \begin{bmatrix} \frac{f_1}{f_2} \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix} \quad (3.119)$$

Elimination:

$$\begin{bmatrix} 1 & \frac{c_1}{b_1} & 0 & 0 & \cdots & 0 \\ 0 & b_2 - \frac{a_2 c_1}{b_1} & c_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & a_{n-1} & b_{n-1} & c_{n-1} & \\ 0 & 0 & 0 & 0 & a_n & b_n \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 - \frac{a_2 f_1}{b_1} \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix} \quad (3.120)$$

Normalization:

$$\begin{bmatrix} 1 & \frac{c_1}{b_1} & 0 & 0 & \cdots & 0 \\ 0 & 1 & \frac{c_2}{b_2 - \frac{a_2 c_1}{b_1}} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & & 0 \\ 0 & 0 & & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & 0 & 0 & 0 & a_n & b_n \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} = \begin{bmatrix} \frac{f_1}{b_1} \\ \frac{f_2 - \frac{a_2 f_1}{b_1}}{b_2 - \frac{a_2 c_1}{b_1}} \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix} \quad (3.121)$$

This is continued until the system looks as follows:

$$\begin{bmatrix} 1 & x_1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & x_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 1 & x_{n-1} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} \quad (3.122)$$

The solution follows immediately in the upward sweep:

$$u_{n-i} = y_{n-i} - x_{n-i} u_{n-i+1}, \quad i = 1, \dots, n-1$$

Sophisticated direct solvers of linear equations will implement the following strategy:

1. Transformation of the system  $\hat{\mathbf{A}}\mathbf{u} = \mathbf{f}$  into a system  $\tilde{\mathbf{A}}\tilde{\mathbf{u}} = \tilde{\mathbf{f}}$  such that  $\tilde{\mathbf{A}}$  can be factored into upper and lower triangulars while maintaining good sparseness (a naive UL decomposition of a sparse matrix will damage sparseness).
2. A symbolic factorization stage where locations of nonzero entries are determined.
3. Actual numerical UL factorization.
4. Solution of the UL system.

**Iterative Solvers.** An iterative solver achieves the solution through an iterative improvement of an initial guess. There are two main

types of iterative solvers. Stationary methods use iteration schemes with parameters that remain fixed during the iterations. Examples of these methods are the Jacobi, Gauss-Seidel, and successive overrelaxation (SOR) methods. Nonstationary methods use parameters that are updated as the iteration proceeds. Examples are the conjugate gradient family and minimal residual methods. The interested reader may consult Barrett (1994) for an extensive reference on iterative methods.

In general, the performance of iterative solvers can be improved significantly if the linear system undergoes a process known as preconditioning. The effectiveness of preconditioners in financial applications is highly problem-dependent. Chapter 4 elaborates on this issue.

Preconditioning consists in replacing the system  $\hat{\mathbf{A}}\mathbf{u} = \mathbf{f}$  with the equivalent system

$$(\hat{\mathbf{A}}\mathbf{C}^{-1})\mathbf{C}\mathbf{u} = \mathbf{f} \quad (3.123)$$

where  $\mathbf{C}$  has been chosen such that the ratio of the largest to the smallest eigenvalue of  $\mathbf{AC}^{-1}$  is as close to 1 as possible. Of course, the matrix  $\mathbf{C}$  that would make the ratio of the extreme eigenvalues equal to 1 is  $\mathbf{A}$  itself (we cannot choose  $\mathbf{C} = \mathbf{A}$  because this would mean solving the problem by finding the inverse of  $\mathbf{A}$ , which is precisely what we are trying to avoid!). The desirable property of  $\mathbf{C}^{-1}$  is to be close to the inverse of  $\mathbf{A}$ . This suggests that preconditioners are in essence versions of linear solvers. Some of the most popular preconditioners are listed in Table 3.2. In addition to these, many of the solvers can be used as preconditioners.

Table 3.3 shows some of the most common solvers that can be used. More detailed information may be obtained in Barrett (1994).

**Table 3.2 Common Preconditioners**

Preconditioner	Main Feature
Diagonal	$\mathbf{C}^{-1}$ is the inverse of the diagonal
Incomplete Choleski	Choleski factorization preserving sparseness
Incomplete LU	LU factorization preserving sparseness
Truncated Neuman polynomial	$\mathbf{C}^{-1}$ is approximate expansion of $\mathbf{A}^{-1}$
Truncated least squares	Minimizes norm of $\mathbf{C}^{-1} - \mathbf{A}^{-1}$
SSOR	Preserves symmetry

Table 3.3 Common Solvers

Solver	Main Feature
Direct	
Diagonal solver	Trivial, primarily a preconditioner
Tridiagonal	One-dimensional only
Gaussian	Gaussian elimination
Iterative	
Jacobi	Primarily a preconditioner
SOR	Successive overrelaxation
SSOR	Symmetric SOR, primarily a preconditioner
Conjugate gradient	Conjugate gradient
CGS	Conjugate gradient squared
Biconjugate gradient	Biconjugate gradient
BCGST	Biconjugate gradient, stabilized
QMR	Quasi-minimal residual method without look-ahead
GMRES	Generalized minimum residual

Any of the iterative solvers can be preconditioned with one of the typical preconditioners shown in Tables 3.2 and 3.3.

The performance of linear solvers is to a large extent responsible for the performance of the solution algorithm. The most appropriate linear solver is highly problem dependent. The ideal combination of preconditioner and solver depends on the structure of the discretization matrix, on the nature of the boundary conditions, on time step, and so forth.

In sophisticated implementations, the combination of preconditioner and solver will allow for an optimal execution strategy. This idea refers to the fact that not all the steps that lead from the original system to the pre-conditioned system need to be repeated at each time step. If matrix  $A$  is changing slowly, then it may not need to be preconditioned at every time step. These are issues of current research in computational finance.

Of the direct solvers listed, the tridiagonal scheme and its generalizations are applicable to one-dimensional problems only. Gaussian elimination, of which the tridiagonal solver is a special case may in principle be used for multidimensional problems but this is never feasible in practice. Of the iterative methods, Jacobi, SOR, and SSOR converge well for diagonally dominant matrices. The conjugate gradient solver requires a symmetric definite matrix to ensure convergence,

but may converge even if this is not true. The remaining methods can handle the asymmetric matrices that arise in multiasset option models due to the correlation terms. More detailed information may be obtained in Barrett (1994).

Although we will not elaborate much on linear equation solvers here, we will describe three of them in some detail: the Jacobi, the Gauss-Seidel, and the successive overrelaxation (SOR) methods. The reason for this is that a particular version of the SOR method plays a pivotal role in pricing American-style options. Furthermore, the Gauss-Seidel method is a generalization of the Jacobi method, and the SOR method in turn is a generalization of the Gauss-Seidel method.

**The Jacobi Method.** Consider the system of linear equations:

$$\sum_{i=1}^{j=n} a_{ij} u_j = f_i \quad (3.124)$$

If we solved for a particular unknown assuming that we know the values of all the others, we would have the following expression:

$$u_i = \frac{1}{a_{ii}} \left( f_i - \sum_{j \neq i} a_{ij} u_j \right) \quad (3.125)$$

This equation suggests an iterative algorithm of the form

$$u_i^{n+1} = \frac{1}{a_{ii}} \left( f_i - \sum_{j \neq i} a_{ij} u_j^n \right) \quad (3.126)$$

where  $n$  stands for iteration number, not to be confused with the time step (this iteration is happening within a given time step).

**The Gauss-Seidel Method.** The Gauss-Seidel method is a simple generalization of the Jacobi method. The only difference is that the changes that occur to the unknowns are incorporated into the scheme as they occur. The algorithm is as follows:

$$u_i^{n+1} = \frac{1}{a_{ii}} \left( f_i - \sum_{j < i} a_{ij} u_j^{n+1} - \sum_{j > i} a_{ij} u_j^n \right) \quad (3.127)$$

Here also we can use the framework we presented in this chapter to analyze the stability and convergence of the scheme.

**The Successive Overrelaxation Method.** The successive overrelaxation method (SOR) is constructed by averaging a Gauss-Seidel iterate with a previous iterate:

$$\tilde{u}_i^{n+1} = \frac{1}{a_{ii}} \left( f_i - \sum_{j < i} a_{ij} u_j^{n+1} - \sum_{j > i} a_{ij} u_j^n \right) \quad (3.128)$$

$$u_i^{n+1} = \omega \tilde{u}_i^{n+1} + (1 + \omega) u_i^n \quad (3.129)$$

The parameter  $\omega$  is called the overrelaxation parameter. Its value strongly affects the rate of convergence of the method. The optimal value of  $\omega$  is in general difficult to compute. In financial applications it is often the case that  $\omega = 1$  is the safest choice.

## FINITE DIFFERENCE APPROACH TO AMERICAN OPTIONS

The application of finite difference schemes to price American-style derivatives is the subject of much ongoing research. Here we present a basic discussion on the finite difference discretization of the partial differential complementarity formulation introduced in Chapter 2. By discretizing the partial differential complementarity problem, we reduce the American option pricing problem to the solution of a sequence of linear complementarity problems (LCP). For greater detail, the reader is referred to the extensive work by Cottle, Pang, and Stone (1992), where convergence issues are thoroughly discussed. For an overview of leading edge research in the application of LCP theory to American option pricing, the reader is referred to the recent work by Huang and Pang (1998).

## The Linear Complementarity Problem

Given a matrix  $\mathbf{A}$  and vectors  $\mathbf{b}$  and  $\mathbf{c}$ , the Linear Complementarity Problem consists in finding vector  $\mathbf{x}$  that satisfies the following conditions:

$$\mathbf{Ax} \geq \mathbf{b} \quad (3.130)$$

$$\mathbf{x} \geq \mathbf{c} \quad (3.131)$$

$$(\mathbf{x} - \mathbf{c}) \cdot (\mathbf{Ax} - \mathbf{b}) = 0 \quad (3.132)$$

To illustrate how the partial differential complementarity formulation of the option pricing problem leads to a sequence of LCPs, assume the following PDCP:

$$\begin{aligned} u(S, t) &\geq F(S, t) \\ \frac{\partial u}{\partial t} - Lu &\geq 0 \\ \left( \frac{\partial u}{\partial t} - Lu \right) (u - F) &= 0 \\ 0 \leq t \leq T, 0 \leq S &\leq \infty \\ u(S, 0) &= F(S, 0), 0 \leq S \leq \infty \end{aligned} \quad (3.133)$$

where

$$Lu = \mu_S \frac{\partial}{\partial S} + \frac{1}{2} \sigma_S^2 \frac{\partial^2}{\partial S^2} - ru \quad (3.134)$$

Consider the application of the Crank-Nicolson scheme to the preceding PDCP. As shown earlier in this chapter, the Crank-Nicolson scheme consists in approximating the  $Lu$  operator as follows:

$$Lu \approx \frac{1}{2} (\mathbf{Au}^{n+1} + \mathbf{Au}^n) + \frac{1}{2} (\mathbf{f}^{n+1} + \mathbf{f}^n) \quad (3.135)$$

where  $\mathbf{A}$  is a matrix and  $\mathbf{f}$  are boundary terms.

Introducing the definitions

$$\mathbf{M} = \mathbf{I} - \frac{\Delta t}{2} \mathbf{A}$$

$$\mathbf{b} = \left( \mathbf{I} + \frac{\Delta t}{2} \mathbf{A} \right) \mathbf{u}^n + \frac{1}{2} (\mathbf{f}^{n+1} + \mathbf{f}^n)$$

and replacing in the definition of the PDCP, we get the following LCP:

$$\begin{aligned} \mathbf{u}^{n+1} &\geq \mathbf{F} \\ \mathbf{M}\mathbf{u}^{n+1} &\geq \mathbf{b} \end{aligned} \tag{3.136}$$

$$\begin{aligned} (\mathbf{u}^{n+1} - \mathbf{F})^T (\mathbf{M}\mathbf{u}^{n+1} - \mathbf{b}) &= 0 \\ \mathbf{u}^0 &= \mathbf{F} \end{aligned} \tag{3.137}$$

where  $\mathbf{F}$  is a discrete approximation to the intrinsic value,  $F$ . This shows that we must solve an LCP at each time step.

An equivalent and more compact version is obtained by making the following substitutions:

$$\mathbf{z} = \mathbf{u} - \mathbf{F} \tag{3.138}$$

$$\mathbf{q} = \mathbf{M}\mathbf{u} - \mathbf{F} \tag{3.139}$$

With this, we get the representation of the LCP discussed by Cottle et al. (1992):

$$\begin{aligned} \mathbf{z} &\geq 0 \\ \mathbf{q} + \mathbf{M}\mathbf{z} &\geq 0 \end{aligned} \tag{3.140}$$

$$\mathbf{z}^T (\mathbf{q} + \mathbf{M}\mathbf{z}) = 0 \tag{3.141}$$

Two fundamental questions in the solution of an LCP refer to the uniqueness of the solution and to the formulation of a suitable method of solution. The answer to both questions depends primarily on the nature of the matrix  $\mathbf{M}$ . As discussed in Cottle, Pang, and

Stone (1992), the vector  $\mathbf{q}$  is not particularly significant in establishing the properties of the LCP.

The LCP has a unique solution for all vectors  $\mathbf{q}$  if and only if the matrix  $\mathbf{M}$  is what is referred to as a P-matrix. A P-matrix is one whose real eigenvalues are all positive; this is equivalent to having all principal minors positive. A positive definite matrix is a particular case of a P-matrix. On the other hand, a P-matrix is not necessarily positive definite, since it need not be symmetric.

The second question we must consider is the selection of a method of solution. Just as it was the case with systems of linear equations, the methods of solution for the LCP fall into two categories: methods that yield the solution in a finite number of steps, called pivoting methods, and iterative methods. Also as is the case with linear systems, the selection of a suitable method is influenced by the size of the problem. A finite difference discretization of the PDCP will typically give rise to a very large matrix  $\mathbf{M}$ , with the same characteristics as the matrices that arise in pricing European options. This is the primary reason we will limit our discussion here to iterative methods.

We now motivate the derivation of iterative methods for solving the LCP by observing that a vector  $\mathbf{z}$  is a solution to an LCP if and only if it satisfies the following relationship:

$$\min[\mathbf{z}, \mathbf{q} + \mathbf{Mz}] = 0 \quad (3.142)$$

If we now represent matrix  $\mathbf{M}$  as the sum of two matrices,  $\mathbf{B}$  and  $\mathbf{C}$ , and assume that a vector  $\mathbf{z}^k$  from the  $k^{th}$  iteration is available, the following recursive algorithm for determining  $\mathbf{z}$  suggests itself:

$$\min[\mathbf{Bz}^{k+1}, \mathbf{q} + \mathbf{Cz}^k + \mathbf{Bz}^{k+1}] = 0 \quad (3.143)$$

Notice that here  $k$  does not refer to a point in time, but it refers to the  $k^{th}$  iteration for the LCP we must solve at a given point in time. We can express this algorithm as a fixed point iteration:

$$\mathbf{Bz}^{k+1} = \mathbf{B}\tilde{\mathbf{z}}^{k+1} \quad (3.144)$$

$$\mathbf{B}\tilde{\mathbf{z}}^{k+1} = \mathbf{Bz}^{k+1} - \min[\mathbf{Bz}^{k+1}, \mathbf{q} + \mathbf{Cz}^k + \mathbf{Bz}^{k+1}] \quad (3.145)$$

Straightforward algebraic manipulations lead to the following iterative algorithm:

$$\mathbf{z}^{k+1} = \max[0, \mathbf{z}^k - \mathbf{B}^{-1}(\mathbf{q} + \mathbf{M}\mathbf{z}^k)] \quad (3.146)$$

There is a parallel between this iterative algorithm and the standard iterative algorithms for solving linear systems. Depending on the choice of matrix  $\mathbf{B}$ , we obtain various LCP versions of the linear system methods. For example, if  $\mathbf{B}$  is chosen as the diagonal of  $\mathbf{M}$ , we obtain the so-called projected Jacobi method. It is called projected because the max ( $\cdot$ ) operator causes the next iterate to be positive.

The most popular algorithm for the solution of the LCP in option pricing is the projected successive overrelaxation algorithm. We obtain this algorithm by the following selection of the  $\mathbf{B}$  matrix:

$$\mathbf{B} = \mathbf{L} + \omega^{-1} \mathbf{D} \quad (3.147)$$

where  $\mathbf{L}$  is the strictly lower triangular part of  $\mathbf{M}$ ,  $\mathbf{D} = \text{diag}(m_{11}, \dots, m_{nn})$ , and  $\omega$  is a relaxation parameter. In this case, care must be taken when applying the max ( $\cdot$ ) operator. The implementation of the PSOR algorithm to the LCP in Equation 3.136 leads to

$$\tilde{u}_i^{k+1} = \frac{1}{m_{ii}} \left( F_i - \sum_{j < i} m_{ij} u_j^{k+1} - \sum_{j > i} m_{ij} u_j^k \right) \quad (3.148)$$

$$u_i^{n+1} = \max[F_i, \omega \tilde{u}_i^{k+1} + (1 - \omega) u_i^k] \quad (3.149)$$

where  $\tilde{u}$  is an intermediate value (this requires these two equations to be solved in succession for each  $i$ ).

The next question to address is under what conditions does the PSOR algorithm converge. For values of  $\omega$  in a range  $1 > \omega \leq 2$ , the PSOR algorithm converges if  $\mathbf{M}$  is an  $H_+$ -matrix (Huang & Pang, 1998).  $\mathbf{M}$  is an  $H_+$ -matrix if the following two conditions are satisfied: The diagonal elements of  $\mathbf{M}$  are positive, and the matrix  $\tilde{\mathbf{M}}$  defined as follows:

$$m_{ii} = |m_{ii}| \quad (3.150)$$

$$\tilde{m}_{ij} = -|m_{ij}|, i \neq j \quad (3.151)$$

is a P-matrix.

A symmetrical, positive definite matrix is a particular case of an H+-matrix. In practical applications, we get a positive definite M if the pricing equation has been transformed into a simple diffusion equation with constant coefficients. If this is not done, the second order discretization of the convection terms will cause the off-diagonal elements of M to be nonsymmetrical. If the matrix is not symmetrical, we can no longer talk of a positive definite matrix in the traditional sense. A nonsymmetrical matrix whose symmetrical part is positive definite is not necessarily an H+-matrix.

Good performance of the PSOR method is typically associated with diagonal dominance. By observing the definition of the matrix M, it is easy to see that diagonal dominance improves as the time step is reduced. When central differences are used for both the convective and diffusion terms, the time step may need to be made sufficiently small for convergence to occur. Huang and Pang (1998) discuss the effect of upwind differencing of convective terms on convergence.

#### DISTORTIONS INDUCED BY DISCRETIZATION

As mentioned at the beginning of the chapter, the introduction of a finite difference approximation changes the nature of the problem we are trying to solve. The reason for this is that the truncation errors are not simply constants, but are proportional to higher order derivatives of the solution function.

Although a precise assessment of the way discretization changes the problem being solved is difficult to accomplish, we can gain some insights by regarding the solution to the pricing equation as made up of Fourier components characterized by dissipation and dispersion. Dissipation can be viewed as the local rate of decay of Fourier components. Dispersion refers to the leading or lagging of Fourier components with respect to each other as they move across the finite

difference grid. The finite difference approximation artificially alters the dissipation and dispersion in the solution field.

The combined dissipation and dispersion induced by the finite difference discretization amounts to a *distortion* of the solution field. We want to assess the nature of this distortion.

The analysis of dissipation and dispersion induced by the numerical approximation is carried out by considering the so-called *modified partial differential equation* (Thomas, 1995; Warming & Hyett, 1974). Consider the pricing equation and central difference approximation introduced at the beginning of the chapter.

$$\begin{aligned} \frac{\partial u}{\partial t} = & \mu \frac{u(x_{i+1}, t) - u(x_{i-1}, t)}{\Delta x} \\ & + \frac{1}{2} \sigma^2 \frac{u(x_{i+1}, t) - 2u(x_i, t) + u(x_{i-1}, t)}{\Delta x^2} - ru \end{aligned} \quad (3.152)$$

We now expand these differences in Taylor series and keep terms of second order. We keep terms of second order because that is the error incurred by this particular central difference approximations. After that is done, the discretized pricing equation becomes equivalent to the following modified partial differential equation:

$$\frac{\partial u}{\partial t} = \mu \frac{\partial u}{\partial x} + \frac{1}{2} \sigma^2 \frac{\partial^2 u}{\partial x^2} + \frac{\Delta x^2 \mu}{6} \frac{\partial^3 u}{\partial x^3} + \frac{\Delta x^2 \sigma^2}{24} \frac{\partial^4 u}{\partial x^4} - ru + O(\Delta x)^3 \quad (3.153)$$

Just as we did in the Fourier stability analysis, we substitute into this equation a function of the form  $u = e^{ikx} e^{(a+ib)t}$ , we find the following local solution to the modified equation:

$$u(x, t) = e^{-\frac{\sigma^2 k^2}{2} \left(1 - \frac{\Delta x^2 \sigma^2 k^2}{12}\right) t} e^{ik \left[x + \mu \left(1 - \frac{\Delta x^2 k^2}{6}\right) t\right]} \quad (3.154)$$

By inspecting this solution, we can make the following observations. The exponent in the first exponential on the right represents dissipation. As a result of the space discretization, the dissipation is less than it should be. The reduction in dissipation is directly proportional to the square of the spatial frequency. This reduction is

due exclusively to the central differencing from the second order derivative in the pricing equation. The second exponential on the right represents a wave traveling at the speed

$$\mu \left( 1 - \frac{\Delta x^2 k^2}{6} \right)$$

In the exact solution, this wave would travel with a speed equal to  $\sim$ . Notice that the change in wave speed is related to the discretization of the first derivative in the pricing equation. The modification of the wave speed is proportional to the square of the spatial frequency. The financial interpretation of these distortions is that the space discretization induces a lower drift and a lower volatility of the underlying process. These distortions are more pronounced if the payoff conditions have discontinuities that tend to populate the high spatial frequencies of the solution.

This analysis refers only to the distortions induced by space discretizations. The introduction of time discretization will induce its own distortions. The nature of such distortions will depend on the particular time differencing scheme, but can be analyzed in a similar fashion.

## STRATEGIES FOR COMPLEX DERIVATIVE STRUCTURES

As discussed in Chapter 2, a complex derivative structure may require the solution of several PDEs simultaneously. In the case of the credit put described in Chapter 2, we need to solve three pricing equations at the same time (the equation for the riskless bond, the equation for the risky bond, and the equation for the credit put itself). This structure, although it seems complex at first sight, is actually relatively straightforward to implement because all three equations can be advanced simultaneously. Furthermore, the three equations are not strongly coupled with each other. The credit put depends on the two bonds, but the bonds don't depend on the credit put nor on each other. If we denote the array of discretized values of the risk-free bond, the defaultable bond, and the put by  $B, B_d$  and  $V$ , respectively, the solution of this problem has the following iterative structure:

$$\begin{bmatrix} \mathbf{A}_{11} & 0 & 0 \\ 0 & \mathbf{A}_{22} & 0 \\ \mathbf{A}_{31} & \mathbf{A}_{32} & \mathbf{A}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{B} \\ \mathbf{B}_d \\ \mathbf{V} \end{bmatrix}^{n+1} = \mathbf{C} \begin{bmatrix} \mathbf{B} \\ \mathbf{B}_d \\ \mathbf{V} \end{bmatrix}^n \quad (3.155)$$

where the  $\mathbf{A}_{ij}$  are block matrices and  $\mathbf{C}$  is a matrix. In fact, there is no fundamental difficulty in solving these equations simultaneously, even if the payoffs of the instruments are coupled in more complex ways.

A more complex situation arises when we have a structure where not all the components can be known at the same time. In this case, we may need to advance some of the components of the structure before we advance others. To illustrate, consider an instrument that has a cash flow proportional to a rate some time after the rate is observed (the simplest case would be a regular interest rate swap). In this case, since we are solving the pricing equations backward, we need to know the payment of the instrument before the rate associated with the payment is observed. Imagine now that that rate is itself a property of another financial instrument (in the case of the swap, this would be a zero coupon bond valued before the payment of the swap happens). What this means is that we would need to solve for the additional financial instrument that determines the rate before we can advance the solution of our derivative. From an implementation viewpoint, this means that the solution to the various components in the structure may have to be staggered in time. In-arrears instruments, where the cash flows occur at the time when they are determined, can be priced with a simpler approach that does not involve staggering.

#### REFERENCES

- Barrett, R., Berry, M., Chan, T.E., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., & van der Vorst, H. (1994). *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Philadelphia: SIAM.
- Cottle, R.W., Pang, J.S., & Stone, R.E. (1992). *The Linear Complementarity Problem*. San Diego: Academic Press.
- Duff, I.S. (1977). A Survey of Sparse Matrix Research. *Proc. IEEE*, 65, 500-535.

- DuFort, E.C., & Frankel, S.P. (1953). Stability Conditions in the Numerical Treatment of Parabolic Differential Equations. *Mathematical Tables and Other Aid to Computation*, 7, 135.
- Huang, J., & Pang, J.S. (1998). Option Pricing and Linear Complementarity. *Journal of Computational Finance*, 2, 31-60.
- Mickens, RE. (1990). *Difference Equations*. New York: Van Nostrand-Reinhold.
- Richtmeyer, R, & Morton, K. (1967). *Difference Methods for Initial-Value Problems*. New York: Interscience.
- Shaw, W. (1998). *Modelling Financial Derivatives with Mathematica*. Cambridge, England: Cambridge University Press.
- Smith, G.D. (1985). *Numerical Solution of Partial Differential Equations: Finite Difference Methods* (3rd ed.). Oxford, England: Clarendon Press.
- Thomas, J.w. (1995). *Numerical Partial Differential Equations*. New York: Springer-Verlag.
- Warming, RF., & Hyett, B.J. (1974). Modified Equation Approach to the Stability and Accuracy Analysis of Finite Difference Methods. *Journal of Computational Physics*, 14, 159-179.

# *Chapter* 4

## Special Issues

In previous chapters, we discussed the finite difference solution of option pricing equations from a theoretical and mathematical perspective. Here we discuss in detail some of the fine points of implementing these algorithms. As in any numerical endeavor, attention to details can yield enormous benefits in terms of code accuracy and efficiency.

### EFFECT OF PAYOFF DISCONTINUITIES ON CONVERGENCE

The errors observed in finite difference solutions have several sources. The first is finite difference truncation error that arises because we have replaced the partial derivatives in the continuous pricing equation with discrete, finite difference approximations. The truncation error associated with many types of discretization is discussed extensively in Chapter 3. For example, with central space differencing in the spatial coordinate  $S$ , and Crank-Nicolson differencing in  $t$ , the truncation error is second order in both space and time:  $\epsilon \sim O(\Delta S^2) + O(\Delta t^2)$ .

However, another source of error can arise if we simply equate the option expiration value on the finite difference grid to the payoff function,  $F(S)$ , sampled at the grid points.

$$V(S_j, T) = F_j = F(S_j) \quad (4.1)$$

The equality is only enforced at the grid points,  $S_j$ . Important features of the payoff function, such as the discontinuity of slope at the strike price for a standard put or call option payoff, or the discontinuity of option value itself for a digital option, may lie between grid

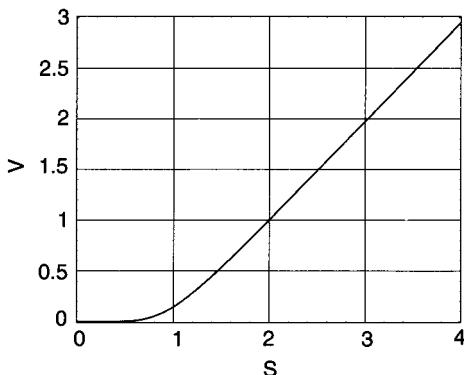
points. This introduces a new source of error that we call quantization error. To reduce it, we must ensure that the calculated present value of an option should depend as little as possible on the location of the strike relative to the grid.

A sometimes useful notion in the implementation of finite difference algorithms is that the value of a function on a grid represents average value of the function over the surrounding grid cell rather than its value sampled at each grid point:

$$F_j \approx \frac{1}{\Delta S} \int_{S_j - \Delta S/2}^{S_j + \Delta S/2} F(s) ds \neq F(S_j) \quad (4.2)$$

For a strictly linear function, the relation becomes an equality,  $F_j = F(S_j)$ . For a standard payoff function (e.g.,  $\max(S - K, 0)$ ), this equality holds at every grid point but one, the one nearest the strike, where the discontinuity in slope gives rise to a difference between averaged and sampled values. Specifically, if the strike falls exactly halfway between grid points, then the equality holds, but if the strike falls exactly on a grid point, then  $F(S_j) = 0$  while  $F_j = \Delta S/8$ . An intuitive approach to reducing quantization error is to set the expiration value of an option on the grid to its cell-averaged value. Then, the numerical integral of the payoff function will be independent of grid position. As noted, for piecewise linear payoff functions, the cell-averaged and the sampled values differ only at the grid points nearest the discontinuities. For more complex payoff functions (e.g., polynomial or exponential), the cell-averaged and sampled payoffs will differ everywhere. In this case, it is optimum to use the sampled value everywhere on the grid except at the grid points nearest discontinuities, where the cell-averaged values can be used.

The numerical results shown in Figures 4.1 and 4.2 demonstrate that this technique reduces quantization error dramatically for a European call option. Figure 4.1 shows the finite difference present option value as a function of spot price  $S$ . The option is struck at  $K = 1$  and the boundaries of the  $S$  grid are  $S_{\min} = 0$  and  $S_{\max} = 5$ . (For clarity, only a portion of the grid is shown in the plots.) In Figures 4.2(a) and (b), we display at-the-money errors in the finite difference calculation compared with the analytic result versus  $I$ , the number of spatial grid points when the expiration option value is set equal to sampled

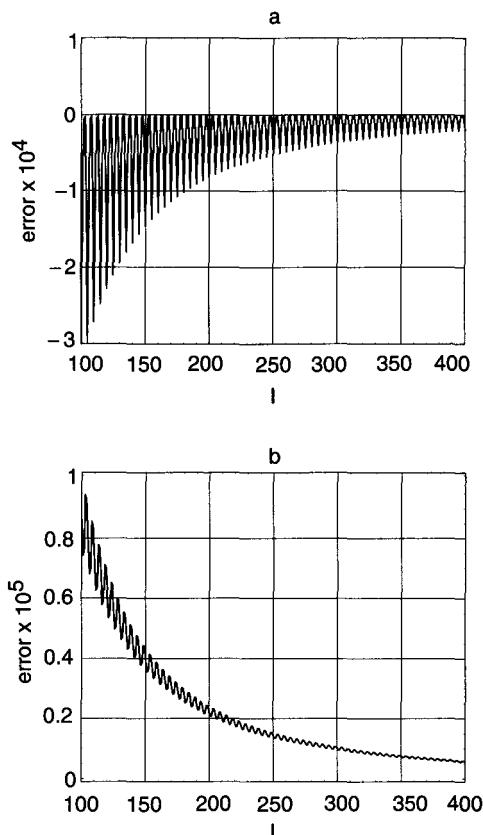


**Figure 4.1** Finite difference present option value for a European call option. Parameters used:  $K = 1$ ,  $\sigma = 0.35$ ,  $r = 0.05$ ,  $D_0 = 0.025$ ,  $T = 1$ ,  $S_{\min} = 0$ ,  $S_{\max} = 5$ .

value of the payoff function  $F(S_i)$ , and when the expiration option value is the cell-averaged payoff function  $F_j$ , respectively. In these two error plots, the error has been scaled by large numbers for clarity. The number of time steps is held fixed for all these results at  $N = 1000$ , which renders the time discretization component of the error negligible compared with the price discretization component.

The error resulting from the sampled payoff function oscillates with an envelope that decreases quadratically with the number of grid points,  $I$ . The oscillations have a period of  $5 = K/(S_{\max} - S_{\min})$ . The error for the cell-averaged payoff function also decreases quadratically with the number of grid points, but is much smoother, and smaller overall by a factor of about 30. It is smooth enough that a reasonable extrapolation can be made to the continuous  $I = \infty$  result from two finite grid results, say  $I = 100$  and  $I = 200$ . Such an extrapolation, which can be used to great advantage for multidimensional problems, is impossible with the sampled payoff results.

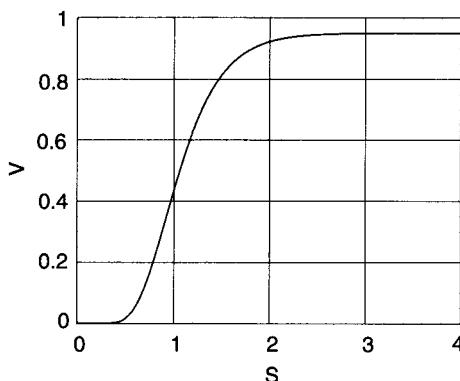
The improvement is even more pronounced for a digital call option. In this case, the option payoff itself, rather than its slope, is discontinuous. Figure 4.3 shows the present option value. Figure 4.4(a) displays the error for the sampled payoff function. It oscillates within an envelope that decreases only as the inverse of  $I$ . With cell averaging of the payoff function, Figure 4.4(b), the envelope of the error is restored to inverse quadratic scaling with  $I$  and is reduced by about a factor of 50. Even with these improvements,



**Figure 4.2** At-the-money errors in the finite difference calculation compared with the Black-Scholes result for a European call option versus  $I$ , the number of spatial grid points. (a) The expiration option value is the sampled payoff function. (b) The expiration option value is the cell average of the payoff function. Parameters used:  $K = 1$ ,  $\sigma = 0.35$ ,  $r = 0.05$ ,  $D_0 = 0.025$ ,  $T = 1$ ,  $S_{\min} = 0$ ,  $S_{\max} = 5$ .

the computed digital option value is not smooth enough to allow accurate extrapolation.

In Chapter 5 we illustrate an alternative approach to reducing quantization error using coordinate transformations. By slightly warping the grid, relatively simple transformations can ensure that critical prices such as strike and barrier prices fall exactly on or midway between grid points while preserving the accuracy of the discretization.



**Figure 4.3** Finite difference present option value for a European digital call option. Parameters used:  $K = 1$ ,  $\sigma = 0.35$ ,  $r = 0.05$ ,  $D_0 = 0.025$ ,  $T = 1$ ,  $S_{\min} = 0$ ,  $S_{\max} = 5$ .

### IMPLEMENTING JUMP CONDITIONS

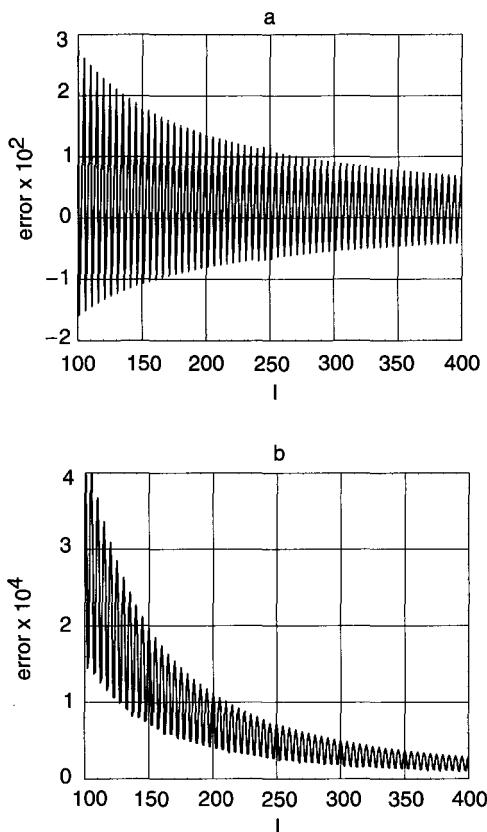
As discussed in previous chapters, jump conditions may be required in option models due to the characteristics of the underlying asset (e.g., discrete dividends), or due to provisions of the option contract itself (e.g., an arithmetic Asian option whose average is sampled at the close of trade on the last day of the week).

Consider the case of discrete dividends. At the  $i^{\text{th}}$  dividend payment time, the following jump condition must be enforced (Wilmott, Dewynne, & Howison, 1993)

$$V(S, t_i^-) = V(S - D_i(S), t_i) \quad (4.3)$$

where  $t_i$  is  $i^{\text{th}}$  dividend payment time,  $t_i^-$  is an instant of time just prior to the  $i^{\text{th}}$  dividend payment time, and  $D_i(S)$  is the amount of the  $i^{\text{th}}$  dividend. The times here refer to forward (calendar) time. In terms of a finite difference calculation in which time is running backward, the time step is adjusted to land exactly on the pending dividend payment time  $t_i$ . Then after the completion of the solution at time  $t_i$ , the option value at spot price  $S$  is replaced by the option value at spot price  $S - D_i(S)$ .

Of course,  $S - D_i(S)$  will not necessarily lie on the finite difference grid, so that the option value  $V(S - D_i(S), t_i)$  must be obtained



**Figure 4.4** At-the-money errors in the finite difference calculation compared with the Black-Scholes result for a European digital call option versus  $I$ , the number of spatial grid points. (a) The expiration option value is the sampled payoff function. (b) The expiration option value is the cell average of the payoff function. Parameters used:  $K = 1$ ,  $\sigma = 0.35$ ,  $r = 0.05$ ,  $D_0 = 0.025$ ,  $T = 1$ ,  $S_{\min} = 0$ ,  $S_{\max} = 5$ .

by interpolation. As discussed in Chapter 3, the most commonly used discretization schemes are second order accurate in price. Therefore, it seems consistent to use piecewise linear interpolation in implementing the jump condition (Wilmott et al., 1993), since this type of interpolation is also second order accurate. However, as shown next, the result can be a spurious numerical diffusion error. This error can be drastically reduced by employing a higher-order interpolation method such as cubic splines.

Temporarily suppressing the time dependence for clarity, we may write the linear interpolation implementation of the jump condition:

$$V(S_k) = \hat{V}(S_k - D_i(S_k)) \quad (4.4)$$

$$\hat{V}(S_k - D_i(S_k)) = (1 - \delta)V(S_j) + \delta V(S_{j+1}) + \epsilon \quad (4.5)$$

$$\epsilon = \frac{1}{2}(S_{j+1} - S_j)^2 \delta(1 - \delta) \frac{\partial^2 V}{\partial S^2}(S_j) + \dots \quad (4.6)$$

$$\delta = \frac{S_k - D_i(S_k) - S_j}{S_{j+1} - S_j} \quad (4.7)$$

Here  $j$  is the index of the grid point  $S_j$  that is closest to, but less than  $S_k - D_i(S_k)$ . The option value  $V(S_k)$  at grid point  $S_k$  is replaced by the linearly interpolated value  $\hat{V}(S_k - D_i(S_k))$  defined in Equation 4.5. Equation 4.6 gives the leading order error term in this approximation. It follows from the definition of the index  $j$ , that  $\delta(1 - \delta) \geq 0$ . Thus, the interpolation error acts as an artificial diffusion or smoothing. In effect, using linear interpolation to implement discrete dividends can add an extraneous numerical volatility. The interpolation error vanishes when either  $\delta = 0$  or  $\delta = 1$  (i.e.,  $S_k - D_i(S_k)$  falls on a grid point), and is maximized when  $\delta = \frac{1}{2}$  (when  $S_k - D_i(S_k)$  falls midway between grid points). For this worst case, the effective volatility in the calculation (averaged over the term of the option  $T$ ) is a function of spot price

$$\frac{\sigma_{\text{eff}}^2(S)T}{2} = \frac{\sigma^2 T}{2} + \frac{\Delta S^2 n_{\text{div}}}{8S^2} \quad (4.8)$$

$$\sigma_{\text{eff}}(S) = \sigma \left( 1 + \frac{\Delta S^2}{S^2} \frac{n_{\text{div}}}{4\sigma^2 T} \right)^{\frac{1}{2}} \quad (4.9)$$

where  $n_{\text{div}}$  is the number of dividends.

A higher order interpolation method can drastically reduce the spurious numerical diffusion. As an example, consider cubic spline interpolation. This is a specific case of piecewise cubic interpolation in which the second derivatives of the interpolating cubic polynomials are forced to be continuous at the grid points. The interpolation formula is conveniently written as piecewise linear interpolation plus a cubic correction term that raises the order of the interpolation error from second to fourth:

$$\begin{aligned}\hat{V}(S_k - D_i(S_k)) &= (1 - \delta)V(S_j) + \delta V(S_{j+1}) \\ &\quad + \frac{(1 - \delta)^2 - 1}{6} V_{SS}(S_j) + \frac{\delta(\delta^2 - 1)}{6} V_{SS}(S_{j+1}) \quad (4.10) \\ &\quad + O[(S_{j+1} - S_j)^4 \delta^2 (1 - \delta)^2]\end{aligned}$$

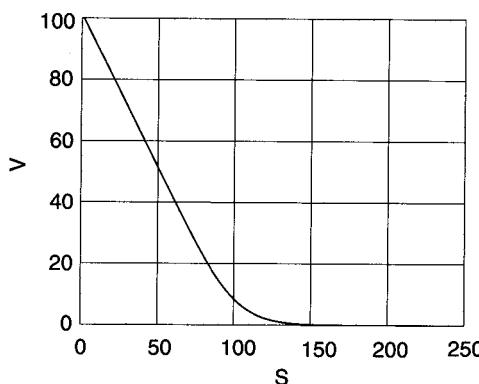
where  $V_{SS}(S_j)$  is the second derivative of the piecewise cubic interpolating polynomials at grid point  $S_j$ . The  $V_{SS}$  coefficients are obtained through solution of a simple tridiagonal system of linear equations. Further details may be found in many numerical analysis texts, for example (deBoor, 1978; Press, Flannery, Teubolsky, & Vetterling, 1986).

A numerical example illustrates the significant reductions in error that can result from implementing jump conditions with a higher order interpolation method. Consider a simple European put option with strike  $K = 100$  and expiration  $T = 1$ . During the life of the option, the underlying asset—a stock index, for example—pays  $n_{\text{div}} = 15$  dividends of  $D_i = 0.5$ , for a total at-the-money dividend yield of 7.5%. The asset volatility is  $\sigma = 0.2$  and the risk-free rate is  $r = 0.06$ . We run four finite difference calculations of present option value, two using linear interpolation for the dividend jump condition, and two using cubic spline interpolation. The grid boundaries,  $S_{\min} = 0$  and  $S_{\max} = 250$  are identical for all calculations. For each interpolation method we use one of two uniform grids, either  $I = 250$  grid points ( $\Delta S = 1$ ) or  $I = 500$  grid points ( $\Delta S = 0.5$ ). On the dense grid  $D_i = \Delta S$ . Hence there is no error for either interpolation method, while on the sparse grid  $D_i = \Delta S / 2$ , and error is maximized for both methods. For either method, the difference between sparse and dense grid calculations is a combination of finite difference truncation error and dividend interpolation error. By comparing the

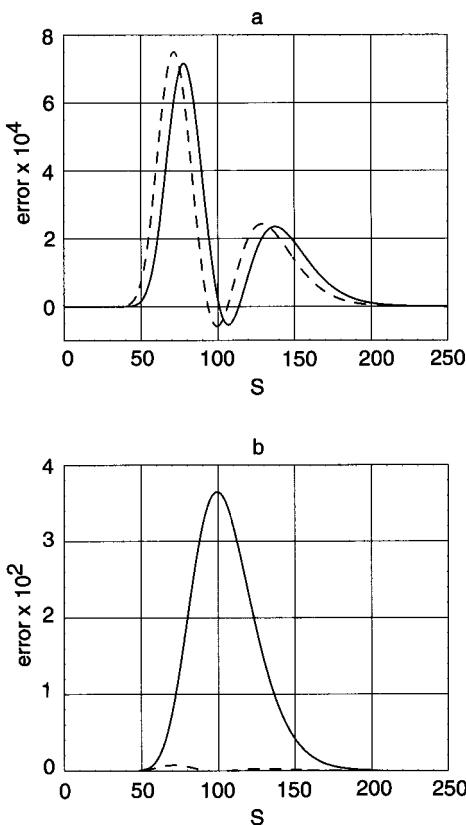
sparse versus dense grid differences with and without dividends, we can isolate the dividend interpolation error.

Figure 4.5 shows present value of the European put using cubic splines and the dense grid. Figure 4.6(a) shows the difference between sparse and dense grid calculations ( $\times 10^4$ ) using cubic splines with dividends (solid curve) and without dividends (dashed curve). The variations in option value with grid density, both with and without dividends, are tiny and except for the displacement along the price axis caused by the dividend payments, nearly identical. Thus, there is no discernible dividend interpolation error. Error is entirely attributable to the truncation error of the finite difference solution itself.

Figure 4.6(b), on the other hand, shows the difference between sparse and dense grid calculations ( $\times 10^2$ ) when linear interpolation is used with dividends (solid curve) and without dividends (dashed curve). In the sparse grid calculation, interpolation error is maximized while in the dense grid calculation it is zero. The differences in option value are about 50 times larger than those attributable to the finite difference truncation error alone. As expected, the option value on the sparse grid is significantly higher due to the numerical diffusion (spurious volatility) associated with the linear interpolation method. The at-the-money pricing error is about 0.036 or about 0.45% of the option value itself. For comparison, the difference in at-



**Figure 4.5** Finite difference present option value for a European put option on asset paying discrete dividends. Parameters used:  $K = 100$ ,  $\sigma = 0.2$ ,  $r = 0.06$ ,  $T = 1$ ,  $S_{\min} = 0$ ,  $S_{\max} = 250$ .



**Figure 4.6** Difference between sparse and dense grid calculations for a European put option on asset paying discrete dividends. (a) Cubic spline interpolation with dividends (solid curve) and without dividends (dashed curve). Both multiplied  $10^4$ . (b) Linear interpolation with dividends (solid curve) and without dividends (dashed curve). Both multiplied  $10^2$ . Parameters used:  $K = 100$ ,  $\sigma = 0.2$ ,  $r = 0.06$ ,  $T = 1$ ,  $S_{\min} = 0$ ,  $S_{\max} = 250$ .

the-money option values (ignoring the dividends) generated by the Black-Scholes formula with  $\sigma_{\text{eff}}(K) = 1.0047\sigma$  versus the true  $\sigma$  is about 0.0345, in agreement with the large pricing error in the finite difference results.

In this simple example we have paid fixed dividends even for very low spot prices,  $S \leq D_i(S)$ , which theoretically yields negative asset prices. There are many possible ways to handle this situation, including simply not paying the dividends or converting the fixed dividend

amounts to percentage dividend yields for very low spot prices. Such details would not materially affect our conclusions regarding interpolation accuracy and we have neglected the issue here.

By picking  $D_i = \Delta S / 2$  on the sparse grid in this example, we have maximized the linear interpolation error. In many cases, the dividend payments will vary with time. However, the average expected error is about two-thirds of this maximum if the variation of the discrete dividend series produces a uniformly distributed  $\delta$ . Only in the relatively uncommon cases that all  $D(S, t_i)$  are much smaller than  $\Delta S$ , or that they are nearly uniform and close to  $\Delta S$ , will the error be reduced significantly. The error will also increase as the number of dividend payments is increased. Therefore, a higher order interpolation method for implementing discrete dividends is recommended.

In this section we discussed only one such method, cubic splines. There are many others that would serve as well, such as piecewise quadratic interpolation. Finally, it might be argued that since the dividend jump condition transports information only from the direction of lower spot prices, that the appropriate interpolation should be "upwinded," using principally information from grid points representing lower spot prices. The cubic spline interpolation discussed uses equally information from both lower and higher spot prices.

Upwinding can become quite difficult to implement for jump conditions associated with complex path dependencies where the direction of information flow can change rapidly on the grid. In practice we have found the details of the interpolation not to make a significant difference as long as the accuracy is higher order than linear.

## BOUNDARY CONDITIONS

In financial problems, one frequently deals with infinite or semi-infinite domains. For example, in a fixed income problem with a simple one factor interest rate model, the solution domain may be  $-\infty \leq r \leq \infty$ ,  $0 \leq r \leq \infty$ , or  $0 < r \leq \infty$ , depending on the specific model. If we transform the pricing equation for an equity option whose underlying process is log normal to  $x = \log(S)$  coordinates, then the solution domain is  $-\infty \leq x \leq \infty$ .

Obviously, an infinite grid cannot be represented in the computer, so we must artificially truncate the solution domain and

replace the deleted portions with boundary conditions that minimize the deleterious effects of the truncation. This problem arises in many other fields, and many techniques for dealing with it have been developed. It is a particularly difficult problem for hyperbolic partial differential equations (i.e., wave propagation problems), where a poor boundary condition can reflect an out-going disturbance back into the solution domain with little loss of amplitude, destroying the desired solution (Randall, 1989).

We are a bit more fortunate in financial modeling, because there is usually a dominant diffusion process, volatility, that attenuates disturbances from an imperfect boundary condition. We place boundaries far enough from the region of interest, so that even if the imposed boundary conditions are imperfect, it does not materially affect the solution. What constitutes "far enough" can be roughly defined. The more accurately the boundary conditions conform to the characteristics of the desired solution, the closer we may place the boundary to the domain of interest.

## Boundary Conditions in One Dimension

Consider a finite difference solution for a simple call option in the Black-Scholes model of constant volatility, interest rate, and dividend yield. At  $S_{\min}$ , we can simply set the option value equal to zero, effectively making the option a down-out call in that region. If  $S_{\min}$  is small enough, the error is negligible. (Of course, if  $S_{\min} = 0$ , the error we make in doing this is zero. But we may not want to extend the grid all the way to  $S_{\min} = 0$  for low volatility or short dated options.) At the  $S_{\max}$  boundary, we could use

$$V(S_{\max}, t) = S_{\max} e^{-D_0(T-t)} - Ke^{-r(T-t)} \quad (4.11)$$

For sufficiently large  $S_{\max}$ , this boundary condition becomes exact. In such a simple case, however, we probably wouldn't be doing a finite difference solution at all, but rather would be using an analytic formula. Usually when a finite difference solution is required (e.g., when there are discrete dividends, when we want to use a local volatility surface  $\sigma(S, t)$  in our model, or when the strike price resets periodically), simple, yet consistent boundary conditions are more difficult to define.

It turns out, however, that in a large number of option structures far from the strike price or other such "interesting" regions, the option value is nearly linear with respect to spot price. This is true even for many exotic and path-dependent options, both European and American. The simplest linearity boundary condition in one dimension is obviously

$$\frac{\partial^2 V}{\partial S^2}(S_{\max}, t) = 0 \quad (4.12)$$

However, this boundary condition requires at least a three-point finite difference operator at the boundary. This would ruin the tridiagonal structure of the space discretization matrix, so that we could no longer use a fast tridiagonal solver. This is not a major disaster, since there are iterative methods we could apply, but there would be a cost in efficiency.

A more economical solution is instead to apply the pricing equation itself as the boundary condition, but with the second derivative term set to zero. For example, in the case of the Black-Scholes equation this "linearity" boundary condition is

$$\frac{\partial V}{\partial t} + (r - D_0) \frac{\partial V}{\partial S} - rV = 0 \quad (4.13)$$

This equation can be discretized in several ways. For example,

$$\begin{aligned} & \frac{V(S, t + \Delta t) - V(S, t)}{\Delta t} \\ & + (r - D_0)S \left( \frac{V(S, t + \Delta t) - V(S - \Delta S, t + \Delta t)}{2\Delta S} \right) \\ & + (r - D_0)S \left( \frac{V(S, t) - V(S - \Delta S, t)}{2\Delta S} \right) \\ & - r \left( \frac{V(S, t + \Delta t) + V(S, t)}{2} \right) = 0 \end{aligned} \quad (4.14)$$

yields an approximation that has errors  $O(\Delta t^2)$  because it is Crank-Nicolson, and  $O(\Delta S)$  because it uses a one-sided difference in the  $S$  direction. In this equation,  $S$  is understood to be the boundary  $S_{\max}$ . We will call this boundary condition BC1. The first order error in  $\Delta S$  is not too grievous because it is proportional to  $\partial^2 V / \partial S^2$ , which we are assuming is zero.

Sometimes we have no reason to postulate a linear dependence of option value on the price, so that the BC1 boundary condition, or any other type of linearity boundary condition, is not appropriate. Furthermore, we may have no other financial argument to which to appeal for a boundary condition. Then we may have no recourse but to apply the pricing equation itself as a boundary condition. Both the drift terms and volatility terms may be discretized with first order one-sided difference operators

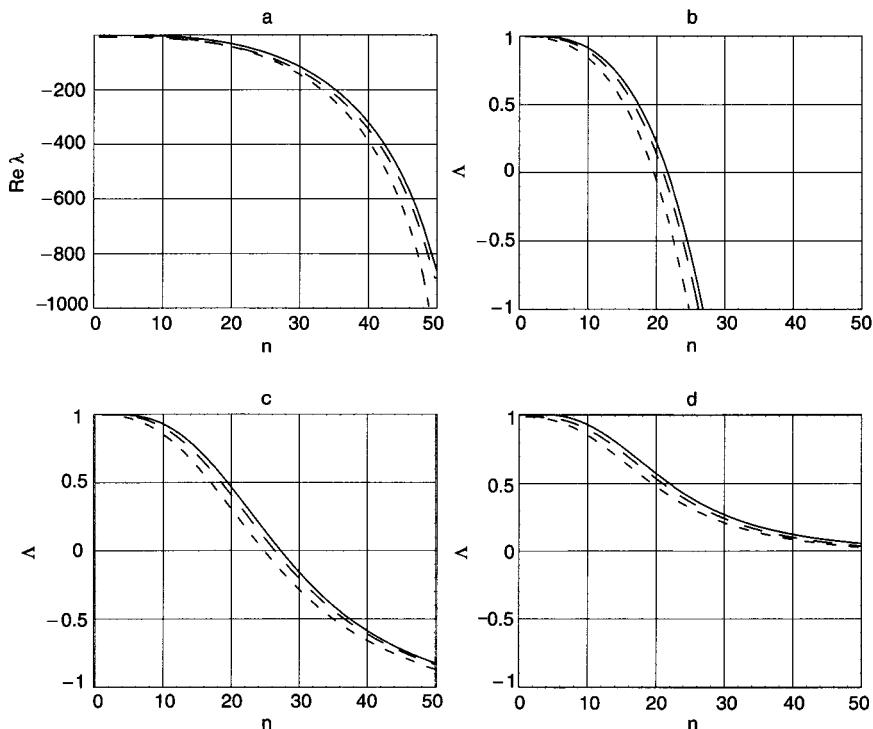
$$\begin{aligned}
 & \frac{V(S, t + \Delta t) - V(S, t)}{\Delta t} \\
 & + \frac{\sigma^2}{2} S^2 \left( \frac{V(S, t + \Delta t) - 2V(S - \Delta S, t + \Delta t) + V(S - 2\Delta S, t + \Delta t)}{2\Delta S^2} \right) \\
 & + \frac{\sigma^2}{2} S^2 \left( \frac{V(S, t) - 2V(S - \Delta S, t) + V(S - 2\Delta S, t)}{2\Delta S^2} \right) \\
 & + (r - D_0) S \left( \frac{V(S, t + \Delta t) - V(S - \Delta S, t + \Delta t)}{2\Delta S} \right) \\
 & + (r - D_0) S \left( \frac{V(S, t) - V(S - \Delta S, t)}{2\Delta S} \right) \\
 & - r \left( \frac{V(S, t + \Delta t) + V(S, t)}{2} \right) = 0
 \end{aligned} \tag{4.15}$$

This discretization, like BC1, is accurate only to  $O(\Delta S)$ . We are now coupling three grid points at the boundary and hence have complicated the solution of the discretized equations. Since this is the case, we can even consider using a second order accurate discretization of the drift terms. We can do so without increasing the coupling of grid points further:

$$\begin{aligned}
& \frac{V(S, t + \Delta t) - V(S, t)}{\Delta t} \\
& + \frac{\sigma^2}{2} S^2 \left( \frac{V(S, t + \Delta t) - 2V(S - \Delta S, t + \Delta t) + V(S - 2\Delta S, t + \Delta t)}{2\Delta S^2} \right) \\
& + \frac{\sigma^2}{2} S^2 \left( \frac{V(S, t) - 2V(S - \Delta S, t) + V(S - 2\Delta S, t)}{2\Delta S^2} \right) \\
& + (r - D_0) S \left( \frac{3V(S, t + \Delta t) - 4V(S - \Delta S, t + \Delta t) + V(S - 2\Delta S, t + \Delta t)}{4\Delta S} \right) \quad (4.16) \\
& + (r - D_0) S \left( \frac{3V(S, t) - 4V(S - \Delta S, t) + V(S - 2\Delta S, t)}{4\Delta S} \right) \\
& - r \left( \frac{V(S, t + \Delta t) + V(S, t)}{2} \right) = 0
\end{aligned}$$

We call this discretization BC2. It remains accurate only to  $O(\Delta S)$ , but now the error is proportional to  $\partial^3 V / \partial S^3$ . A direct solution using a simple tridiagonal algorithm can no longer solve these discrete equations in a single pass. One can apply a more complex banded matrix solver algorithm, apply the tridiagonal solution iteratively (Wilmott et al., 1993), use an SOR solver, or apply one of the more powerful iterative solvers from the conjugate gradient family (see Chapter 3). This is only worthwhile if the new term (even if discretized with low accuracy) reduces the error caused by our imposition of boundary conditions. In the case of an option where linearity holds asymptotically as the boundary is moved to increasing larger prices, the extra term adds little accuracy. As will be demonstrated, however, it can greatly improve accuracy in cases where linearity does not hold.

We can use the matrix eigenvalue analysis introduced in Chapter 3 to investigate the effect of these various boundary conditions on the stability of the finite difference scheme. In Figure 4.7(a) we display the eigenvalues  $\lambda_i$  and corresponding amplification factors  $\Lambda_i$  for a second order central difference discretization of the Black-Scholes equation in spot price  $S$  coordinates. Three boundary conditions are shown, Neumann, the linearity condition BC1, and use of the pricing equation itself on the boundary, BC2. A grid of 51 points is used, with  $S_{\min} = 0$  and  $S_{\max} = 250$  ( $\Delta S = 5$ ). The interest



**Figure 4.7** Eigenvalues  $\lambda_i$  and corresponding amplification factors  $\Lambda_i$  for a second order central difference discretization of the Black-Scholes equation in spot price  $S$  coordinates for three boundary conditions: BC1 (dashed curves), BC2 (solid curves), and Neumann (dotted curves). (a) The 51 eigenvalues. (b) The amplification factors  $\lambda_i$  for explicit Euler time discretization. (c) Amplification factors for Crank-Nicolson time discretization. (d) Amplification factors for implicit Euler time discretization. The time step size is  $1/40$  year ( $T = 1$  and  $N = 40$ ). Other parameters used:  $I = 50$ ,  $T = 1$ ,  $D_0 = 0.025$ ,  $r = 0.05$ ,  $\sigma = 0.5$ ,  $S_{\min} = 0$ ,  $S_{\max} = 250$ ,  $(\Delta S = 5)$ .

rate, dividend yield, and volatility are respectively,  $r = 0.05$ ,  $D_0 = 0.025$ ,  $\sigma = 0.5$ . The eigenvalues are only slightly sensitive to the particular boundary condition chosen. The difference between BC1 and BC2 is particularly small. Therefore, the effect of boundary discretization on stability is also quite small. This insensitivity can be traced to the strong diffusion.

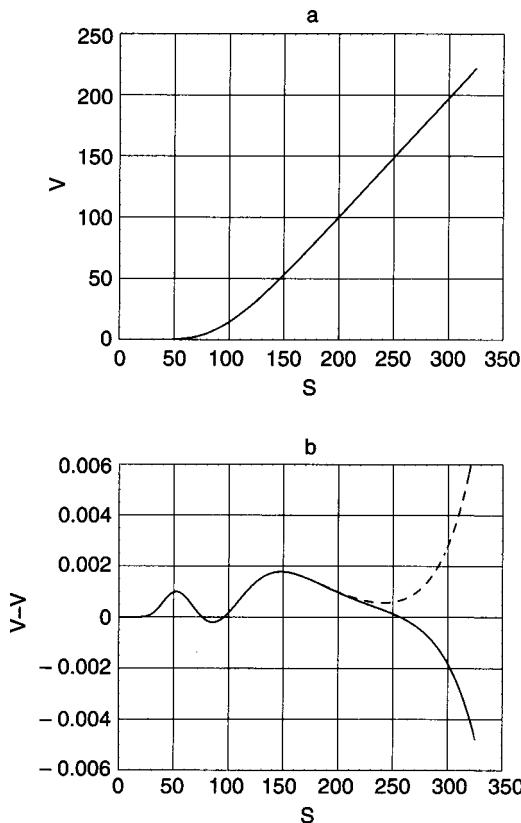
Figures 4.7(b-d) display the amplification factors for the explicit Euler, Crank-Nicolson, and implicit Euler time discretization, respectively with  $\Delta t = 1/40$  (i.e., 40 time steps per year). The explicit Euler scheme is unstable for these parameters. Almost half of the eigenmodes are unstable, and the particulars of the boundary condition make little difference. On the other hand, both the Crank-Nicolson and implicit Euler time discretizations are unconditionally stable, with the magnitude of all  $\Lambda$  bounded by 1 for both boundary conditions, although again the stability is little affected by the discretization of the boundary condition.

Figure 4.8(a) displays finite difference results for present option value for a call struck at  $K = 100$ . Figure 4.8(b) displays the error in present option value when using the linearity condition BC1 (solid curve) and when using BC2 (dashed curve).  $S_{\max} = 320$  and  $\Delta S = 2$ . The number of time steps is made very large ( $N = 5000$ ) so that the time discretization contribution to truncation error is negligible. Near the strike price, the total error is dominated by the truncation error of the discretized pricing equation in the interior region. However, near the  $S_{\max}$  boundary, error is mostly influenced by our artificial imposition of approximate boundary conditions. The errors are approximately equal in magnitude for both BC1 and BC2.

Table 4.1 displays the error in present option value for both at-the-money,  $S = K$  and well into-the-money  $S = 1.5K$  calls, for both BC1 and BC2 boundary conditions, as the boundary  $S_{\max}$  is moved farther from the strike.  $\Delta S$  is kept constant as  $S_{\max}$  is increased. The error for BC1 is large for small  $S_{\max}$  because the boundary is too close to the strike for linearity to hold. As  $S_{\max}$  is increased, the boundary condition-induced error rapidly decreases and total error approaches the lower limit set by the truncation error of the discretized pricing equation in the interior region. The boundary condition-induced error becomes negligible when  $S_{\max} / S \approx \exp(2\sigma\sqrt{T})$ . However, the addition of the volatility term to the boundary condition BC2 is never of benefit for this problem.

Now consider pricing a simple zero coupon bond under a Cox-Ingersoll-Ross interest rate model. The pricing equation is

$$\frac{\partial B}{\partial t} + \frac{1}{2}\sigma^2 r \frac{\partial^2 B}{\partial r^2} + (a - br) \frac{\partial B}{\partial r} - r B = 0 \quad (4.17)$$



**Figure 4.8** Finite difference results for a European call option. (a) Present option value versus spot price. (b) Error in present option value versus spot price using two different boundary conditions. BC1 (solid curve) and BC2 (dashed curve). Parameters are:  $D_0 = 0.025$ ,  $I = 175$ ,  $K = 100$ ,  $N = 5000$ ,  $r = 0.05$ ,  $\sigma = 0.35$ ,  $S_{\min} = 0$ ,  $S_{\max} = 350$ ,  $T = 1$ .

with  $B = 1$  at maturity,  $t = T$ . In this model, the interest rate cannot become negative if  $\sigma < \sqrt{2\alpha}$  and we can use  $r = 0$  as a lower boundary. The boundary condition at  $r = 0$  is simply

$$\frac{\partial B}{\partial t} + \alpha \frac{\partial B}{\partial r} = 0 \quad (4.18)$$

**Table 4.1 Call Option: Two Boundary Conditions**

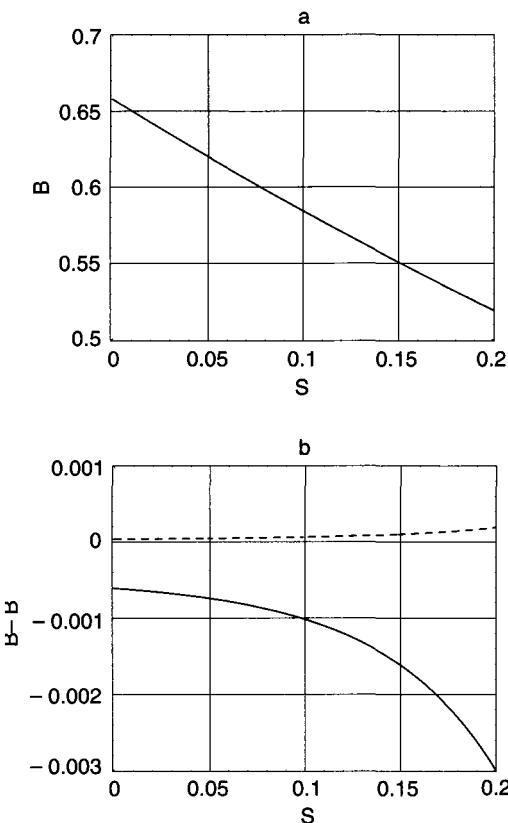
$S_{\max}$	$S = K$ BC1	$S = K$ BC2	$S = 1.5K$ BC1	$S = 1.5K$ BC2
150	-0.1214	0.1983	-2.4394	4.4041
175	-0.0063	0.0101	-0.2994	0.4936
200	-0.0001	0.0006	-0.0278	0.0473
225	0.0002	0.0002	-0.0008	0.0055
250	0.0002	0.0002	0.0016	0.0021
275	0.0002	0.0002	0.0018	0.0018
300	0.0002	0.0002	0.0018	0.0018
325	0.0002	0.0002	0.0018	0.0018
350	0.0002	0.0002	0.0018	0.0018

Error in present option value versus for two spot prices,  $S = K = 100$  and  $S = 1.5K = 150$  using boundary conditions BC1 and BC2 versus  $S_{\max}$ . Other parameters are:  $D_0 = 0.025$ ,  $K = 100$ ,  $N = 5000$ ,  $r = 0.05$ ,  $\sigma = 0.35$ ,  $T = 1$ .

The theoretical upper boundary condition  $B = 0$  as  $r \rightarrow \infty$  is of little practical use if we retain  $r$  as the coordinate. The grid would be orders of magnitude larger than necessary if we make  $r_{\max}$  large enough to apply  $B(r_{\max}) = 0$  with acceptable error. We could make a coordinate transformation, the subject of Chapter 5, to reduce the required size of the grid by effectively stretching the grid spacing  $\Delta r$ .

Alternatively, we can retain  $r$  as the coordinate and use a much smaller  $r_{\max}$  if we can find an appropriate boundary condition. Since no obvious boundary condition is suggested by financial arguments other than the foregoing  $B = 0$ , we can try both the linearity boundary condition BC1 and the use of the pricing equation itself BC2 on the boundary. We have an analytic result with which to compare for the CIR bond model, so we can determine whether either of these boundary conditions is accurate enough.

Figure 4.9(a) displays finite difference results for present bond value for a bond of  $T = 10$  year maturity valued under a CIR interest rate model. The interest rate model drift parameters are  $a = 0.04$ ,  $b = 0.8$ , and the volatility parameter is  $\sigma = 0.275$ .  $r_{\max} = 0.2$  and  $\Delta r = 0.001$ . Figure 4.9(b) displays the error in present bond value when using the linearity condition BC1 (solid curve) and when using BC2 (dashed curve). The number of time steps is again very large ( $N = 5000$ ) so that



**Figure 4.9** Finite difference results for a zero coupon bond under the CIR model. (a) Present bond value versus spot interest rate. (b) Error in present option bond versus spot interest rate using two different boundary conditions. BC1 (solid curve) and BC2 (dashed curve). Parameters are:  $a = 0.04$ ,  $b = 0.8$ ,  $I = 200$ ,  $N = 5000$ ,  $r_{\max} = 0.2$ ,  $\sigma = 0.275$ ,  $T = 10$ .

the time discretization contribution to error is negligible. For both boundary conditions, the total error is largest near the  $r_{\max}$  boundary, but extends throughout the solution domain. However, the magnitude of error for the boundary condition BC2, which includes the volatility term, is more than an order of magnitude smaller than that for BC1.

Table 4.2 displays the error in present value for the same bond at two spot interest rates,  $r = r^*$  and  $r = 1.5r^*$ , where  $r^* = a/b$  is the equilibrium interest rate, and for both boundary conditions as the

**Table 4.2 CIR Bond: Two Boundary Conditions**

$r_{\max}$	$r = r^*$ BC1	$r = r^*$ BC2	$r = 1.5r^*$ BC1	$r = 1.5r^*$ BC2
0.10	-0.005323	0.000338	-0.006067	0.000390
0.15	-0.002021	0.000123	-0.002315	0.000143
0.20	-0.000742	0.000043	-0.000854	0.000051
0.25	-0.000265	0.000015	-0.000307	0.000017
0.30	-0.000092	0.000005	-0.000107	0.000006
0.35	-0.000031	0.000002	-0.000036	0.000002
0.40	-0.000010	0.000001	-0.000011	0.000001
0.45	-0.000002	0.000000	-0.000003	0.000000
0.50	-0.000000	0.000000	-0.000000	0.000000

Error in present bond value for two spot interest rates,  $r = r^*$  and  $r = 1.5r^*$  using boundary conditions BC1 and BC2 versus  $r_{\max}$ . Parameters are:  $a = 0.04$ ,  $b = 0.8$ ,  $N = 5000$ ,  $\sigma = 0.275$ ,  $T = 10$ .

boundary  $r_{\max}$  is increased. ( $\Delta r$  is kept constant as  $r_{\max}$  is increased.) The error for BC2 is more than an order of magnitude smaller than that for the linearity condition BC1, for almost any size grid. The truncation error of the discretized pricing equation in the interior region is so small in this case that the total error does not reach an asymptotic limit (to the precision of the table) as  $r_{\max}$  is increased.

### Boundary Conditions in Multiple Dimensions

In one dimension, only three equations are required in addition to the payoff condition: a pricing equation for the interior of the solution domain, and two boundary conditions. In  $n$  dimensions, there are generally  $3^n$  required additional equations: a pricing equation for the interior of the solution domain and  $3^n - 1$  boundary conditions. For example, consider the pricing equation for a two-asset option for which the price processes are lognormal:

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{1}{2} \sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} + \frac{1}{2} \sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} \\ + (r - D_{01}) S_1 \frac{\partial V}{\partial S_1} + (r - D_{02}) S_2 \frac{\partial V}{\partial S_2} - rV = 0 \end{aligned} \quad (4.19)$$

In addition to the pricing equation, we must supply boundary conditions for two  $S_1$  boundaries, two  $S_2$  boundaries, and for each of the four vertices where these boundaries intersect. Conceivably, a different boundary condition may be applied to each of these regions. Sometimes, provisions of the financial contract under consideration will lead to simple and well-defined boundary conditions for some or all of the boundary regions (e.g.,  $V = 0$  or  $\partial V / \partial S_1 = 0$ ). Even if this is not the case for a particular boundary, then the linearity boundary condition described in the previous section can be easily generalized for multiple dimensions.

In applying a linearity boundary condition in multiple dimensions, there are several possible approaches. Eliminating all of the second order derivative terms from the pricing equation results in

$$\frac{\partial V}{\partial t} + (r - D_{01})S_1 \frac{\partial V}{\partial S_1} + (r - D_{02})S_2 \frac{\partial V}{\partial S_2} - rV = 0 \quad (4.20)$$

This equation would be discretized in the same way as its one-dimensional counterpart, Equation 4.13, using first order one-sided difference operators for the drift terms. These discretizations only couple the boundary to one interior grid point, and hence do not add to the complexity of the solver.

However, one need not be quite so restrictive. We can use a boundary condition that specifies only that option value be linear in the direction normal to a particular boundary. For example, on an  $S_2$  boundary we can apply the boundary condition

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{1}{2}\sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + \rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} + (r - D_{01})S_1 \frac{\partial V}{\partial S_1} \\ + (r - D_{02})S_2 \frac{\partial u}{\partial S_2} - rV = 0 \end{aligned} \quad (4.21)$$

This boundary condition says that near the  $S_2$  boundaries there is no diffusion in the  $S_2$  direction, normal to the boundary, but there may be in the  $S_1$  direction, parallel to the boundary. The correlation, or cross-derivative term may be deleted or retained. Either way, this equation can also be discretized using one-sided difference operators

for the terms containing S2 derivatives and standard central differencing for the terms containing S1 derivatives. At corner points, that is, at points which lie on both S1 and S2 boundaries, we can use

$$\frac{\partial V}{\partial t} + \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} + (r - D_{01}) S_1 \frac{\partial V}{\partial S_1} + (r - D_{02}) S_2 \frac{\partial V}{\partial S_2} - rV = 0 \quad (4.22)$$

where deleting the correlation term results in Equation 4.20. The use of these boundary conditions in multidimensional option problems is illustrated in several examples later in this chapter and in Chapter 6.

### **CONTINUOUS AND DISCRETE SAMPLING MODELS FOR PATH-DEPENDENT OPTIONS**

Pricing equations were derived for both continuously and discretely sampled path-dependent options in Chapter 2. As mentioned there, the underlying processes can only be observed discretely. Although not particularly common, it is possible for financial contracts to call for a sampling rate so high that it is practically modeled as continuous. Often, the continuous sampling assumption simply facilitates an analytic solution.

#### **Continuous Sampling**

As shown in Chapter 2, the continuous sampling model of path dependency introduces new convection terms to the pricing equation. For example, the pricing equation for a continuously sampled arithmetic Asian option can be written

$$\frac{\partial V}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} + (r - D_0) S \frac{\partial V}{\partial S} + S \frac{\partial V}{\partial g} - rV = 0 \quad (4.23)$$

where

$$g = \int_0^t S(\tau) d\tau$$

is the running time integral of spot price. An alternative formulation (Zvan, Forsyth, & Vetzal, 1998) is

$$\frac{\partial V}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} + (r - D_0) S \frac{\partial V}{\partial S} + \frac{(S - A)}{t} \frac{\partial V}{\partial A} - rV = 0 \quad (4.24)$$

where

$$A = \frac{1}{t} \int_0^t S(\tau) d\tau$$

is the running average price.

The numerical solution of such equations can be difficult. The difficulty arises because standard central differencing of the drift term can give rise to oscillations when the accompanying diffusion in the direction of the drift is small. The oscillations are not symptoms of numerical instability, rather they arise due to numerical dispersion. That is, Fourier components with different wavelengths drift across the finite difference grid at slightly different rates. The gradual separation of these components manifests in the development of oscillations. The departure of the numerical drift rate from the correct value was shown in Chapter 3 to be proportional to  $k^2 \Delta S^2$ , where  $k$  is the wavenumber of the Fourier mode. If the grid is very fine, then only components with very large wavenumbers (very short wavelengths) separate from the rest. But these components also decay rapidly if there is any diffusion, resulting in a smooth solution. On a coarse grid, or if diffusion is small or absent, longer wavelength components begin to separate from each other and do not decay rapidly enough to avoid the oscillations. This qualitative description of the phenomenon can be made quantitative and leads to well-defined criteria for determining when oscillations may arise, at least in one dimension (Zvan et al., 1998).

Several discretization techniques have been developed to deal with this situation. Upwind differencing is a standard technique in computational fluid dynamics, but it is only accurate to first order in  $\Delta g$ , and the leading order error term effectively introduces a large spurious numerical diffusion in the  $g$  direction. The oscillations are diminished by this numerical diffusion at the expense of accuracy.

A family of nonlinear algorithms generally described as TVD (total variation diminishing), also developed for computational fluid dynamics, can be used for continuously sampled path dependent option-pricing problems. In fact, they may be used for any drift-dominated pricing equation. TVD methods retain accuracy by introducing (locally) only as much numerical diffusion as is required to eliminate the introduction of the previously described oscillations. The nonlinear nature of the algorithm mixes Fourier components so that a rigorous analysis of their accuracy is difficult to obtain. The methods hold promise for the numerical solution of pricing equations that do not yield to the standard differencing techniques we have discussed. They are however, more complex than these standard differencing schemes, and they destroy the linearity of the numerical solution. That is, if  $V_i(S,g,t)$  is the numerical solution obtained for payoff  $F_i(S,g,T)$  where  $i = \{1,2\}$ , then the numerical solution for payoff  $F_1(S,g,T) + F_2(S,g,T)$  is not necessarily equal to  $V_1(S,g,t) + V_2(S,g,t)$  when a TVD algorithm is employed. Since complex option structures are often modeled as a sum of simpler ones, the nonlinearity of TVD may be a disadvantage. We will not discuss the implementation of TVD algorithms in this text.

Our simple description of the dispersive oscillations and their stabilization by diffusion seems to hold out little hope of solving pricing equations for continuously sampled path-dependent options with standard finite difference algorithms because the volatility corresponding to the path dependency is exactly zero. However, numerical experimentation shows that predictions of dispersive oscillations in multidimensional problems based on such one-dimensional arguments can be overly gloomy. Whether or not special algorithms are required depends strongly on the particular pricing equation being solved and parameter regime of interest.

Consider Equation 4.24. In this formulation, the drift term in the  $A$  direction becomes very large as  $t \rightarrow 0$  for  $S \neq A$ . While the singularity at  $t = 0$  can be avoided by assuming  $A = S$ , the drift term is certainly dominant in much of the solution domain. The results of solving Equation 4.24 using standard second order accurate central differencing in the  $S$  and  $A$  directions and Crank-Nicolson time differencing can be disastrous (Zvan et al., 1998), with oscillations quickly overwhelming the solution.

Methods for solving drift-dominated pricing equations typically involve the addition of numerical diffusion, either generally

(upwind discretization) or locally (TVD algorithms). Therefore, an alternative technique worth exploring is to add an artificial volatility term to the pricing equation itself

$$\frac{\partial V}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} + (r - D_0) S \frac{\partial V}{\partial S} + \frac{\sigma_g^2}{2} g^2 \frac{\partial^2 V}{\partial g^2} + S \frac{\partial V}{\partial g} - rV = 0 \quad (4.25)$$

and then use a standard discretization. Extrapolation of the finite  $\sigma_g$  option values to  $\sigma_g = 0$ , if such an extrapolation can be done reasonably, then yields the solution sought. One can think of this technique as explicitly placing the  $g$ -direction diffusion into the pricing equation itself, rather than burying it in the details of numerical solution algorithm.

Table 4.3 displays numerical results for a continuously sampled fixed strike arithmetically averaged Asian call option. The option expires in one year and is struck at  $K = 100$ , with payoff  $V(S, g, T) = \max(g - K, 0)$ . The market parameters are  $D_0 = 0.0$ ,  $r = 0.10$ , and  $\sigma = 0.40$ . Second order central discretization in the price coordinates and Crank-Nicolson discretization in time are used. A biconjugate gradient-stabilized solver is used to solve the multidimensional discretized equations. For each of three spot prices  $S = \{95, 100, 105\}$ , finite difference present option value is shown for five values of artificial volatility  $\sigma_g^2$  ranging from 0.01 to 0. Evidently, the error induced by the artificial volatility  $\sigma_g$  scales as  $\sigma_g^2$ .

The values shown for  $\sigma_g^2 = 0$  are the results of actual finite difference calculations. These calculated values agree exactly with extrapolations from the finite  $\sigma_g^2 = 0$  data. Perhaps surprisingly, the option value surface  $V(S, g, t = 0)$  is smooth and oscillation-free. A reasonable conjecture is that while dispersion may attempt to generate oscillations in the  $g$  direction, they are localized and have a strong variation in the  $S$  direction. They are therefore subject to diffusive stabilization by the actual volatility  $\sigma$ .

One should not presume that Equation 4.25 can always be solved with  $\sigma_g^2 = 0$ . It is certain that there are other parameter regimes for which the numerical solution of this equation using standard central differencing will not be successful. For example, numerical experiments when both  $\sigma$  and  $\sigma_g$  are small show evidence of dispersive oscillations, especially in the greeks. In such cases, the extrapolation of

**Table 4.3 Continuously Sampled Fixed Strike Asian Call**

Varying Artificial Volatility		
$S$	$\sigma_g^2$	$V(S)$
95	0.01000	8.6150
	0.00100	8.3903
	0.00010	8.3681
	0.00001	8.3668
	0	8.3665
100	0.01000	11.3762
	0.00100	11.1439
	0.00010	11.1208
	0.00001	11.1193
	0	11.1191
105	0.01000	14.5154
	0.00100	14.2858
	0.00010	14.2630
	0.00001	14.2615
	0	14.2612

Results for a continuously sampled fixed strike arithmetically averaged Asian call option using an artificial volatility of the running average  $g$ . The option expires in one year and is struck at  $K = 100$ . First column is spot price  $S$ , the second is the square of artificial volatility  $\sigma_g^2$ , the third column is the finite difference result with  $J = 100$  grid points in the  $g$  dimension. Parameters are  $D_0 = 0.0$ ,  $r = 0.10$ ,  $\sigma = 0.40$ ,  $S_{\min} = 0$ ,  $S_{\max} = 250$ .

finite  $\sigma_g$  numerical results to  $\sigma_g = 0$  may provide the desired result. If not, then advanced discretization techniques mentioned earlier may be required.

### Discrete Sampling

For many path-dependent options, continuous sampling is an approximation. For accurate results, a discrete sampling model is required. The difference between discretely and continuously sampled models for lookback options, for example, scales as  $\sqrt{\delta t_s}$ , where

$\Delta t_s$  is the sampling interval. A continuously sampled model of an option sampled as frequently as hourly, can still yield unacceptably large errors. The departure of discretely sampled Asian option models from their continuously sampled counterparts scales more mildly, as  $\Delta t_s^{\frac{1}{2}}$ , but a continuous sampling model of an option sampled daily can be quite inaccurate. Finally, for those options that are sampled so frequently that a discrete sampling model at the actual sampling rate would become impractical, numerical results from lower sampling rate models can be used to extrapolate to the desired model.

As discussed in Chapter 2, convection terms in the path-variable direction are replaced by jump conditions for discretely sampled path-dependent options. For a discretely sampled fixed strike arithmetic Asian call option in the running sum formulation, one solves the one-dimensional Black-Scholes equation subject to the expiration condition

$$V(S, g, T) = \max(g/N_s - K, 0) \quad (4.26)$$

where  $N_s$  is the number of samples, the path variable  $g$  is the running sum of spot prices  $S$

$$g(t_k) = \sum_{i=1}^{i=k} S(t_i)$$

and the jump condition at sampling times  $t_i$

$$V(S, g, t_i^-) = V(S, g + S, t_i) \quad (4.27)$$

where  $t_i$  is the time of the  $i^{th}$  sample, and  $t_i^-$  is just before the  $i^{th}$  sample. The variable  $g$  appears only as a parameter in the pricing equation, and in the expiration and boundary conditions.

In the running average formulation, the expiration condition is

$$V(S, A, T) = \max(A - K, 0) \quad (4.28)$$

where the path variable  $A$  is the running average of spot prices  $S$

$$A(t_k) = \frac{\sum_{i=1}^{i=k} S(t_i)}{k}$$

and the jump condition at sampling times  $t_i$

$$V(S, A, t_i^-) = V(S, A + \frac{S - A}{i}, t_i) \quad (4.29)$$

In either formulation, the jump conditions are implemented numerically as interpolations. In this section we will use the running sum formulation of the problem.

The current value of a discretely sampled Asian option varies linearly with the sampling interval  $\delta t_s$ . This gives us an alternative method of valuing continuously sampled Asian options by simply extrapolating to the  $\delta t_s = 0$  limit. Table 4.4 displays numerical results for a discretely sampled fixed strike arithmetically averaged Asian call option that verify the linear dependence on  $\delta t_s$ . The option expires in one year and is struck at  $K = 100$ . The market parameters are  $D_0 = 0.0$ ,  $r = 0.10$ , and  $\sigma = 0.40$ . The first column of the table is spot price  $S$  and the second is the sampling interval  $\delta t_s$ . Shown in the third column is the finite difference present option value,  $V_{fd}(S)$ . (Finite difference pricing of discretely sampled Asian options is described in more detail in Chapter 6. Here we simply analyze the results of the computations.) For each spot price  $S$ , present option value is shown for seven sampling intervals ranging from 0.1 year to 0.001 year. The rows of Table 4.4 corresponding to  $\delta t_s = 0$  contain the results of fitting a line to the seven finite  $\delta t_s$  calculations and extrapolating to  $\delta t_s = 0$ ,  $V_{fd0}(S)$ . The fourth column is a geometrical conditioning approximation  $V_{gc}(S)$  (Curran, 1995). The finite difference and approximate analytical option values are in good agreement, and the agreement improves as the sampling interval decreases. Finally, the last column is the  $\delta t_s$  finite difference result less the finite difference result extrapolated to  $\delta t_s = 0$ . The difference in option value varies linearly with  $\delta t_s$ .

The Asian option values obtained by extrapolation of a discretely sampled model to  $\delta t_s$  to 0 agree fairly well with those obtained by solving the continuously sampled model Equation 4.23. In

**Table 4.4 Discretely Sampled Fixed Strike Asian Call**

Varying Sampling Rate					
$S$	$\delta t_s$	$V_{fd}(S)$	$V_{gc}(S)$	$V_{fd}(S) - V_{fd0}(S)$	
95	0.1000	9.2149	9.2197	0.8509	
	0.0400	8.6974	8.7053	0.3334	
	0.0200	8.5283	8.5340	0.1643	
	0.0080	8.4304	8.4314	0.0664	
	0.0040	8.3972	8.3972	0.0332	
	0.0020	8.3804	8.3801	0.0164	
	0.0010	8.3719	8.3715	0.0079	
	0	8.3640	—	—	
100	0.1000	12.0348	12.0390	0.9133	
	0.0400	11.4803	11.4881	0.3589	
	0.0200	11.2982	11.3043	0.1768	
	0.0080	11.1929	11.1940	0.0715	
	0.0040	11.1573	11.1572	0.0358	
	0.0020	11.1392	11.1388	0.0177	
	0.0010	11.1300	11.1296	0.0086	
	0	11.1215	—	—	
105	0.1000	15.2168	15.2202	0.9487	
	0.0400	14.6415	14.6483	0.3734	
	0.0200	14.4519	14.4575	0.1838	
	0.0080	14.3424	14.3430	0.0743	
	0.0040	14.3054	14.3048	0.0373	
	0.0020	14.2866	14.2857	0.0185	
	0.0010	14.2771	14.2762	0.0090	
	0	14.2681	—	—	

Results for a discretely sampled fixed strike arithmetically averaged Asian call option. The option expires in one year and is struck at  $K = 100$ . First column is spot price  $S$ , the second is the sampling interval  $\delta t_s$ , the third is the finite difference result, the fourth is Curran's geometrical conditioning result, and the last column is the finite difference result sampled at  $\delta t_s$  less the finite difference result extrapolated to  $\delta t_s = 0$ . Parameters are:  $D_0 = 0.0$ ,  $r = 0.10$ ,  $\sigma = 0.40$ .

a case where solution of this equation is not possible, then we expect agreement with the solution of Equation 4.25 extrapolated to  $\sigma_g = 0$ . Residual differences in this example may be the result of boundary positions, and time step and grid size differences.

Table 4.5 displays numerical results for a discretely sampled fixed strike lookback put option. We solve the one-dimensional Black-Scholes equation subject to the expiration condition

$$V(S, g, T) = \max(K - g, 0) \quad (4.30)$$

and to the jump condition (Wilmott et al., 1993) at sampling times  $t_i$

$$V(S, g, t_i^-) = V(S, \min(g, S), t_i) \quad (4.31)$$

The data in Table 4.5 shows the scaling of lookback option values with  $\sqrt{\delta t_s}$ . The option expires in one year and is struck at  $K = 100$ .

**Table 4.5 Discretely Sampled Fixed Strike LookBack Put**

Varying Sampling Rate				
$S$	$\delta t_s$	$V_{fd}(S)$	$V_{fd}(S) - V_a(S)$	$(V_{fd}(S) - V_a(S)) / \sqrt{\delta t_s}$
95	0.0200	13.0458	-0.8810	-6.2296
	0.0080	13.3389	-0.5880	-6.5739
	0.0040	13.4953	-0.4315	-6.8230
	0.0020	13.6151	-0.3118	-6.9713
	0.0008	13.7360	-0.1908	-6.7462
	0.0004	13.8074	-0.1194	-5.9704
	cont	13.9268	—	—
100	0.0200	9.7621	-0.6137	-4.3392
	0.0080	9.9628	-0.4129	-4.6164
	0.0040	10.0717	-0.3040	-4.8065
	0.0020	10.1555	-0.2203	-4.9255
	0.0008	10.2403	-0.1354	-4.7880
	0.0004	10.2906	-0.0852	-4.2579
	cont	10.3757	—	—
105	0.0200	7.6416	-0.4031	-2.8500
	0.0080	7.7720	-0.2727	-3.0487
	0.0040	7.8433	-0.2014	-3.1841
	0.0020	7.8984	-0.1463	-3.2718
	0.0008	7.9544	-0.0903	-3.1925
	0.0004	7.9877	-0.0570	-2.8493
	cont	8.0447	—	—

Results for a discretely sampled fixed strike lookback put option. The option expires in one year and is struck at  $K = 100$ . First column is spot price  $S$ , the second is the sampling interval  $\delta t_s$ , the third is the finite difference result, the fourth is the finite difference result sampled at  $\delta t_s$  less the analytic result for continuous sampling. The fifth column is the fourth column divided by  $\sqrt{\delta t_s}$ . Parameters are:  $D_0 = 0.02$ ,  $r = 0.05$ ,  $\sigma = 0.15$ ,  $S_{\min} = 0$ ,  $S_{\max} = 150$ ,  $g_{\min} = 0$ ,  $g_{\max} = 120$ ,  $I = 600$ ,  $J = 120$ ,  $g_0 = 95$ .

The market parameters are  $D_0 = 0.02$ ,  $r = 0.05$ , and  $\sigma = 0.15$ . The minimum observed spot price at the time of pricing the option is  $g_0 = 95$ . The first column of the table is spot price  $S$  and the second is the sampling interval  $\delta t_s$ . In the third column is the finite difference present option value,  $V_{\text{fd}}(S)$ . For each spot price  $S$ , present option value is shown for seven sampling intervals ranging from 0.02 year to 0.0004 year. The rows of the table labeled “cont” are the corresponding analytical results for continuous sampling. The fourth column is the difference between the discretely sampled finite results and the continuous analytic formula. Finally in the fifth column is the difference in the fourth column divided by  $\sqrt{\delta t_s}$ . The fact that the numbers in this column are nearly constant for each spot price demonstrates the scaling with  $\sqrt{\delta t_s}$ .

## PERFORMANCE OF SOLVERS FOR MULTIDIMENSIONAL PROBLEMS

In this section, we demonstrate the convergence behavior of two- and three-asset American and European basket call option models, and show the relative performance of several solvers for each model. The PDEs being solved are, in the case of the two-asset basket

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{1}{2} \sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + \frac{1}{2} \sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} + \rho_{12} \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} \\ + (r - D_{01}) S_1 \frac{\partial V}{\partial S_1} + (r - D_{02}) S_2 \frac{\partial V}{\partial S_2} - rV = 0 \end{aligned} \quad (4.32)$$

and for the three-asset basket

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{1}{2} \sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + \frac{1}{2} \sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} + \frac{1}{2} \sigma_3^2 S_3^2 \frac{\partial^2 V}{\partial S_3^2} \\ + \rho_{12} \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} + \rho_{13} \sigma_1 \sigma_3 S_1 S_3 \frac{\partial^2 V}{\partial S_1 \partial S_3} \\ + \rho_{23} \sigma_2 \sigma_3 S_2 S_3 \frac{\partial^2 V}{\partial S_2 \partial S_3} + (r - D_{01}) S_1 \frac{\partial V}{\partial S_1} \\ + (r - D_{02}) S_2 \frac{\partial V}{\partial S_2} + (r - D_{03}) S_3 \frac{\partial V}{\partial S_3} - rV = 0 \end{aligned} \quad (4.33)$$

We use the BC1 boundary conditions discussed in a previous section for all boundaries in all the computations described in this section. The correlation terms are retained in the boundary condition equations. These boundary conditions work remarkably well for both European and American basket options. Again drawing on material presented earlier in this chapter, we use for the expiration conditions the cell-averaged value of the two-asset or three-asset basket call payoff functions:

$$F(S_1, S_2) = \max(n_1 S_1 + n_2 S_2 - K, 0) \quad (4.34)$$

$$F(S_1, S_2, S_3) = \max(n_1 S_1 + n_2 S_2 + n_3 S_3 - K, 0) \quad (4.35)$$

Table 4.6 demonstrates the convergence behavior of the two- and three-asset basket call option finite difference models as the grid size and number of time steps is increased. Theoretically, the finite difference truncation error scales quadratically with the inverse of grid size and with the number of time steps. The third column of the table shows the present option value at the spot prices  $S_1=S_2=100$  (two-asset) and  $S_1=S_2=S_3=100$  (three-asset) for various grid sizes. We fit these data with a polynomial in the inverse of the grid size. Only the constant and second order terms are significant, demonstrating the predicted quadratic scaling. The constant term is the extrapolated infinite grid size value,  $V_{ext}$ . This value is shown for both two- and three-asset options. In the fourth column, we display the truncation error itself,  $V - V_{ext}$ . The nearly perfect inverse square scaling of the error with grid size is evident. Without the cell-averaging of the expiration condition, this nearly perfect scaling is not obtained, and extrapolation to the continuous result cannot be reliably done. For both options, we also show a semianalytic result obtained by moment-matching techniques. In this method, one approximates the distribution of the sum of lognormals (which itself is not lognormal), with a lognormal distribution chosen to have the correct zeroth and second moments (Tavella, 1997). The moment matching analytic approximation agrees with the extrapolated finite difference result to about 1%. A biconjugate gradient stabilized (BCGST) solver was used in all calculations.

**Table 4.6 Basket Call Option Convergence**

(Grid Size) × Time Steps		$V$	$V - V_{\text{ext}}$
2D	(25 × 25) × 5	4.6189	0.2653
	(50 × 50) × 10	4.4299	0.0764
	(100 × 100) × 20	4.3733	0.0198
	(200 × 200) × 40	4.3585	0.0050
	extrapolated	4.3535	—
	moment matching	4.4023	0.0443
3D	(20 × 20 × 20) × 10	5.5337	0.0580
	(30 × 30 × 30) × 15	5.5012	0.0255
	(40 × 40 × 40) × 20	5.4903	0.0145
	(50 × 50 × 50) × 25	5.4850	0.0092
	extrapolated	5.4758	—
	moment matching	5.4868	0.0110

Convergence of two- and three-asset European basket call option calculations as a function of grid size and time steps.  $V$  is the present option value  $V(S_1 = 100, S_2 = 100)$  (2D) or  $V(S_1 = 100, S_2 = 100, S_3 = 100)$  (3D).  $V_{\text{ext}}$  is the value extrapolated to infinite grid size and infinite time steps from the three finite grid-size values. The rightmost column shows the convergence of the finite grid size calculations to this value as grid size is increased. The parameters for the two-asset basket are:  $\sigma_1 = 0.25$ ,  $\sigma_2 = 0.35$ ,  $p_{12} = -0.65$ ,  $r = 0.045$ ,  $D_{01} = 0.05$ ,  $D_{02} = 0.07$ ,  $T = 1$ ,  $n_1 = 0.58$ ,  $n_2 = 0.42$ ,  $K = 100$ ,  $S_{1\min} = 0$ ,  $S_{1\max} = 300$ ,  $S_{2\min} = 0$ ,  $S_{2\max} = 400$ . The additional or modified parameters for the three-asset basket are:  $\sigma_3 = 0.2$ ,  $p_{23} = 0.5$ ,  $p_{31} = 0.25$ ,  $D_{03} = 0.04$ ,  $n_1 = 0.38$ ,  $n_2 = 0.22$ ,  $n_3 = 0.40$ ,  $S_{3\min} = 0$ ,  $S_{3\max} = 300$ .

For the American version of the options, the constraint, for example,  $V(S1, S2, t) \sim F(S1, S2)$  must be continuously enforced. The simplest approach is to advance the solution using a linear solver, and then simply impose the American early exercise constraint in a Draconian fashion. In this case, the solution with which one is left is only correct to first order in the time step. It does not achieve the second order accuracy of the Crank-Nicolson method because the resulting approximate solution is not a self-consistent solution to the linear complementarity problem. Alternatively, one can advance the solution using the successive overrelaxation (SOR) iterative solver at each time step. This yields a self-consistent solution to the linear complementarity problem, but SOR is not always the most efficient solver for a large multidimensional system. SOR depends for good convergence on the matrix being diagonally dominant. In solving

time-dependent problems, the time difference  $(V(S_1, S_2, t + \Delta t) - V(S_1, S_2, t)) / \Delta t$  appears along the diagonal, so that if the time step is small enough, SOR will converge. If the time step is large, however, SOR may not converge at all.

A more robust approach for the multidimensional American option problem is to use an efficient solver such as BCGST, apply the early exercise constraint, and then use this approximate solution as the initial guess for an SOR solver. Most of the work is done by the BCGST solver and the application of the early exercise constraints. The SOR solver finetunes the solution to make it self-consistent and second order accurate in time.

Table 4.7 displays the performance of five solvers for European and American basket call option models. For all two-asset options, the grids were 100 x 100 with 20 time steps; while for all three-asset options, the grids were 40 x 40 x 40 with 20 time steps. For the European option models, at every time step, the iterative solvers have a convergence criterion of

$$\|Ax - b\| \leq \text{tol} (\|A\| \|x\| + \|b\|) \quad (4.36)$$

where  $\text{tol} = 10^{-7}$ . For the American models, the initial iterative solver pass has  $\text{tol} = 10^{-4}$ ; then the early exercise constraint is applied, and finally an SOR solver is applied with  $\text{tol} = 10^{-7}$ . The third column of the table gives the CPU time for the model relative to the European option using an SOR solver. The fourth column gives the average number of iterations required per time step. For the American models, two iteration counts are given. The first is for the initial iterative solver pass, the second for the final SOR refinement.

The conjugate gradient (CG) solver is not guaranteed to converge and, in fact, does not for the European options. For the American options, the much larger tolerance allowed the CG solver prevented it from diverging. Biconjugate gradient stabilized (BCGST) solvers converged for all models. Because each iteration of BCGST is essentially equivalent to two of CG, the iteration count is about half of that for CG, while the CPU time is approximately the same (where CG converges). The quasi-minimum residual (QMR) solvers also converged for all models but were consistently less efficient.

**Table 4.7 Basket Call Option Solver Performance**

Type	Solver	Relative CPU Time	Ave Iter/ Time Step
2D European $(100 \times 100) \times 20$	SOR	1.0	127.2
	CG	—	—
	BCGST	0.38	22.7
	BCGST-ILU	0.18	5.0
	QMR	0.80	36.9
2D American $(100 \times 100) \times 20$	SOR	0.43	50.6
	CG	0.26	5.2/24.6
	BCGST	0.26	2.7/25.0
	BCGST-ILU	0.36	2.0/33.8
	QMR	0.34	5.1/25.2
3D European $(40 \times 40 \times 40) \times 20$	SOR	1.0	43.2
	CG	—	—
	BCGST	0.68	14.5
	BCGST-ILU	0.25	3.0
	QMR	2.34	11.7
3D American $(40 \times 40 \times 40) \times 20$	SOR	0.63	25.4
	CG	0.40	3.6/12.1
	BCGST	0.40	2.2/11.8
	BCGST-ILU	0.62	2.0/15.0
	QMR	0.87	3.5/11.8

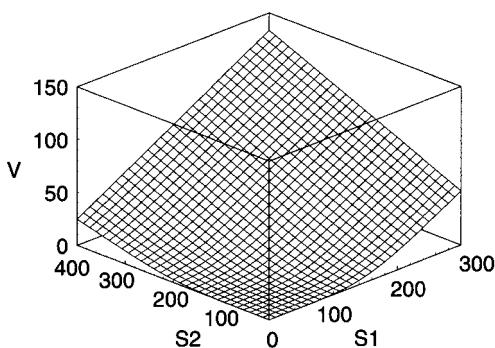
Performance of various solvers on the two- and three-asset American and European basket call option model. Parameters for the two-asset basket are:  $\sigma_1 = 0.25$ ,  $\sigma_2 = 0.35$ ,  $\rho_{12} = -0.65$ ,  $r = 0.045$ ,  $D_{01} = 0.05$ ,  $D_{02} = 0.07$ ,  $T = 1$ ,  $n_1 = 0.58$ ,  $n_2 = 0.42$ ,  $K = 100$ ,  $S_{1\min} = 0$ ,  $S_{1\max} = 300$ ,  $S_{2\min} = 0$ ,  $S_{2\max} = 400$ . The additional or modified parameters for the three-asset basket are:  $\sigma_3 = 0.2$ ,  $\rho_{23} = 0.5$ ,  $\rho_{31} = 0.25$ ,  $D_{03} = 0.04$ ,  $n_1 = 0.38$ ,  $n_2 = 0.22$ ,  $n_3 = 0.40$ ,  $S_{3\min} = 0$ ,  $S_{3\max} = 300$ .

The solver labeled BCGST-ILU is a biconjugate gradient stabilized solver but with a ILU (Incomplete LU) preconditioner. A complete LU decomposition is essentially equivalent to inverting the matrix, and hence is not feasible, because a dense preconditioning matrix is produced. The ILU decomposition used here neglects any elements that are zero in the original matrix. The ILU preconditioner reduced the average number of BCGST iterations required per time step by about a factor of five for the European options. The overall CPU time savings is a bit more than a factor of two because the preconditioner itself requires substantial computation. For the

American options, the bare BCGST solver only used about two iterations per time step to reach the greatly relaxed tolerance. Since the iteration count cannot be reduced below one per time step, the ILU preconditioner is essentially wasted arithmetic for the American option. The bare SOR solvers converged, but took many more iterations and more CPU time than the CG or BCGST solvers. Convergence of SOR was slowed by the relatively large time step compared with the fine price discretization. Had the price grids been made much finer, SOR would not have converged at all. Although not discussed here, Alternating Direction Implicit (ADI) solvers can be fast. The correlation terms must be treated properly (Craig & Sneyd, 1990).

Figure 4.10 shows the present value of the three-asset basket call option as a function of  $S_1$  and  $S_2$  on a slice along the  $S_3 = 100$  plane.

For simplicity, all the numerical results discussed in this section have been for uniformly spaced grids. Substantial increases in accuracy for a given grid size, and conversely, substantial decreases in grid size for a given accuracy can often be obtained through the use of coordinate transformations. These are discussed in Chapter 5. Coordinate transformations may also flatten the eigenvalue spectrum of the matrix representing the discretized pricing equation, resulting in improved solver convergence.



**Figure 4.10** Present value of a European three-asset basket call option. Shown is a slice through the three-dimensional data on the plane  $S_3 = 100$ . The parameters are those in the note for Table 4.6.

**NUMERICAL SOLUTION OF PIDEs:  
JUMP-DIFFUSION AND PURE JUMP MODELS**

As shown in Chapter 2, pricing equations include convolution integrals when the underlying process include jumps, for example, the pricing equations become partial-integro-differentiation equations (PIDEs). As a specific example, consider Merton's jump diffusion model (1976) for the underlying process:

$$dS/S = (\mu - \lambda k) dt + \sigma dW + (\eta - 1) dU \quad (4.37)$$

where  $dW$  is a standard Brownian motion and  $dU$  is a Poisson process of intensity  $\lambda$ . That is,

$$dU = \begin{cases} 0 & \text{with probability } 1 - \lambda dt \\ 1 & \text{with probability } \lambda dt \end{cases}$$

We take  $\eta$  to be a lognormally distributed jump amplitude with probability density

$$G(\eta) = \frac{\exp\left(-\frac{1}{2}\left(\frac{\log(\eta)}{\delta}\right)^2\right)}{(2\pi)^{\frac{1}{2}}\delta\eta} \quad (4.38)$$

where  $k$  is the expectation  $E(\eta - 1)$  and is given by

$$k = E(\eta - 1) = \exp(\delta^2/2) - 1 \quad (4.39)$$

The Poisson and Brownian motion processes are assumed to be uncorrelated. When a jump occurs, the asset price jumps from  $S$  to  $\eta S$ . The pricing equation is given by

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r - D_0 - \lambda k)S \frac{\partial V}{\partial S} - rV \\ + \lambda \left( \int_0^\infty V(S\eta) G(\eta) d\eta - V \right) = 0 \end{aligned} \quad (4.40)$$

This may be alternatively written as

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r - D_0 - \lambda k) S \frac{\partial V}{\partial S} - rV + \lambda \left( \int_{-\infty}^{\infty} V(S e^x) G(x) dx - V \right) = 0 \quad (4.41)$$

with

$$G(x) = \frac{\exp\left(-\frac{1}{2}\left(\frac{x}{\delta}\right)^2\right)}{(2\pi)^{\frac{1}{2}}\delta} \quad (4.42)$$

The convolution integral requires option value at the postjump spot price  $V(S e^x)$  and the jump size probability distribution  $G(x)$ .  $V$  is known on the finite difference grid, so that the value  $V(S e^x)$  can be obtained by interpolation and/or extrapolation. Usually the jump probability distribution  $G(x)$  will be an analytic function, and its value can be obtained for any  $x$  by simply evaluating a formula. However, since this evaluation will be done a very large number of times and may be complex, it is computationally more efficient to compute  $G(x)$  on a different  $x$  grid and to evaluate it by interpolation also. The  $x$  grid must chosen to resolve the variation of  $G(x)$  with sufficient detail. Once the postjump option value and jump size distribution are known, the numerical integration over  $x$  may be implemented numerically in any number of ways (e.g., by Gaussian quadrature or by Newton-Cotes type formulas) (Press et al., 1986). Despite the myriad possibilities, a simple trapezoidal rule integration over the  $x$  grid is often sufficient.

Another issue in the evaluation of the convolution integrals is how to approximate postjump option values that lie off the finite difference grid (e.g.,  $x < \log(S_{\min}/S)$  or  $x > \log(S_{\max}/S)$  in this example). If there is an analytic approximation or limiting form, that might be used. If not, then one must resort to some extrapolation. As has been noted, often option values are linear near the grid boundaries, and certainly linear extrapolation from the last two grid points or from more than two grid points in the neighborhood of the boundary is the easiest method to choose. It is not always appropriate. A simple case in

point is when we write our pricing equation in log-price coordinates. In this case, an option value that is linear in spot price would be exponential in the log of spot price so that an exponential extrapolation would be correct. In any case, the extrapolation technique chosen must be consistent with the boundary conditions of the pricing equation.

Most often one will use a time differencing scheme, such as Crank-Nicolson, that is at least partially implicit. This brings stability with the cost of coupling adjacent unknown option values on the finite difference grid. In the case of implicit time differencing, the convolution integral in the pricing equation introduces a significant complication for the numerical solution of the pricing equation. The partial differential operators in the pricing equation couple only closely neighboring points on the grid. In fact, for the typical second order central differencing scheme, only nearest neighbors are coupled, leading to sparse systems of discrete equations that are relatively easy to solve. However, the convolution integral couples grid points over an extended range of 5; in this example, a range of order  $5 \exp(30)$ . For typical parameters, every grid point could be coupled to tens or perhaps in extreme cases even hundreds of neighboring points, rendering the discrete system of equations dense, and hence much more difficult to solve.

The most elementary way of dealing with this problem is called "lagging." If we write the pricing equation as

$$\frac{\partial V}{\partial t} + LV + \lambda \left( \int_{-\infty}^{\infty} V(Se^x) G(x) dx - V \right) = 0 \quad (4.43)$$

where

$$LV = \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r - D_0 - \lambda k) S \frac{\partial V}{\partial S} - rV \quad (4.44)$$

then we can use Crank-Nicolson time differencing, for example, in the PDE portion of the equation, and just evaluate the integral term at the "old" (lagged) time step:

$$\frac{V^{n+1} - V^n}{\Delta t} + L \frac{V^{n+1} + V^n}{2} + \lambda \left( \int_{-\infty}^{\infty} V^n(Se^x) G(x) dx - V^n \right) = 0 \quad (4.45)$$

Now the unknown option values at the new time step  $V^{n+1}$  are coupled only by the partial differential operators, leading to a sparse system of discrete equations. The convolution term couples  $V$  over a wide range of  $S$ , but these are all known values  $V^n$ .

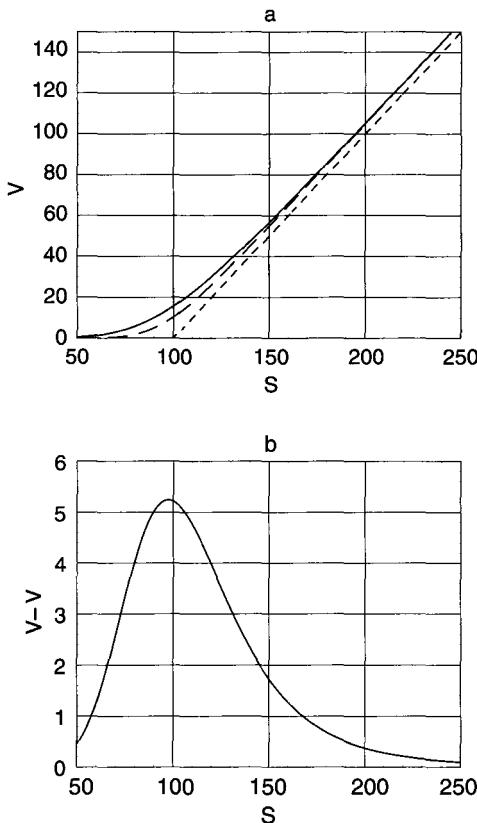
The drawback to this simple approach is obvious. The convolution term is centered at time  $t^n$ , while the rest of the discrete system has been carefully centered at  $t^{n+\frac{1}{2}}$ . Hence the accuracy falls to  $O(\Delta t)$  from  $O(\Delta t^2)$ . If we are using a first order accurate time differencing scheme, such as implicit Euler on the PDE portion of the equation, then no harm has been done. Most often, however, we would like to take advantage of the greater accuracy of Crank-Nicolson, or some other second order method.

To restore the discrete equations to second order accuracy in  $\Delta t$ , we must also center the convolution term at  $t^{n+\frac{1}{2}}$ . To avoid dense systems of discrete equations, we can adopt an iterative approach. We write the time-discretized pricing equation as

$$\begin{aligned} & \frac{V^{m+1} - V^n}{\Delta t} + L \frac{V^{m+1} + V^n}{2} \\ & + \lambda \left( \int_{-\infty}^{\infty} \frac{V^m (Se^x) + V^n (Se^x)}{2} G(x) dx - \frac{V^m + V^n}{2} \right) = 0 \end{aligned} \quad (4.46)$$

At the beginning of each time step, the iteration begins with  $V^m = V^n$  in the convolution term, then proceeds by solving for a new  $V^{m+1}$ , and substituting the new  $V^{m+1}$  for  $V^m$  in the convolution term. Iteration proceeds until a convergence criterion is met, at which point  $V^{m+1} \approx V^{n+1}$ , and a new time step is begun.

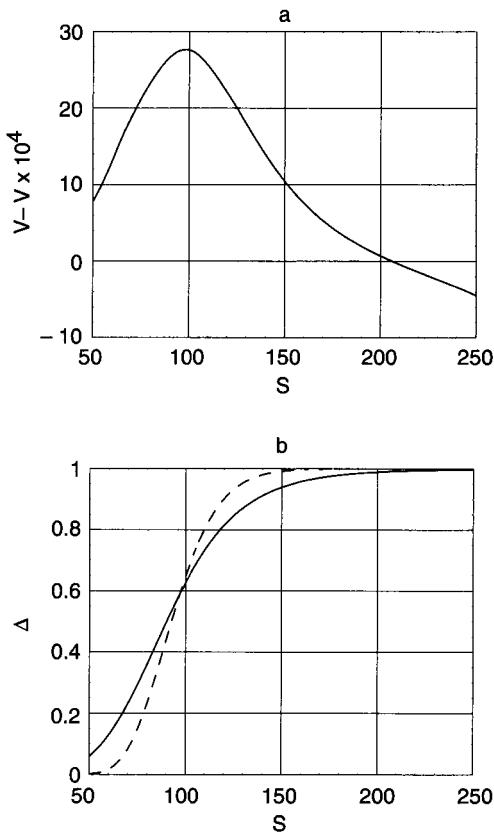
Figures 4.11 and 4.12 display numerical results for a European call option struck at  $K = 100$  and with expiration in  $T_{\max} = 1$  year. The volatility is  $\sigma = 0.2$ , while the coefficients of the jump process are  $\lambda = 2$  and  $\delta = 0.2$ . In Figure 4.11(a), the solid curve is the present option value under the jump-diffusion process, the dashed curve is results from setting the jump intensity  $\lambda = 0$  (i.e., the pure diffusion result), while the dotted curve is the payoff condition. The difference between the jump-diffusion process and the pure diffusion process is shown in Figure 4.11(b), and is quite substantial. The jump-diffusion model results in a value about 50% larger than the pure diffusion value. Figure 4.12(a) displays the difference between



**Figure 4.11** European call option results. (a) Present value under jump-diffusion model (solid curve), pure diffusion model (dotted curve), and payoff value (dashed curve). (b) Difference between jump-diffusion and pure diffusion present option values. Parameters used:  $T = 1$ ,  $K = 100$ ,  $D_0 = 0.0$ ,  $r = 0.05$ ,  $\sigma = 0.2$ ,  $\lambda = 2$ ,  $\delta = 0.2$ ,  $I = 300$ ,  $N = 40$ ,  $S_{\min} = 27.25$ ,  $S_{\max} = 367$ ,  $n_x = 100$ ,  $x_{\max} = 1$ ,  $x_{\min} = -1$ .

finite difference present option value and Merton's analytic formula. This difference is of order  $10^{-4}$  of the option value, demonstrating the accuracy of the numerical technique. Finally, Figure 4.12(b) shows the jump-diffusion  $\Delta$  (solid curve) and the pure diffusion  $\Delta$  (dotted curve). Again there are substantial differences.

Independent of the numerical integration algorithm used and of the interpolation and extrapolation techniques used to obtain the post-jump option values  $V(S e^x)$ , from the finite difference grid in the



**Figure 4.12** European call option results. (a) Error ( $\times 10^4$ ) in finite difference present value compared with Merton's analytic formula. (b) Present option  $\Delta$ , under jump-diffusion model (solid curve) and under pure diffusion model (dashed curve). Parameters used same as those for Figure 4.11.

convolution term of the pricing PIDE, the numerical convolution can always be expressed as a vector-matrix multiplication

$$\int_{-\infty}^{\infty} V(Se^x) G(x) dx \approx \mathbf{A}\mathbf{V} \quad (4.47)$$

where  $\mathbf{V}$  is the vector of option values at the grid points and  $\mathbf{A}$  is the convolution matrix. In principle,  $\mathbf{A}$  is a dense matrix, but if we eliminate those elements corresponding to  $|x| \gg \delta$  (hence very small values of  $G(x)$ ), then  $\mathbf{A}$  becomes banded. The number of significant

bands is proportional to the average jump size divided by the grid spacing  $n_B \approx (S\delta)/\Delta S$ . When the jump probability distribution is independent of time,  $\mathbf{A}$  can be precomputed and efficiently stored, retaining just the significant bands. The additional computational work involved in calculating the convolution term is thus proportional to  $In_B$ , where  $I$  is the grid size.

If  $n_B$  is a number of order 10, it is probably easiest to implement the numerical convolution as the vector-matrix multiplication. When  $n_B$  is much larger, then the computational work involved in the vector-matrix multiplication becomes onerous, and it may be worthwhile to work in Fourier transform space (Andersen & Andreason, 1999). The Fourier convolution theorem states that the Fourier transform of a convolution is just the product of the Fourier transform of the convolution kernel and the transform of the function (Morse & Feshbach, 1955). Since numerical Fourier transforms can be carried out very efficiently via the FFT algorithm, with computational work proportional to  $I \log(I)$ , this approach will be more efficient for large  $n_B$ . To make use of this technique, however, the numerical integration must represent a true convolution operator (i.e., the rows of  $\mathbf{A}$  must be identical except for a horizontal shift). This eliminates any special treatment, such as extrapolation, at the ends of the finite difference grid. This restriction and the periodicity imposed by the FFT itself mean that information from the region near  $S_{\min}$  can leak into the computation of  $V$  near  $S_{\max}$  and vice versa. If the grid is not large enough to accommodate this "wrap-around," numerical errors can result in the interior region of interest.

The techniques described here can also be applied to models with singular jump distributions. Consider the variance gamma process model (Madan, Carr, & Chang, 1998). This model may be written:

$$\frac{\partial V}{\partial t} + (r - D_0 + \mu_Q)S \frac{\partial V}{\partial S} - rV + \int_{-\infty}^{\infty} [V(Se^x) - V(S)]k_Q(x)dx = 0$$

$$k_Q(x)dx = \frac{\exp(\theta x/\sigma^2)}{v|x|} \exp\left(-\frac{\left(\frac{2}{v} + \frac{\theta^2}{\sigma^2}\right)^{\frac{1}{2}}}{\sigma}|x|\right) dx \quad (4.48)$$

where  $\mu_Q = (1/v) \log(1-v(\theta+\sigma^2/2))$  is the jump-induced drift.

In our numerical treatment of the kernel singularity, we divide the convolution into three parts

$$\begin{aligned}
 & \int_{-\infty}^{\infty} [V(Se^x) - V(S)] k_Q(x) dx \\
 &= \int_{-\infty}^{-\varepsilon} [V(Se^x) - V(S)] k_Q(x) dx \\
 &\quad + \int_{-\varepsilon}^{\varepsilon} [V(Se^x) - V(S)] k_Q(x) dx \\
 &\quad + \int_{\varepsilon}^{\infty} [V(Se^x) - V(S)] k_Q(x) dx \\
 &= I_1 + I_2 + I_3
 \end{aligned} \tag{4.49}$$

where the integrand in the second integral may be expanded in a Taylor series

$$\begin{aligned}
 I_2 &\approx \int_{-\varepsilon}^{\varepsilon} \left[ (e^x - 1) S \frac{\partial V}{\partial S} + (e^x - 1)^2 \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} k_Q(x) dx \right] \\
 &= I_{2a} S \frac{\partial V}{\partial S} + I_{2b} \frac{1}{2} S^2 \frac{\partial^2 V}{\partial S^2}
 \end{aligned} \tag{4.50}$$

yielding

$$\begin{aligned}
 & \frac{\partial V}{\partial t} + I_{2b}(\theta, \sigma, v) \frac{1}{2} S^2 \frac{\partial^2 V}{\partial S^2} + (r - D_0 + \mu_Q + I_{2a}(\theta, \sigma, v)) S \frac{\partial V}{\partial S} - rV \\
 &+ \int_{-\infty}^{-\varepsilon} [V(Se^x) - V(S)] k_Q(x) dx \\
 &+ \int_{\varepsilon}^{\infty} [V(Se^x) - V(S)] k_Q(x) dx = 0
 \end{aligned} \tag{4.51}$$

In the limiting case:  $v \rightarrow 0$ , with  $\varepsilon$  fixed, we recover the Black-Scholes equation.

$$I_1 \rightarrow 0, I_3 \rightarrow 0, I_{2a} \rightarrow \theta + \frac{\sigma^2}{2}, I_{2b} \rightarrow \sigma^2 \tag{4.52}$$

In a finite difference solution, it is natural (but not required) that  $\varepsilon = \Delta S/K$ , where  $K$  is the strike price or another characteristic price.

## REFERENCES

- Andersen, L., & Andreason, J. (1999). Jumping Smiles. *Risk*, 12, 65-68.
- Craig, I.J.D., & Sneyd, A.D. (1990). An Alternating Direction Implicit Scheme for Parabolic Systems of Partial Differential Equations. *Computers and Math with Applications*, 20(3), 53-62.
- Curran, M. (1995). Beyond Average Intelligence. In R. Jarrow (Ed.), *Over the Rainbow: Developments in Exotic Options and Complex Swaps*. London: Risk Publications.
- deBoor, C. (1978). *A Practical Guid to Splines*. New York: Springer-Verlag.
- Madan, P., Carr, P., & Chang, E. (1998). The Variance Gamma Process and Option Pricing. *European Financial Review*, 1.
- Merton, R. (1976). Option Pricing When Underlying Stock Returns Are Discontinuous. *Journal of Financial Economics*, 3, 125-144.
- Morse, P., & Feshbach, H. (1995). *Methods of Mathematical Physics* (p. 464). New York: McGraw-Hill.
- Press, W., Flannery, B., Teukolsky, S., & Vetterling, W. (1986). *Numerical Recipes*. Cambridge, England: Cambridge University Press.
- Randall, C. (1989). Absorbing Boundary Conditions for the Elastic Wave Equation: Velocity-Stress Formulation. *Geophysics*, 59, 1141-1152.
- Tavella, D. (1997, September 1). Foreign Exchange Basket Options. *Derivatives Week*.
- Wilmott, P., Dewynne, J., & Howison, S. (1993). *Option Pricing: Mathematical Models and Computation*. Oxford, England: Oxford Financial Press.
- Zvan, R., Forsyth, P.A., & Vetzal, K.R. (1998). Robust Numerical Methods for PDE Models of Asian Options. *Journal of Computational Finance*, 1, 39-78.

## *Chapter 5*

# Coordinate Transformations

In this chapter, we explain how to concentrate grid points near critical prices such as strike or barrier prices, or to assure that the finite difference grid passes through a series of perhaps irregularly spaced critical prices. The technique we use is the coordinate transformation. This systematic technique preserves the accuracy of the discretized pricing equations. Coordinate transformations can also be used to generate grids that move in synchrony with critical prices that move in time. By transforming the pricing equation, we can achieve better performance, requiring fewer grid points for a given accuracy or better accuracy for a given level of computational work.

Coordinate transformations have wide application in the numerical solution of partial differential equations in many fields of physics and engineering, where their principal uses include implementation of nonuniform, moving, or multidimensional conforming grids. Suitably chosen coordinate transformations can also be used to eliminate specific troublesome terms of equations. For a formal and comprehensive treatment of coordinate transformation methodology for general partial differential equations, the interested reader may refer to Knupp and Steinberg (1993) and the references therein. Here we summarize the general results and give specific illustrative examples.

In Chapter 3, we introduced the simplest version of a coordinate transformation when we defined backward time,  $t = T - t$ . A simple example of a "space" coordinate transformation is the logarithmic price transformation. For example, the one-dimensional Black-Scholes equation for a lognormal price process

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r - D_0) S \frac{\partial V}{\partial S} - rV = 0 \quad (5.1)$$

may be written under the transformation  $x = \log(S/S_0)$

$$\frac{\partial u}{\partial t} + \frac{1}{2} \sigma^2 \frac{\partial^2 u}{\partial x^2} + (r - D_0 - \frac{1}{2} \sigma^2) \frac{\partial u}{\partial x} - ru = 0 \quad (5.2)$$

where  $u(x) = V(\log(S/S_0))$ , and  $S_0$  is a constant.

Other transformations with specific properties can be used to achieve much more accurate and efficient finite difference solutions, depending on the financial features of the contract under consideration. Although  $u$  and  $V$  are mathematically distinct functions, we will henceforth no longer burden the notation by using two names for value, retaining  $V$  for both transformed and original pricing equations.

### ONE-DIMENSIONAL, TIME-INDEPENDENT TRANSFORMATIONS

The simplest coordinate transformations are one-dimensional and time-independent. To illustrate, we return to the one-dimensional Black-Scholes equation formulated in price  $S$  coordinates. We designate  $S$  the "original" space coordinate, and we wish to write the equation in a "new" coordinate  $\xi$ . The coordinate transformation itself is most conveniently specified by an equation that defines the original space coordinate in terms of the new one:

$$S = S(\xi) \quad (5.3)$$

Then, using the chain rule, it is straightforward to write Equation 5.1 in the new coordinate  $\xi$

$$\frac{\partial V}{\partial t} + \frac{\sigma^2}{2} \frac{S^2(\xi)}{J(\xi)} \frac{\partial}{\partial \xi} \left( \frac{1}{J(\xi)} \frac{\partial V}{\partial \xi} \right) + (r - D_0) \frac{S(\xi)}{J(\xi)} \frac{\partial V}{\partial \xi} - rV = 0 \quad (5.4)$$

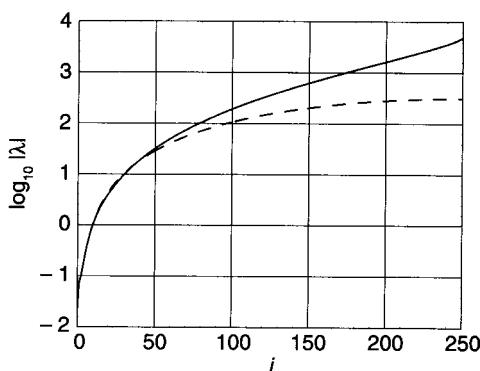
where

$$J(\xi) = \frac{dS(\xi)}{d\xi} \quad (5.5)$$

is the Jacobian of the transformation. Using this general form, it is straightforward to recover the Black-Scholes equation in log-price coordinates, Equation 5.2:

$$\begin{aligned} S(\xi) &= S_0 \exp(\xi) \\ J(\xi) &= S_0 \exp(\xi) \\ \frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 \frac{\partial^2 V}{\partial \xi^2} + (r - D_0 - \frac{1}{2} \sigma^2) \frac{\partial V}{\partial \xi} - rV &= 0 \end{aligned} \quad (5.6)$$

Coordinate transformations can affect finite difference calculations in many ways. Among them are stability, accuracy, and solver convergence rate. For example, consider what happens to the eigenvalues of the discretization matrix  $A$  discussed in Chapter 3 under the log-price coordinate transformation. Figure 5.1 shows the distribution of  $|\lambda_i|$  for the discretizations of a one-dimensional Black-Scholes equation, with and without the logarithmic transformation. The way this coordinate transformation affects the problem is by dramatically “flattening” the distribution of the discretization matrix eigenvalues. This means that if we are using an explicit scheme,



**Figure 5.1** Eignenvalues  $\log_{10}(|\lambda_i|)$  for a second order central difference discretization of the Black-Scholes equation in spot price coordinates (solid curve) and in log-price coordinates (dashed curve). Parameters used:  $K = 1$ ,  $\sigma = 0.2$ ,  $r = 0.07$ ,  $D_0 = 0.02$ ,  $\xi_{\min} = -2$ ,  $\xi_{\max} = 2$ ,  $S_{\min} = \exp(\xi_{\min})$ ,  $S_{\max} = \exp(\xi_{\max})$ ,  $I = 250$ .

we are able to use a larger number of grid points for a given time step (or a larger time step for a given number of grid points) before encountering stability problems. If we are using an implicit scheme, stability is not critically affected, but accuracy may be.

In obtaining Equation 5.6 from Equation 5.4, we have expanded the chained derivatives in the diffusion term. Because the transformation in this case is quite simple, it is easy to do. Most often, however, the coordinate transformation is more complex than this, even in one dimension, and it makes more sense to spatially discretize Equation 5.4 directly:

$$\frac{\partial V}{\partial t} + \frac{\sigma^2}{2} \frac{S(\xi)^2}{J(\xi)} \left( \frac{V(\xi + \Delta\xi) - V(\xi)}{J\left(\xi + \frac{\Delta\xi}{2}\right)\Delta\xi^2} - \frac{V(\xi) - V(\xi - \Delta\xi)}{J\left(\xi - \frac{\Delta\xi}{2}\right)\Delta\xi^2} \right) + (r - D_0) \frac{S(\xi)}{J(\xi)} \left( \frac{V(\xi + \Delta\xi) - V(\xi - \Delta\xi)}{2\Delta\xi} \right) - rV(\xi) = 0 \quad (5.7)$$

This discrete equation can be expanded in a Taylor series (with the aid of a computer algebra system!) and it can be verified that it is accurate to second order in  $\Delta\xi$ . That is, the leading order error term, after subtracting the continuous equation, is proportional to  $\Delta\xi^2$ . The error expression is lengthy, and here we reproduce only a portion of the leading order term:

$$\Delta\xi^2 \left( \frac{\sigma^2 S(\xi)^2}{24 J(\xi)^2} \right) \left( \frac{\partial^4 V(\xi)}{\partial \xi^4} + 2 \frac{\partial^3 V(\xi)}{\partial \xi^3} \frac{\log(J(\xi))}{\partial \xi} + \dots \right) \quad (5.8)$$

Thus, error depends not only on the variation of the dependent variable  $V(\xi)$ , but also on the variation of  $\log(J(\xi))$ . If we specify a coordinate transformation that is not sufficiently smooth, then we introduce another source of error.

Without loss of generality, we can use a uniformly spaced grid in the new coordinate,  $\xi_{j+1} - \xi_j = \Delta\xi$ . This uniform spacing in  $\xi$  implies an effective nonuniform spacing in the original coordinate  $\Delta S(\xi) = J(\xi)\Delta\xi$  if  $S(\xi)$  is a nonlinear function. The meaning of sufficiently smooth can now be clarified as follows. Let the variation of  $V$  be characterized by the scale length  $L$ . Then the coordinate transformation

does not introduce additional error if the fractional change in  $\Delta S(\xi)$  over the length  $L$  is small.

For nontrivial transformations, it is usually most efficient to precompute the original coordinate  $S_j$  and Jacobian  $J_{j+1/2}$  arrays for use in the discretized equations:

$$\begin{aligned} S_j &= S(\xi_j) \\ J_{j+1/2} &= \frac{S_{j+1} - S_j}{\xi_{j+1} - \xi_j} = \frac{S_{j+1} - S_j}{\Delta\xi} \end{aligned} \quad (5.9)$$

Then Equation 5.7 can be written in index form

$$\begin{aligned} \frac{\partial V}{\partial t} + \sigma^2 \frac{S_j^2}{J_{j+1/2} + J_{j-1/2}} &\left( \frac{V_{j+1} - V_j}{J_{j+1/2} \Delta\xi^2} - \frac{V_j - V_{j-1}}{J_{j-1/2} \Delta\xi^2} \right) \\ + (r - D_0) \frac{S_j}{J_{j+1/2} + J_{j-1/2}} &\left( \frac{V_{j+1} - V_{j-1}}{\Delta\xi} \right) - rV_j = 0 \end{aligned} \quad (5.10)$$

where we have approximated  $2J_j = J_{j+1/2} + J_{j-1/2}$ , which is consistent with the second order accuracy of the discretization.

## Transformations Place Grid Points at Selected Positions

Sometimes it is necessary for greatest accuracy that one or more critical points have a specific position on the grid (e.g., a point should lie exactly on grid a point, or midway between two grid points), but that the grid may be otherwise nearly uniform. That is, we wish to construct a grid that is nearly uniform and smoothly varying, but “pinned” at the critical points. In such cases, spline transformations are particularly useful. Let  $\{B_k\}$  be the set of critical points, where  $1 \leq k \leq K$  and assume that  $0 \leq \xi \leq 1$ . Then we may find a set of associated grid points  $\{\xi_k\}$

$$\xi_k = \frac{1}{I} \text{Round} \left( \frac{B_k - S_{\min}}{S_{\max} - S_{\min}} I \right) \quad (5.11)$$

and further define

$$\xi'_k = \xi_k + \frac{\beta_k}{2I} \quad (5.12)$$

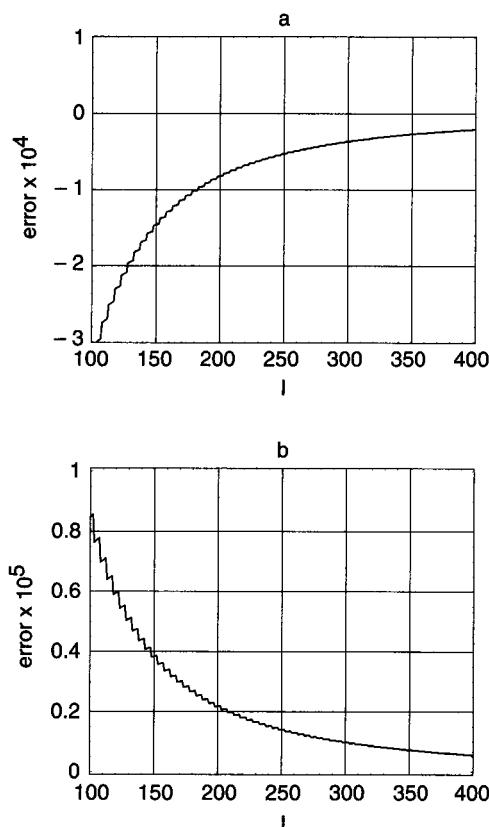
We can fit a piecewise continuous polynomial interpolator to the  $K + 2$  set of coordinate pairs  $\{0, S_{\min}\}, \{\xi'_1, B_1\}, \dots, \{\xi'_K, B_K\}, \{1, S_{\max}\}$ . We set  $\beta_k = 0$  to make  $B_k$  lie on a grid point, and  $\beta_k = 1$  to make  $B_k$  lie midway between grid points, thereby obtaining a continuous function  $S(\xi)$  such that

$$\begin{aligned} S(0) &= S_{\min} \\ S(\xi_k) &= B_k \quad \beta_k = 0 \\ \frac{S(\xi_k) + S(\xi_{k+1})}{2} &= B_k \quad \beta_k = 1 \\ S(1) &= S_{\max} \end{aligned} \quad (5.13)$$

Simple piecewise linear interpolation yields a piecewise constant  $\Delta S(\xi) = J(\xi)\Delta\xi$ . The jumps in  $\Delta S$  decrease the order of accuracy of the discretization as shown in Equation 5.8. An interpolating function with at least one continuous derivative is required. Cubic spline interpolation, which has a continuous second derivative is a superior choice.

This type of transformation provides an alternative solution to the problem of quantization error. In Chapter 4 we discussed a simple approach based on placing the cell-averaged value of the payoff on the finite difference grid rather than simply sampling the payoff onto the grid. The sensitivity to calculated option value to the location of the strike price relative to the grid is greatly reduced by this simple approach. However, using coordinate transformations that pin the strike price to particular locations on the grid can be equally or more effective. In this case we simply sample the payoff onto a finite difference grid that is slightly nonuniform and adjusted so that the option strike falls at a particular location.

In Figure 5.2 we display at-the-money errors in the finite difference calculation of the European call option of Figure 4.1 compared to the analytic result versus  $I$ , the number of spatial grid points.



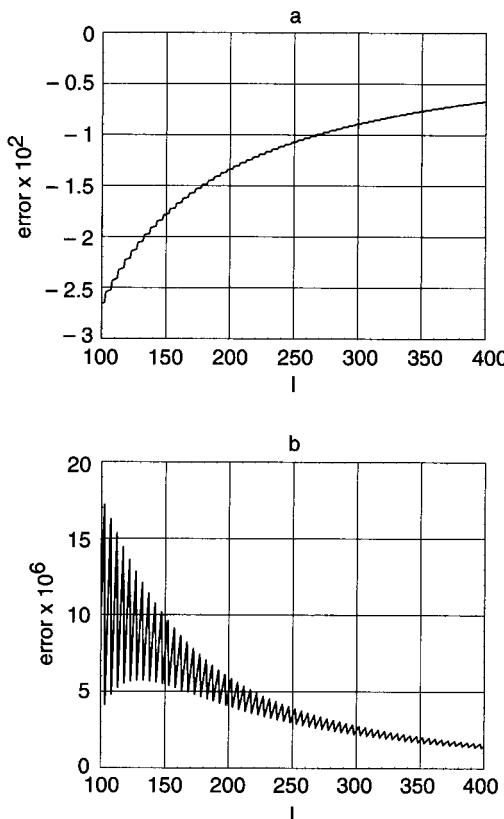
**Figure 5.2** At-the-money errors in the finite difference calculation compared to the Black-Scholes result for a European call option versus  $I$ , the number of spatial grid points. The expiration option value is the sampled payoff function. (a) The coordinate transformation places the strike exactly on a grid point. (b) The coordinate transformation places the strike midway between two grid points. Parameters used:  $K = 1$ ,  $\sigma = 0.35$ ,  $r = 0.05$ ,  $D_0 = 0.025$ ,  $T = 1$ ,  $S_{\min} = 0$ ,  $S_{\max} = 5$ .

The expiration option value is the sampled value of the payoff function  $F(S_i)$ . In Figure 5.2(a), the coordinate transformation is adjusted so that the strike price falls exactly on a grid point, while in Figure 5.2(b), the strike falls midway between two grid points. In these two error plots, the error has been scaled by a large number for clarity. The number of time steps is held fixed for all these

results at  $N = 1000$ , which renders the time discretization component of the error negligible compared to the price discretization component. With the strike falling on a grid point we get a very smooth error curve, roughly corresponding to the envelope of that in Figure 4.2(a), which corresponds to sampling the payoff onto a uniform grid. When the transformation is adjusted so that the strike falls midway between two grid points, the error is quite similar to that in Figure 4.2(b), which corresponds to placing the cell-average payoff onto the grid. The error with the second coordinate transformation is the same order of magnitude, but smoother than that with the cell average. Thus extrapolation to the continuous result is more reliable, and the coordinate transformation technique can be applied to more exotic payoffs.

The improvement of the coordinate transformation technique over the cell-averaging technique is even more pronounced for a digital call option. Figure 4.3 shows the present option value. In Figure 5.3 we display at-the-money errors in the finite difference calculation of the digital call option of Figure 4.3 compared to the analytic result versus  $I$ , the number of spatial grid points. In Figure 5.3(a), the coordinate transformation is adjusted so that the strike price falls exactly on a grid point, while in Figure 5.3(b), the strike falls midway between two grid points. With strike falling on a grid point we get a very smooth error curve roughly corresponding to the envelope of that in Figure 4.3(a), which corresponds to sampling the payoff onto a uniform grid. However, when the transformation is adjusted so that the strike falls midway between two grid points, the error is drastically reduced even relative to that in Figure 4.3(b), which corresponds to placing the cell-average payoff onto the grid. This very significant improvement in accuracy for discontinuous payoffs will be shown later to carryover into *discretely* sampled barrier options, where the sampling periodically creates a discontinuous option value at the barrier. This will be discussed further in Chapter 6.

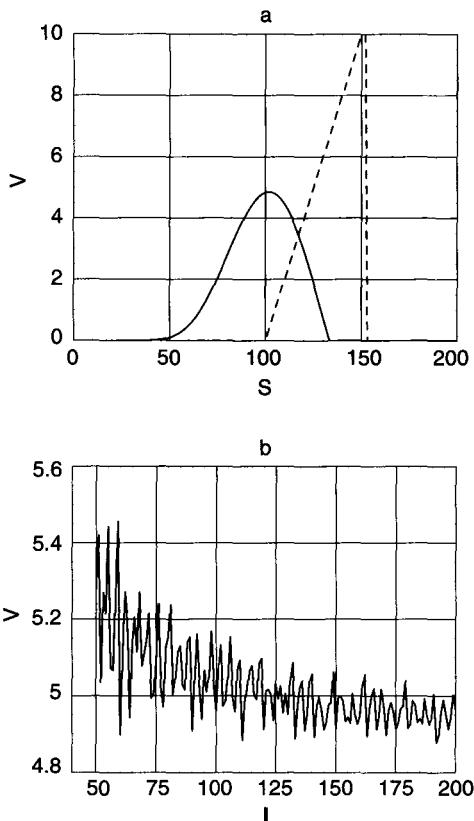
For *continuously* sampled barrier options, we want to place barriers exactly at grid points. Figures 5.4 and 5.5 display results for an up-and-out call option in which the barrier is periodically reset. We use a finite difference grid that encompasses all the possible barrier positions, and then enforce the condition  $V(S_j, t) = 0$  for  $S_j \sim B(t) \in \{B_k\}$ . An SOR solver can be used to do this, but it is much more efficient to simply incorporate the barrier condition directly into the



**Figure 5.3** At-the-money errors in the finite difference calculation compared to the Black-Scholes result for a European digital call option versus  $L$ , the number of spatial grid points. The expiration option value is the sampled payoff function. (a) The coordinate transformation places the strike exactly on a grid point. (b) The coordinate transformation places the strike midway between two grid points. Parameters used:  $K = 1$ ,  $\sigma = 0.35$ ,  $r = 0.05$ ,  $D_0 = 0.025$ ,  $T = 1$ ,  $S_{\min} = 0$ ,  $S_{\max} = 5$ .

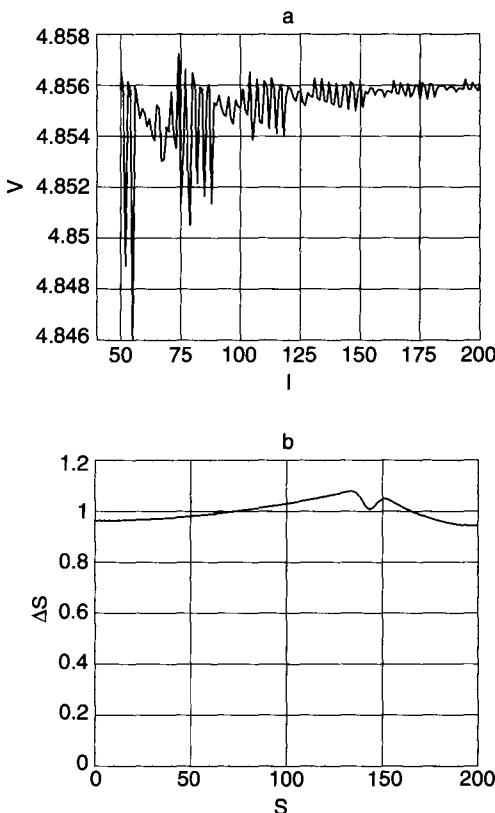
matrix, substituting it for the pricing equation at grid points above the barrier. This is discussed in Chapter 6. An efficient tridiagonal solver can then be used to solve for option value.

The option expires in one year and is struck at 100. The financial parameters are  $c_r = 0.3$ ,  $r = 0.05$ , and  $D_0 = 0$ . The barrier resets quarterly, sequentially taking the values  $B_k = \{133.3, 139.69, 145.8, 153.11\}$ . We have chosen odd-looking numbers intentionally, so that



**Figure 5.4** Results for an up-out call option struck at 100, expiring in one year, for which the barrier is reset quarterly. (a) Expiration (dashed curve, scaled by  $\frac{1}{5}$  for plotting) and present option values (solid curve). (b) Variation of present option value at a spot price of  $S = 100$  as a function of grid size when a uniform mesh is used.

no uniform grid of reasonable size can include all of them. In Figure 5.4(a) the dashed curve is the option payoff (scaled by  $V_s$  for plotting purposes) and the solid curve is the present option value. The increase of the barrier position with time is evident. Figure 5.4(b) is a plot of present option value at a spot value of  $S = 100$  as a function of grid size when a uniform mesh is used. Even with 200 grid points, option value is uncertain by more than 2%. For spot prices closer to the initial barrier position, fractional variations in option value are much higher.



**Figure 5.5** Results for a same option as in Figure 5.4. (a) Variation of present option value at a spot price of  $S = 100$  as a function of grid size when a uniform mesh is used. (b)  $\Delta S$  as a function of  $S$  with a grid size of 200.

The same variation of option value using a grid “pinned” to the set of barrier prices, and interpolated to the spot value  $S = 100$  with a cubic spline is shown in Figure 5.5(a). The transformation is adjusted to put the strike price midway between two grid points, while the barrier prices are exactly on grid points. The barrier condition is incorporated directly into the matrix. The variation of calculated option value with grid size has been reduced by about factor of one hundred. Finally, Figure 5.5(b) displays the local  $\Delta S$  as a function of  $S$  with a grid size of  $I = 200$ . It is smooth and the fractional change in  $\Delta S$  from grid point to grid point is of order 1% at most. This ensures that the coordinate transformation has

introduced no significant errors in addition to the finite difference truncation error itself.

### Transformations That Concentrate Grid Points

Often it is of interest to concentrate finite difference grid points near critical points such as barrier prices. There are an unlimited number of ways of achieving this with coordinate transformations, but one simple approach is to let:

$$J(\xi) = A(\alpha^2 + (S(\xi) - B)^2)^{\frac{1}{2}} \quad (5.14)$$

where  $A$  is a constant to be determined,  $\alpha$  is a prescribed uniformity parameter, and  $B$  is the critical price (by critical price we mean the area where additional resolution is desired). Obviously, if  $|S - B| \gg \alpha$ , then the effective grid spacing in  $S$  is proportional to  $S$ . That is, the transformation is approximately logarithmic. However, in the neighborhood  $|S - B| \leq \alpha$ , the effective spacing in  $S$  is nearly uniform.

Integrating Equation 5.14 and supplying the boundary conditions  $S(\xi = 0) = S_{\min}$  and  $S(\xi = 1) = S_{\max}$  yields the coordinate transformation with the desired properties

$$S(\xi) = B + \alpha \sinh(c_2 \xi + c_1(1 - \xi)) \quad (5.15)$$

where

$$\begin{aligned} c_1 &= \sinh^{-1}(S_{\min} - B)/\alpha \\ c_2 &= \sinh^{-1}(S_{\max} - B)/\alpha \end{aligned} \quad (5.16)$$

This transformation maps the interval  $0 \leq \xi \leq 1$  to the interval  $S_{\min} \leq S \leq S_{\max}$ . The value of  $\xi$  corresponding to the critical point  $S(\xi_B) = B$  is

$$\xi_B = \frac{-c_1}{c_2 - c_1} \quad (5.17)$$

The value of  $\Delta S$  is minimized at  $\xi_B$ . Grid points are therefore concentrated near the critical point. Setting  $\alpha \ll S_{\max} - S_{\min}$  yields a highly nonuniform grid, while  $\alpha \gg S_{\max} - S_{\min}$  yields a uniform grid.

Two particularly simple special cases of the preceding transformation can be easily obtained. If we rederive Equation 5.15 with  $B = \alpha = 0$  and let  $\xi_{\min}$  and  $\xi_{\max}$  float, we can set  $A = 1$  and recover the familiar logarithmic transformation

$$\begin{aligned} S(\xi) &= K \exp(\xi) \\ J(\xi) &= K \exp(\xi) \end{aligned} \quad (5.18)$$

where  $K$  is a constant of integration.

Another simple special case can be obtained if we let  $B = 0$ ,  $\alpha = K$ ,  $S_{\min} = \xi_{\min} = 0$  and let  $\xi_{\max}$  float, we can set  $A = 1$  and find

$$\begin{aligned} S(\xi) &= K \sinh(\xi) \\ J(\xi) &= K \cosh(\xi) \end{aligned} \quad (5.19)$$

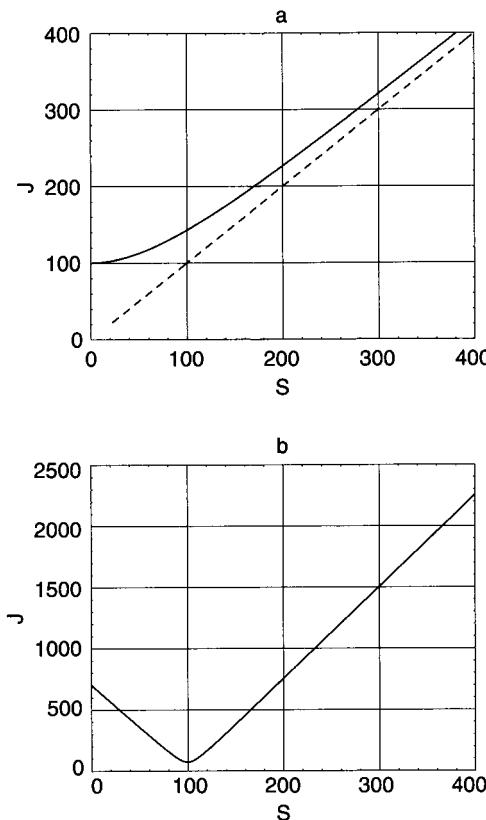
The sinh transformation is similar to the logarithmic transformation, but has the advantage that the effective grid spacing in  $S$  remains finite for  $S \rightarrow 0$ . Thus, one can set  $S_{\min} = 0$  and perhaps make use of simple boundary conditions if they are available. The approximately exponential (linear) growth in  $\Delta S$  with  $\xi(S)$  for large  $S$  is preserved. Figure 5.6 displays the Jacobian  $J$  as a function of the original coordinate  $S$  for these two special cases and for the general form Equation 5.15.

The simple transformation Equation 5.15 is easily generalized to one that concentrates grid points at each of a series of critical points  $\{B_k\}$ . We define a global Jacobian that combines the individual Jacobians for each critical point. The combination can be done in many ways, but one that yields a very smooth transformation is the harmonic squared average. We define

$$J_k(\xi) = (\alpha_k^2 + (S(\xi) - B_k)^2)^{\frac{1}{2}} \quad (5.20)$$

and

$$J(\xi) = A \left[ \sum_{k=1}^{k=n} J_k(\xi)^{-2} \right]^{-\frac{1}{2}} \quad (5.21)$$



**Figure 5.6** Jacobian  $J(\xi)$  versus  $S(\xi)$  for three one-dimensional coordinate transformations. (a) Dashed line is  $S(\xi) = K \exp(\xi)$ ,  $-1.5 \leq \xi \leq 1.5$ . Solid curve is  $S(\xi) = K \sinh(\xi)$ ,  $0 \leq \xi \leq 4$ .  $K = 100$  for both transformations. (b) Coordinate transformation of Equation 5.15, with  $B = 100$ ,  $\alpha = 10$ , and  $0 \leq \xi \leq 1$ .

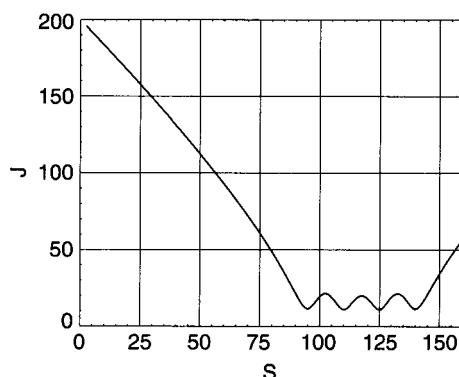
Near each critical point, the global Jacobian  $J(\xi)$  is dominated by the behavior of the local  $J_k(\xi)$ , but the influence of nearby critical points ensures that the transitions between them are smooth. The global Jacobian must be integrated numerically to yield the transformation  $S(\xi)$ . This can be done with any standard ODE integrator such as Runge-Kutta (Press, Flannery, Teukolsky, & Vetterling, 1986), using  $S(\xi = 0) = S_{\min}$  as an initial condition. A value for the normalization

constant  $A$  is guessed and the numerical integration yields  $S(\xi = 1)$ . Then  $A$  is adjusted iteratively until the other boundary condition  $S(\xi = 1) = S_{\max}$  is met. Since  $S(\xi = 1)$  is monotonically increasing with  $A$ , the numerical iteration is guaranteed to converge.

Figure 5.7 displays the Jacobian of a transformation that concentrates grid points around four critical points. The Jacobian reaches a local minimum at each critical point and varies smoothly between them.

Earlier in this chapter we showed how proper location of critical prices on the grid can dramatically improve the accuracy of finite difference calculations. We used spline-based coordinate transformations to smoothly vary the location of grid points on an otherwise nearly uniform grid. The same accuracy improvements with proper grid point location are obtained when we use a coordinate transformation to concentrate grid points. The question arises, however, how to accomplish both a prescribed grid point concentration and precise location of critical price points on the grid with a single coordinate transformation. One way would be to include a sufficient number of adjustable parameters in the transformation to allow all the conditions to be satisfied and then solve a system of nonlinear equations for these parameter values.

However, it is much easier to approach the coordinate transformation in two stages. First we generate a grid point clustering



**Figure 5.7** Jacobian  $J(\xi)$  versus  $S(\xi)$  for the coordinate transformation of Equation 5.21 with  $S_{\min} = 0$ ,  $S_{\max} = 160$ ,  $\alpha = 3$ , and  $B_k = \{95, 110, 125, 140\}$ .

coordinate transformation by solving (probably numerically), Equation 5.14 or Equations 5.20 and 5.21, for example. This yields the grid

$$S_j = S(\xi_j) \quad (5.22)$$

We can then apply a secondary transformation that will adjust the  $S_j$  by slightly stretching the  $\xi_j$  grid so that the critical points fall in the correct relationship to the grid, just as we did previously in Equations 5.11–5.13. We will call this second stretching transformation  $\psi(\xi)$ .

Since  $S_j$  is not equally spaced, we replace Equation 5.11 with

$$\begin{aligned} \xi_k^* &= \xi_k + (\xi_{k+1} - \xi_k) \frac{B_k - S_k}{S_{k+1} - S_k} \\ \hat{\xi}_k &= \frac{1}{I} \text{Round}(\xi_k^* I) \end{aligned} \quad (5.23)$$

where  $\xi_k$  is the grid point that minimizes  $|S_k - B_k|$  subject to  $S_k \leq B_k$ , and again define

$$\xi'_k = \hat{\xi}_k + \frac{\beta_k}{2I} \quad (5.24)$$

We next fit a piecewise continuous polynomial interpolator  $\psi(\xi)$  to the  $K+2$  set of coordinate pairs  $\{(0, 0), \{\xi_1^*, \xi_1'\}, \dots, \{\xi_K^*, \xi_K'\}, (1, 1)\}$ . Again we set  $\beta_k = 0$  to make  $B_k$  lie on a grid point, and  $\beta_k = 1$  to make  $B_k$  lie midway between grid points. Then

$$\begin{aligned} S(0) &= S_{\min} \\ S(\psi(\xi_k)) &= B_k \quad \beta_k = 0 \\ \frac{S(\psi(\xi_k)) + S(\psi(\xi_{k+1}))}{2} &= B_k \quad \beta_k = 1 \\ S(1) &= S_{\max} \end{aligned} \quad (5.25)$$

The grid  $S_j = S(\psi(\xi_j))$  will have both the desired grid point concentration and the correct relationship with the set of critical points. As we will see in Chapter 6, this combination can yield remarkable increases in accuracy for pricing barrier options in particular.

## ONE-DIMENSIONAL, TIME-DEPENDENT TRANSFORMATIONS

To illustrate the modeling of time-dependent barrier options, consider a corridor call option with continuously sampled moving barriers. For a lognormal underlying process, the pricing equation and boundary conditions are

$$\begin{aligned}
 \frac{\partial V}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} + (r - D_0) S \frac{\partial V}{\partial S} - rV &= 0 \\
 V(S \leq L(t), t) &= 0 \\
 V(S \geq U(t), t) &= 0 \\
 V(S, T) &= \max(S - K, 0)
 \end{aligned} \tag{5.26}$$

where the moving lower and upper barriers are  $L(t)$  and  $U(t)$ , respectively.

We define a new coordinate system,  $\xi$  and  $\tau$ , where,  $0 \leq \xi \leq 1$  and  $0 \leq \tau \leq T$ , and then define a moving grid in terms of them:

$$\begin{aligned}
 S(\xi, \tau) &= L(\tau) + \xi(U(\tau) - L(\tau)) \\
 t &= \tau
 \end{aligned} \tag{5.27}$$

In terms of the transformed coordinates, the Black-Scholes equation becomes

$$\begin{aligned}
 \frac{\partial}{\partial \tau} (J(\xi, \tau) V) + \frac{\partial}{\partial \xi} (J_t(\xi, \tau) V) \frac{\sigma^2}{2} S^2(\xi, \tau) \frac{\partial}{\partial \xi} \left( \frac{1}{J(\xi, \tau)} \frac{\partial V}{\partial \xi} \right) \\
 + (r - D_0) S(\xi, \tau) \frac{\partial V}{\partial \xi} - r J(\xi, \tau) V = 0 \\
 V(\xi = 0, \tau) &= V(S = L(t), t) = 0 \\
 V(\xi = 1, \tau) &= V(S = U(t), t) = 0 \\
 V(\xi, \tau = T) &= \max(S(\xi, T) - K, 0)
 \end{aligned} \tag{5.28}$$

where  $J(\xi, \tau) = \partial S(\xi, \tau) / \partial \xi$  is the spatial Jacobian, and  $J_t(\xi, \tau) = \partial S(\xi, \tau) / \partial \tau$  is the time Jacobian. Because the coordinate transformation is linear in  $\xi$ , the effective grid spacing,  $\Delta S = J \Delta \xi$ , is uniform. Although one could substitute the analytic expressions for  $J$  and  $J_t$  into the transformed

equation in this simple case, it is often more efficient not to do so, just differencing the equation as is with the Jacobians kept as arrays. Chapter 6 includes an illustration of the use of this type of transformation.

### MULTIDIMENSIONAL, TIME-INDEPENDENT TRANSFORMATIONS

The introduction of multidimensional transformations is a significantly more complex task than the one-dimensional case discussed in the previous sections. In multidimensional transformations, we can distinguish three levels of difficulty:

1. *Time-independent factored multidimensional transformations.* In this case, the transformations in the various directions are independent of each other and are independent of time.
2. *Time-independent general multidimensional transformations.* The transformations in the various directions influence one another but are independent of time.
3. *Time-dependent multidimensional transformations.* Here the transformations can move along with the critical features of the pricing problem.

In this book, we will discuss only the first two cases. The case of multidimensional time-dependent transformations is algebraically cumbersome but not intrinsically more difficult than the other two. In addition, we will illustrate the transformations in two dimensions only. The generalization to more dimensions is again cumbersome, but straightforward. A computer algebra system can be useful in transforming the hosted equations into the new coordinates. Consider the two-asset Black-Scholes pricing equation for the case of lognormal processes  $S_1(t)$  and  $S_2(t)$ .

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{1}{2} \sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} + \frac{1}{2} \sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} \\ + (r - D_{01}) S_1 \frac{\partial V}{\partial S_1} + (r - D_{02}) S_2 \frac{\partial V}{\partial S_2} - rV = 0 \end{aligned} \quad (5.29)$$

### Factored Multidimensional, Time-Independent Transformations

In this case, each of the original coordinates depends on only one of the new coordinates:

$$\begin{aligned} S_1 &= S_1(\xi) \\ S_2 &= S_2(\eta) \end{aligned} \quad (5.30)$$

This kind of coordinate transformation is called a factored transformation and produces much simpler transformed equations than in the general case discussed in the next subsection. There are two Jacobians

$$\begin{aligned} J_1 &= \frac{\partial S_1}{\partial \xi} \\ J_2 &= \frac{\partial S_2}{\partial \eta} \end{aligned} \quad (5.31)$$

with

$$\begin{aligned} \frac{\partial V}{\partial S_1} &= \frac{1}{J_1} \frac{\partial V}{\partial \xi} \\ \frac{\partial V}{\partial S_2} &= \frac{1}{J_2} \frac{\partial V}{\partial \eta} \end{aligned} \quad (5.32)$$

and

$$\begin{aligned} \frac{\partial^2 V}{\partial S_1^2} &= \frac{1}{J_1} \frac{\partial}{\partial \xi} \left( \frac{1}{J_1} \frac{\partial V}{\partial \xi} \right) \\ \frac{\partial^2 V}{\partial S_1 \partial S_2} &= \frac{1}{J_1 J_2} \left( \frac{\partial^2 V}{\partial \xi \partial \eta} \right) \\ \frac{\partial^2 V}{\partial S_2^2} &= \frac{1}{J_2} \frac{\partial}{\partial \eta} \left( \frac{1}{J_2} \frac{\partial V}{\partial \eta} \right) \end{aligned} \quad (5.33)$$

With these expressions, Equation 5.29 becomes

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{1}{2} \sigma_1^2 \frac{S_1^2}{J_1} \frac{\partial}{\partial \xi} \left( \frac{1}{J_1} \frac{\partial V}{\partial \xi} \right) + \rho \sigma_1 \sigma_2 \frac{S_1 S_2}{J_1 J_2} \left( \frac{\partial^2 V}{\partial \xi \partial \eta} \right) \\ \frac{1}{2} \sigma_2^2 \frac{S_2^2}{J_2} \frac{\partial}{\partial \eta} \left( \frac{1}{J_2} \frac{\partial V}{\partial \eta} \right) + (r - D_{01}) \frac{S_1}{J_1} \frac{\partial V}{\partial \xi} + (r - D_{02}) \frac{S_2}{J_2} \frac{\partial V}{\partial \eta} - rV = 0 \end{aligned} \quad (5.34)$$

Such a factored transformation might be useful, for example, for two-asset digital or barrier options.

### General Multidimensional, Time-Independent Transformations

In this more general coordinate transformation, the original price coordinates  $S_1$  and  $S_2$  are each defined in terms of both transformed  $\xi$  and  $\eta$

$$\begin{aligned} S_1 &= S_1(\xi, \eta) \\ S_2 &= S_2(\xi, \eta) \end{aligned} \quad (5.35)$$

Then we may write

$$\begin{aligned} \frac{\partial V}{\partial \xi} &= \frac{\partial V}{\partial S_1} \frac{\partial S_1}{\partial \xi} + \frac{\partial V}{\partial S_2} \frac{\partial S_2}{\partial \xi} \\ \frac{\partial V}{\partial \eta} &= \frac{\partial V}{\partial S_1} \frac{\partial S_1}{\partial \eta} + \frac{\partial V}{\partial S_2} \frac{\partial S_2}{\partial \eta} \end{aligned} \quad (5.36)$$

Inverting these equations, we have

$$\begin{aligned} \frac{\partial V}{\partial S_1} &= \frac{1}{J} \left( \frac{\partial V}{\partial \xi} \frac{\partial S_2}{\partial \eta} - \frac{\partial V}{\partial \eta} \frac{\partial S_2}{\partial \xi} \right) \\ \frac{\partial V}{\partial S_2} &= \frac{1}{J} \left( \frac{\partial V}{\partial \eta} \frac{\partial S_1}{\partial \xi} - \frac{\partial V}{\partial \xi} \frac{\partial S_1}{\partial \eta} \right) \end{aligned} \quad (5.37)$$

where the Jacobian is defined as follows

$$J = \left( \frac{\partial S_1}{\partial \xi} \frac{\partial S_2}{\partial \eta} - \frac{\partial S_1}{\partial \eta} \frac{\partial S_2}{\partial \xi} \right) \quad (5.38)$$

These expressions may also be written

$$\begin{aligned}\frac{\partial V}{\partial S_1} &= \frac{1}{J} \left[ \frac{\partial}{\partial \xi} \left( V \frac{\partial S_2}{\partial \eta} \right) - \frac{\partial}{\partial \eta} \left( V \frac{\partial S_2}{\partial \xi} \right) \right] \\ \frac{\partial V}{\partial S_2} &= \frac{1}{J} \left[ \frac{\partial}{\partial \eta} \left( V \frac{\partial S_1}{\partial \xi} \right) - \frac{\partial}{\partial \xi} \left( V \frac{\partial S_1}{\partial \eta} \right) \right]\end{aligned}\quad (5.39)$$

Then, using the preceding equations, we can write expressions for the second derivatives

$$\begin{aligned}J \frac{\partial^2 V}{\partial S_1^2} &= \frac{\partial}{\partial \xi} \left[ \frac{1}{J} \left( \frac{\partial V}{\partial \xi} \frac{\partial S_2}{\partial \eta} - \frac{\partial V}{\partial \eta} \frac{\partial S_2}{\partial \xi} \right) \frac{\partial S_2}{\partial \eta} \right] \\ &\quad - \frac{\partial}{\partial \eta} \left[ \frac{1}{J} \left( \frac{\partial V}{\partial \xi} \frac{\partial S_2}{\partial \eta} - \frac{\partial V}{\partial \eta} \frac{\partial S_2}{\partial \xi} \right) \frac{\partial S_2}{\partial \eta} \right]\end{aligned}\quad (5.40)$$

$$\begin{aligned}2J \frac{\partial^2 V}{\partial S_1 \partial S_2} &= \frac{\partial}{\partial \xi} \left[ \frac{1}{J} \frac{\partial V}{\partial \eta} \left( \frac{\partial S_1}{\partial \xi} \frac{\partial S_2}{\partial \eta} + \frac{\partial S_1}{\partial \eta} \frac{\partial S_2}{\partial \xi} \right) - \frac{2}{J} \frac{\partial V}{\partial \xi} \frac{\partial S_1}{\partial \eta} \frac{\partial S_2}{\partial \eta} \right] \\ &\quad + \frac{\partial}{\partial \eta} \left[ \frac{1}{J} \frac{\partial V}{\partial \xi} \left( \frac{\partial S_1}{\partial \xi} \frac{\partial S_2}{\partial \eta} + \frac{\partial S_1}{\partial \eta} \frac{\partial S_2}{\partial \xi} \right) - \frac{2}{J} \frac{\partial V}{\partial \eta} \frac{\partial S_1}{\partial \xi} \frac{\partial S_2}{\partial \xi} \right]\end{aligned}\quad (5.41)$$

$$\begin{aligned}J \frac{\partial^2 V}{\partial S_2^2} &= \frac{\partial}{\partial \eta} \left[ \frac{1}{J} \left( \frac{\partial V}{\partial \eta} \frac{\partial S_1}{\partial \xi} - \frac{\partial V}{\partial \xi} \frac{\partial S_1}{\partial \eta} \right) \frac{\partial S_1}{\partial \xi} \right] \\ &\quad - \frac{\partial}{\partial \xi} \left[ \frac{1}{J} \left( \frac{\partial V}{\partial \eta} \frac{\partial S_1}{\partial \xi} - \frac{\partial V}{\partial \xi} \frac{\partial S_1}{\partial \eta} \right) \frac{\partial S_1}{\partial \eta} \right]\end{aligned}\quad (5.42)$$

Collecting these expressions for the various derivatives in Equation 5.29 yields the general transformed two-dimensional pricing equation. The complexity introduced into multidimensional pricing equations by a general coordinate transformation is obviously considerable. The algebraic burden and probability of errors can be reduced with computer algebra systems, many of which can generate blocks of code in programming languages such as C. The return for the effort can be considerable however, easily improving program performance by an order of magnitude or more. In the next

section, we demonstrate how this return may accrue for even the simplest of the multidimensional coordinate transformations, the linear transformation.

## Multidimensional Linear Transformations

In a linear coordinate transformation, the new and original coordinates are related by a set of linear equations. For example, in two dimensions

$$\begin{aligned} S_1 &= \alpha\xi + \beta\eta \\ S_2 &= \gamma\xi + \delta\eta \end{aligned} \tag{5.43}$$

$$\begin{aligned} \xi &= \hat{\alpha}S_1 + \hat{\beta}S_2 \\ \eta &= \hat{\gamma}S_1 + \hat{\delta}S_2 \end{aligned} \tag{5.44}$$

where  $\alpha, \beta, \dots$ , are constants. More generally, we can write

$$\begin{aligned} S &= \hat{\Gamma}\xi \\ \xi &= \hat{\Gamma}^{-1}S \end{aligned} \tag{5.45}$$

where  $\hat{\Gamma}$  is the transformation matrix,  $S = (S_1, S_2, \dots)$  is the vector of original coordinates and  $\xi = (\xi, \eta, \dots)$  is the vector of new coordinates. Linear transformations describe a rotation and stretching of coordinate axes. The new coordinates may or may not be orthogonal. Linear transformations can be used for eliminating the cross derivative term from the transformed pricing equation (Clewlow & Strickland, 1998). This allows the use of very simple ADI (Alternating Direction Implicit) solvers (Richtmeyer & Morton, 1967) that do not deal with cross derivatives. In principle, elimination of these terms should also improve the convergence of more advanced solvers, such as those of the conjugate gradient family. In practice, however, such gains are often fairly modest.

A linear coordinate transformation can often be much more profitably used to orient the new coordinate system along the direction of greatest variation of the payoff function. A dense grid can then be used to capture the variation of the payoff function in that direction,

and a much coarser grid can be used in the transverse directions. Consider a two-asset basket call option with equal weights of the two assets. The payoff function is

$$V(S_1, S_2, T) = \max(S_1 + S_2 - K, 0) \quad (5.46)$$

When pricing a basket option in Chapter 4, we used on the order of 100 grid points in both  $S_1$  and  $S_2$  directions to achieve reasonable accuracy. If instead we make the linear transformation

$$\begin{aligned} S_1 &= \frac{1}{2}(\xi - \eta) \\ S_2 &= \frac{1}{2}(\xi + \eta) \end{aligned} \quad (5.47)$$

$$\begin{aligned} \xi &= S_1 + S_2 \\ \eta &= -S_1 + S_2 \end{aligned} \quad (5.48)$$

then the payoff function is simply

$$V(\xi, \eta, T) = \max(\xi - K, 0) \quad (5.49)$$

The option payoff has no dependence on the coordinate  $\eta$ . Of course, the option value  $V(\xi, \eta, t)$  will develop a variation with  $\eta$  as we integrate the pricing equation back to the present due to diffusion and drifts, but often this variation will remain mild in comparison with that in the  $\xi$  direction. We may be able to use a factor of 5 to 10 fewer grid points in the  $\eta$  direction. The overall gain in numerical efficiency may even be greater than this. The larger  $\Delta\eta$  yields a better conditioned matrix, so the solver convergence rate also improves.

The transformed pricing equation can be written

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{1}{2} \sigma_\xi^2 \frac{\partial^2 V}{\partial \xi^2} + \frac{1}{2} \sigma_\eta^2 \frac{\partial^2 V}{\partial \eta^2} + \sum \frac{\partial^2 V}{\partial \xi \partial \eta} \\ + \mu_\xi \frac{\partial V}{\partial \xi} + \mu_\eta \frac{\partial V}{\partial \eta} - rV = 0 \end{aligned} \quad (5.50)$$

where

$$\begin{aligned}
 \sigma_{\xi}^2 &= \frac{1}{4}(\sigma_1^2(\xi - \eta)^2 + \sigma_2^2(\xi + \eta)^2 + 2\rho\sigma_1\sigma_2(\xi^2 - \eta^2)) \\
 \sigma_{\eta}^2 &= \frac{1}{4}(\sigma_1^2(\xi - \eta)^2 + \sigma_2^2(\xi + \eta)^2 - 2\rho\sigma_1\sigma_2(\xi^2 - \eta^2)) \\
 \Sigma &= \frac{1}{4}(-\sigma_1^2(\xi - \eta)^2 + \sigma_2^2(\xi + \eta)^2) \\
 \mu_{\xi} &= \frac{1}{2}((r - D_{01})(\xi - \eta) + (r - D_{02})(\xi + \eta)^2) \\
 \mu_{\eta} &= \frac{1}{2}(-(r - D_{01})(\xi - \eta) + (r - D_{02})(\xi + \eta)^2)
 \end{aligned} \tag{5.51}$$

A minor complication introduced by this coordinate transformation is that the coefficients of the transformed pricing equation are no longer constant. Because none of the coefficients appears inside a derivative operator, they are all evaluated at grid points and second order accuracy of the discretized equations is maintained. If the original pricing equation contained variable coefficients,  $\sigma_1(S_1, t)$ , for example, then we have not added to the complexity of the solution in any way.

Table 5.1 shows the large gain in the efficiency that the linear transformation makes in pricing a European two-asset basket call option. Shown are results for both the original  $(S_1, S_2)$  price coordinates and for the  $(\xi, \eta)$  coordinate system rotated into alignment with the variation of the payoff function. The option expires in one year and is struck at  $K = 100$ . Present option value is shown in the third column for spot prices of  $S_1 = S_2 = 50$  for four grid sizes. The rows labeled "extrapolated" are the values extrapolated to infinite grid size assuming an inverse squared relationship with grid size. The difference between the finite grid values and the extrapolated values shows this inverse squared relationship for both coordinate systems. The errors for the rotated coordinates, however, are about one half those for the original coordinates even though one-fifth as many grid points were used in the  $\eta$  direction as in the  $\xi$  direction. The relative CPU times in the rightmost column show about a factor of 20 reduction for comparable accuracy in the rotated coordinate system.

Plots of present option value shown in Figure 5.8 make clear why the coordinate transformation is so effective. In the original

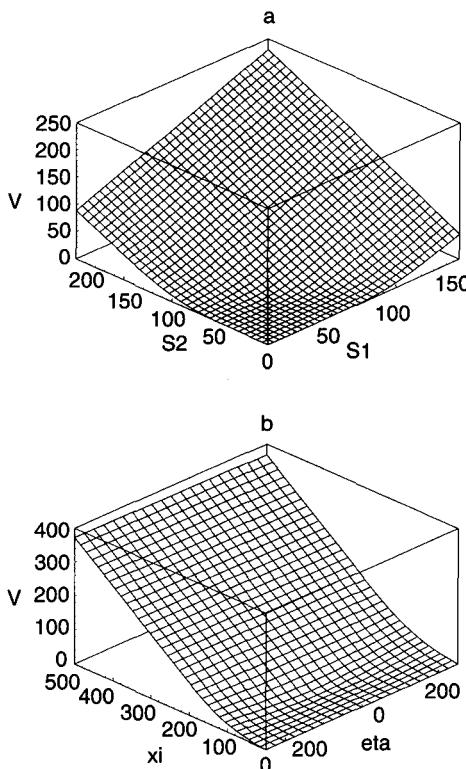
**Table 5.1 Two-Asset Basket Call Option**

Coordinates	(Grid Size) × Time Steps	$V$	$V - V_{\text{ext}}$	Relative CPU Time
$(S_1, S_2)$	$(50 \times 50) \times 25$	4.6886	0.0716	1.0
	$(100 \times 100) \times 50$	4.6356	0.0186	12.6
	$(150 \times 150) \times 75$	4.6253	0.0083	51.4
	$(200 \times 200) \times 100$	4.6216	0.0046	129.0
	extrapolated	4.6170	—	—
$(\xi, \eta)$	$(50 \times 10) \times 25$	4.6377	0.0196	0.08
	$(100 \times 20) \times 50$	4.6291	0.0110	0.9
	$(150 \times 30) \times 75$	4.6229	0.0048	5.1
	$(200 \times 40) \times 100$	4.6203	0.0022	12.9
	extrapolated	4.6181	—	—

Convergence of two-asset European basket call option calculations as a function of grid size and time steps, for both native  $(S_1, S_2)$  spot price coordinates, and for coordinates rotated to align with the payoff  $(\xi = (S_1 + S_2)/2, \eta = (S_2 - S_1)/2)$ . The third column is the present option value  $V(S_1 = 50, S_2 = 50)$ .  $V_{\text{ext}}$  is the value extrapolated to infinite grid size and infinite time steps from the finite grid-size values. The fourth column shows the convergence of the finite grid size calculations to this value as grid size is increased. The rightmost column is relative CPU time. The parameters for the two-asset basket are:  $\sigma_1 = 0.25$ ,  $\sigma_2 = 0.35$ ,  $p = -0.65$ ,  $r = 0.045$ ,  $D_{01} = 0.05$ ,  $D_{02} = 0.07$ ,  $T = 1$ ,  $K = 100$ ,  $S_{1\min} = 0$ ,  $S_{1\max} = 150$ ,  $S_{2\min} = 0$ ,  $S_{2\max} = 200$ ,  $\xi_{\min} = 0$ ,  $\xi_{\max} = 500$ ,  $\eta_{\min} = -250$ ,  $\eta_{\max} = 250$ .

$(S_1, S_2)$  coordinates shown on Figure 5.8(a), option value has substantial variation in both coordinate directions. Many grid points are required to resolve this variation in each direction. The same present option value surface in the new coordinates, shown in Figure 5.8(b), varies only slightly in the  $\eta$  direction.

In the interest of clarity, a particularly simple payoff function has been chosen for this example, with equal weighting of both assets. However, a similar coordinate transformation can be used for unequally weighted payoffs, and in more dimensions. In three dimensions, one new coordinate can be aligned with the payoff  $\xi = n_1 S_1 + n_2 S_2 + n_3 S_3$ . If we insist on an orthogonal system of new coordinates, then there remains one degree of freedom in choosing the second and third coordinates. If we allow nonorthogonal coordinates, then there are two degrees of freedom. These degrees of freedom can be exploited to reduce the cross-drift terms (i.e.,  $\eta \frac{\partial V}{\partial \xi}$ ) that tend to rotate the solution out of the  $\xi$  plane and therefore



**Figure 5.8** Present value of a European call option. (a) Calculated and plotted in  $(S_1, S_2)$  price coordinates. (b) Calculated and plotted in a  $(\xi, \eta)$  coordinate system rotated into alignment with the variation of the payoff function. The parameters for the two-asset basket are:  $\sigma_1 = 0.25$ ,  $\sigma_2 = 0.35$ ,  $\rho = -0.65$ ,  $r = 0.045$ ,  $D_{01} = 0.05$ ,  $D_{02} = 0.07$ ,  $T = 1$ ,  $K = 100$ ,  $S_{1\min} = 0$ ,  $S_{1\max} = 150$ ,  $S_{2\min} = 0$ ,  $S_{2\max} = 200$ ,  $\xi_{\min} = 0$ ,  $\xi_{\max} = 500$ ,  $\eta_{\min} = -250$ ,  $\eta_{\max} = 250$ .

increase the number of grid points required in the transverse directions. For the sake of clarity, we did not invoke this degree of freedom in the previous example. When the payoff functions of the options of interest are not as simple as those for the basket option, other coordinate transformations that greatly reduce the dependence of option value on one or more new coordinates may still be found.

**REFERENCES**

- Clewlow, L., & Strickland, C. (1998). *Implementing Derivative Models*. Chichester, England: Wiley.
- Knupp, P., & Steinberg, S. (1993). *Fundamentals of Grid Generation*. Boca Raton, FL: CRC Press.
- Press, W., Flannery, B., Teukolsky, S., & Vetterling, W. (1986). *Numerical Recipes*. Cambridge, England: Cambridge University Press.
- Richtmeyer, R., & Morton, K. (1967). *Difference Methods for Initial-Value Problems*. New York: Interscience.

# *Chapter* 6

## Numerical Examples

In this chapter, we illustrate the use of finite difference techniques for the solution of example option pricing problems.

### BARRIER OPTIONS

Barrier options are particularly challenging for traditional numerical pricing methods, which often display very poor convergence, especially for hedge parameters. For example, methods of improving the local resolution of lattices in the neighborhood of barriers have been developed but are computationally and logically complex (Ahn, Figlewski, & Gao, 1999). With finite difference methods, a simple coordinate transformation can generate moving and/or nonuniform grids, tying grids to moving barriers, concentrating grid points near barriers where gradients are steepest, and placing barrier prices in definite positions relative to the gridpoints.

### Time-Dependent Barriers

To illustrate the modeling of time-dependent barrier options, consider a European corridor call option with continuously sampled moving barriers as studied by Rogers and Zane (1997) and introduced in Chapter 5. For simplicity, we make the Black-Scholes assumptions of constant interest rate, dividend yield, and volatility, although these assumptions could be easily relaxed:

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} + (r - D_0) S \frac{\partial V}{\partial S} - rV &= 0 \\ V(S \leq L(t), t) &= 0 \\ V(S \geq U(t), t) &= 0 \\ V(S, T) &= \max(S - K, 0) \end{aligned} \tag{6.1}$$

where the moving lower and upper barriers are  $L(t)$  and  $U(t)$ , respectively.

We define a new coordinate system,  $\xi$  and  $\tau$ , where,  $0 \leq \xi \leq 1$  and  $0 \leq \tau \leq T$ , and then define a moving grid in terms of them,

$$\begin{aligned} S(\xi, \tau) &= L(\tau) + \xi(U(\tau) - L(\tau)) \\ t &= \tau \end{aligned} \quad (6.2)$$

In the transformed coordinates, the pricing equation, boundary conditions, and expiration conditions become

$$\begin{aligned} &\frac{\partial}{\partial \tau}(J(\xi, \tau)V) + \frac{\partial}{\partial \xi}(J_t(\xi, \tau)V) + \frac{\sigma^2}{2}S^2(\xi, \tau)\frac{\partial}{\partial \xi}\left(\frac{1}{J(\xi, \tau)}\frac{\partial V}{\partial \xi}\right) \\ &+ (r - D_0)S(\xi, \tau)\frac{\partial V}{\partial \xi} - rJ(\xi, \tau)V = 0 \\ &V(\xi = 0, \tau) = 0 \\ &V(\xi = 1, \tau) = 0 \\ &V(\xi, \tau = T) = \max(S(\xi, T) - K, 0) \end{aligned} \quad (6.3)$$

where  $J(\xi, \tau) = \partial S(\xi, \tau)/\partial \xi$  is the spatial Jacobian, and  $J_t(\xi, \tau) = \partial S(\xi, \tau)/\partial \tau$  is the time Jacobian. We difference the transformed equation using second order central differencing in space and Crank-Nicolson differencing in time:

$$\begin{aligned} &\frac{J_j^{n+1}V_j^{n+1} - J_j^nV_j^n}{\Delta \tau} + \frac{J_{tj+1}^{n+1/2}(V_{j+1}^{n+1} + V_j^n) - J_{tj-1}^{n+1/2}(V_{j-1}^{n+1} + V_{j-1}^n)}{4\Delta \xi} \\ &+ \frac{\sigma^2}{4\Delta \xi^2} \left( S_j^{n+1} \right)^2 \left( \frac{V_{j+1}^{n+1} - V_j^{n+1}}{J_{j+1/2}^{n+1}} - \frac{V_j^{n+1} - V_{j-1}^{n+1}}{J_{j-1/2}^{n+1}} \right) \\ &+ \frac{\sigma^2}{4\Delta \xi^2} \left( S_j^n \right)^2 \left( \frac{V_{j+1}^n - V_j^n}{J_{j+1/2}^n} - \frac{V_j^n - V_{j-1}^n}{J_{j-1/2}^n} \right) + (r - D_0)S_j^{n+1} \frac{V_{j+1}^{n+1} - V_{j-1}^{n+1}}{4\Delta \xi} \\ &+ (r - D_0)S_j^n \frac{V_{j+1}^n - V_{j-1}^n}{4\Delta \xi} \\ &- \frac{r}{4} \left( J_j^{n+1} + J_j^n \right) \left( V_j^{n+1} + V_j^n \right) = 0 \end{aligned} \quad (6.4)$$

where

$$\begin{aligned}
 S_j^n &= S(\xi_j, \tau_n) \\
 J_{j+1/2}^n &= \frac{S_{j+1}^n - S_j^n}{\xi_{j+1} - \xi_j} = \frac{S_{j+1}^n - S_j^n}{\Delta\xi} \\
 J_{tj}^{n+1/2} &= \frac{S_j^{n+1} - S_j^n}{\Delta\tau} \\
 J_j^n &= \frac{J_{j+1/2}^n + J_{j-1/2}^n}{2}
 \end{aligned} \tag{6.5}$$

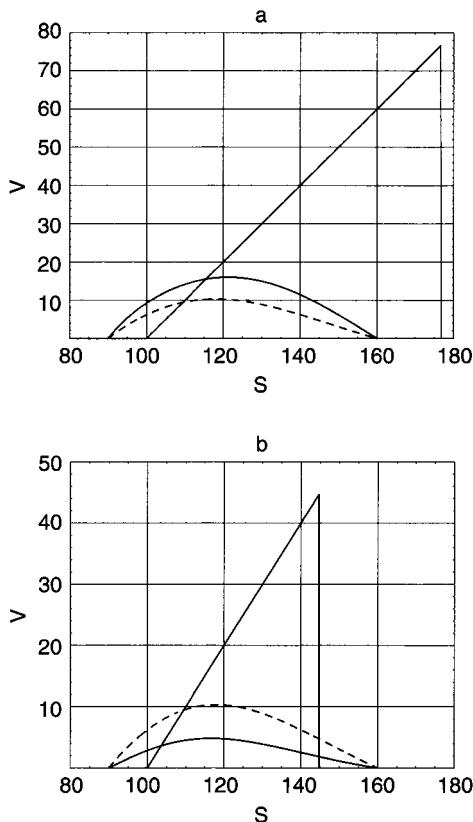
The boundary and expiration conditions are simply

$$\begin{aligned}
 V_0^n &= V_I^n = 0 \\
 V_j^N &= \max(S_j^N - K, 0) \quad 1 \leq j \leq I-1 \\
 V_0^N &= V_I^N = 0
 \end{aligned} \tag{6.6}$$

We can use a simple tridiagonal solver to solve the discretized system of equations.

Figure 6.1 displays finite difference results for both exponentially widening barriers and exponentially narrowing barriers. Shown are both expiration and present option values, and for comparison, present option value with stationary barriers fixed at the present positions. For widening barriers, present value is greater than the stationary barrier value, since the probability of knockout is reduced as time progresses. For narrowing barriers, the knockout probability increases with time and the stationary barrier value is greater. With  $I=200$  grid points and  $N=200$  time steps,  $V(S=95, t=0) = 5.3678$  for the widening option, within 0.0001 of its asymptotic value 5.3679.

In this example, we have specified the moving barrier trajectories as analytic functions. However, this need not be the case. The functions  $L(t)$  and  $U(t)$  could just as easily have been solutions to complex ordinary differential equations, or might have come from the solution of other PDEs (e.g.,  $U(t)$  could be a forward swap rate). Because we do not need an analytic functional form for the barrier trajectories, we can even use tabular data, interpolating when necessary.



**Figure 6.1** Results for a continuously sampled European corridor call with exponentially moving barriers. (a) Expiration and present option value with exponentially widening barriers,  $L(t) = L_0 \exp(-\delta_1 t)$  and  $U(t) = U_0 \exp(\delta_2 t)$  with  $U_0 = 160$ ,  $L_0 = 90$ , and  $\delta_1 = \delta_2 = 0.1$  (solid lines), and for comparison, present option value with stationary barriers  $U = U_0$  and  $L = L_0$  (dashed line). (b) Same as (a) except with exponentially narrowing barriers,  $L(t) = L_0 \exp(\delta_1 t)$  and  $U(t) = U_0 \exp(-\delta_2 t)$  (solid lines). Parameters common to both figures:  $T = 1$ ,  $K = 100$ ,  $r = 0.1$ ,  $\sigma = 0.25$ .

However, if the barrier positions move discontinuously, then  $J_t$  becomes infinite. In this case, the time dependent coordinate transformation can be used only during the continuous portion of barrier motion. At the times when barriers jump, the current option value can be interpolated onto a new grid, and the time integration begun

again. If the barrier motion includes only jumps and no continuous motion, then a "pinned" grid that encompasses all possible barrier positions can be used as illustrated in Chapter 5.

## Nonuniform Grids and Discrete Sampling

To illustrate the use of nonuniform grids, we price a European up-and-out call with a discretely sampled, fixed barrier. In modeling continuously sampled barriers, as in the previous example, it is sufficient to place the barriers on the boundaries and enforce a boundary condition of zero option value. It is of no consequence that the gradient of the option price is discontinuous at the barriers, even if the barriers are moving in time, because we never solve the pricing equation (which includes second derivative terms that might become singular) there. We always invoke a boundary condition instead.

To model discretely sampled barriers, however, we must place the barriers *inside* the solution region, allow the option value to "diffuse" over the barriers between sampling times, and enforce the knockout conditions only at the sampling times. Consequently, we are solving the pricing equation at the barrier itself and must resolve the very strong gradients that are periodically created there. To do this efficiently, we can use a coordinate transformation that results in a concentration of grid points near a barrier. In this example, the transformation is time independent because the single barrier is stationary, although one certainly can use a time-dependent nonuniform grid for discretely sampled moving barriers.

We denote the stationary up-out barrier by  $B$  and make use of the transformation discussed in Chapter 5, Equation 5.15:

$$S(\xi) = B + \alpha \sinh(c_1(1 - \xi) + c_2\xi) \quad (6.7)$$

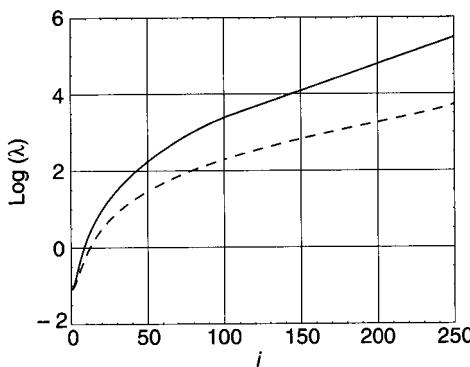
where

$$\begin{aligned} c_1 &= \sinh^{-1}(S_{\min} - B)/\alpha \\ c_2 &= \sinh^{-1}(S_{\max} - B)/\alpha \end{aligned} \quad (6.8)$$

and where  $\alpha$  is a parameter that determines the uniformity of the  $S$  grid.  $\Delta S$  reaches a minimum at  $\xi_B$ , where  $S(\xi_B) = B$ . Setting  $\alpha \ll$

$S_{\max} - S_{\min}$  yields a highly nonuniform grid, while  $\alpha \gg S_{\max} - S_{\min}$  yields a uniform grid.

The periodic enforcement of the knockout condition produces discontinuities in the option value as the calculation proceeds. The discontinuities correspond to excitation of the eigenmodes with the largest eigenvalues. Because the grid spacing becomes very small near the barrier, the largest eigenvalues  $|\lambda_i|$  of the spatial discretization can become exceedingly large. In Figure 6.2, we display the eigenvalues  $|\lambda_i|$  of the discretized pricing equation with a nearly uniform grid  $\alpha = 2000$  (dashed curve) and with a very nonuniform grid  $\alpha = 2$  (solid curve). Here  $S_{\min} = 0$ ,  $S_{\max} = 150$ , and  $I = 250$  grid points. Other parameters are in the figure caption. With the nonuniform grid, the largest eigenvalues are increased by a factor of about 50. We expect them to scale approximately with the inverse square of the smallest  $\Delta S$  on the grid. For  $\alpha = 2$ , the smallest  $\Delta S$  is about one-tenth that for a uniform grid. Thus the expected scaling is verified. The smallest eigenvalues are unaffected by the coordinate transformation. These small eigenvalues depend on the size of the grid  $S_{\max} - S_{\min}$ , not on the details of how it is subdivided.

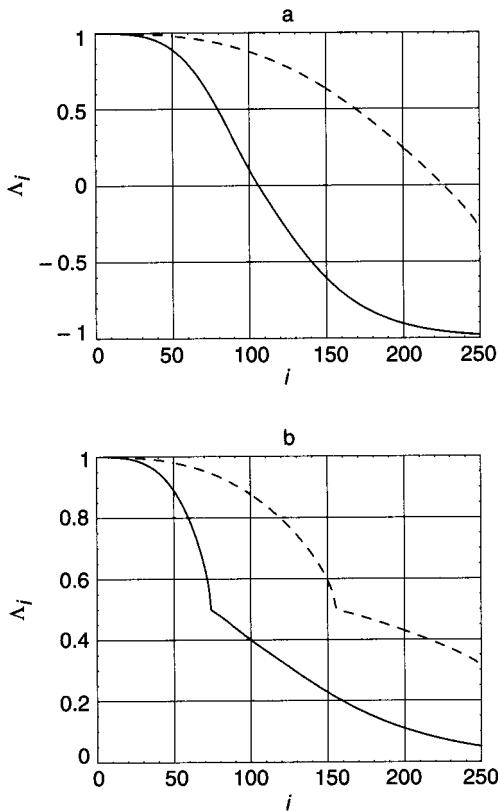


**Figure 6.2** Eigenvalues  $\log_{10}(|\lambda_i|)$  and for a second order central difference discretization of the Black-Scholes equation with the coordinate transformation Equation 5.1.2. A grid of  $I = 250$  grid points is used. Nearly uniform grid  $\alpha = 2000$  (dashed curve), very nonuniform grid  $\alpha = 2$  (solid curve). Other parameters are:  $D_0 = 0.02$ ,  $r = 0.07$ ,  $\sigma = 0.2$ ,  $S_{\min} = 0$ ,  $S_{\max} = 150$ ,  $B = 125$ .

This presents a problem for Crank-Nicolson time discretization. Although it is unconditionally stable, the amplification factor for such modes, as shown in Chapter 3, approaches  $\Lambda = -1$ . Hence these modes, periodically excited by the barrier condition, decay very slowly, while simply oscillating in time. Another way to think of this is that when the local  $\Delta S$  is very small, then the characteristic grid diffusion time,  $\tau_d = \Delta S^2 / (\sigma S)^2$  becomes extremely small. A time step that is appropriate for most of the grid, where the local  $\Delta S$  is larger, may result in  $\Delta t \gg \tau_d$  near the barrier and hence give rise to this oscillatory behavior.

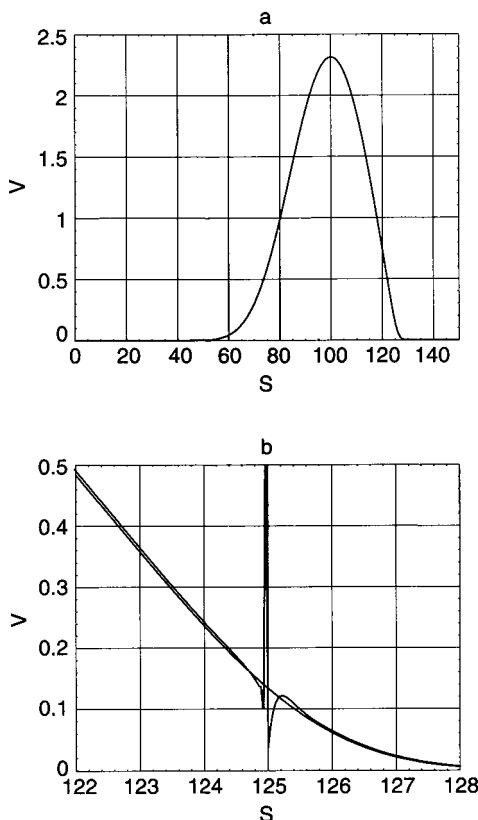
In Figure 6.3(a), we display the Crank-Nicolson amplification factors  $\Lambda_i$  corresponding to the eigenvalues in Figure 6.2. The amplification factors are calculated for  $N = 1500$  time steps per year. If the barrier is sampled daily, then this represents about 6 time steps between barrier samplings. For the uniform grid (dashed curve), the large eigenvalue modes have an amplification factor of about  $\Lambda = -0.3$ . Thus these modes decay away in the time steps between barrier samplings and present no problems. For the highly nonuniform grid, however, the amplification factors of the largest eigenvalues approach  $\Lambda = -1$  quite closely. They do not decay significantly between barrier samplings, but rather, simply accumulate. On the other hand, the amplification factors of the three-level time discretization method shown in Figure 6.3(b) approach  $\Lambda = 0$  for the most troublesome modes, which decay dramatically between barrier samplings. Again, we are not describing the decay of these modes *accurately*. The finite difference grid is by definition unable to do so. Fortunately, this is not necessary. These modes do not contribute to the final solution, so we simply want them to go away, leaving the lower eigenvalue modes, which the finite difference scheme does describe accurately. The eigenvalue view of the problem and the spot price view of the problem are dual views. Now we examine the numerical results for option value in the more familiar spot price view.

Figure 6.4 displays finite difference results for a knockout call with  $K = 100$ ,  $B = 125$ , and 250 equally spaced sampling dates. The last sample is at expiration. We have used just 500 grid points in  $S$ , with  $S_{\min} = 0$  and  $S_{\max} = 150$ , but have set  $\alpha = 2 \ll S_{\max}$ , so that the coordinate transformation concentrates grid points near the barrier  $B$ . The present option value is displayed in Figure 6.4(a) as a function of spot price over the entire finite difference grid. Figure 6.4(b) compares the



**Figure 6.3** Amplification factors  $\Lambda_i$  for  $N = 1500$  time steps per year corresponding to the eigenvalues in Figure 6.2. Dashed curves are for a uniform grid,  $\alpha = 2000$  and solid curves are for a highly nonuniform grid,  $\alpha = 2$ . (a) Crank-Nicolson discretization. (b) Three-level time discretization. Parameters used:  $T = 1$ ,  $K = 100$ ,  $D_0 = 0.02$ ,  $r = 0.07$ ,  $\sigma = 0.2$ ,  $S_{\max} = 150$ ,  $B = 125$ ,  $N = 1500$ .

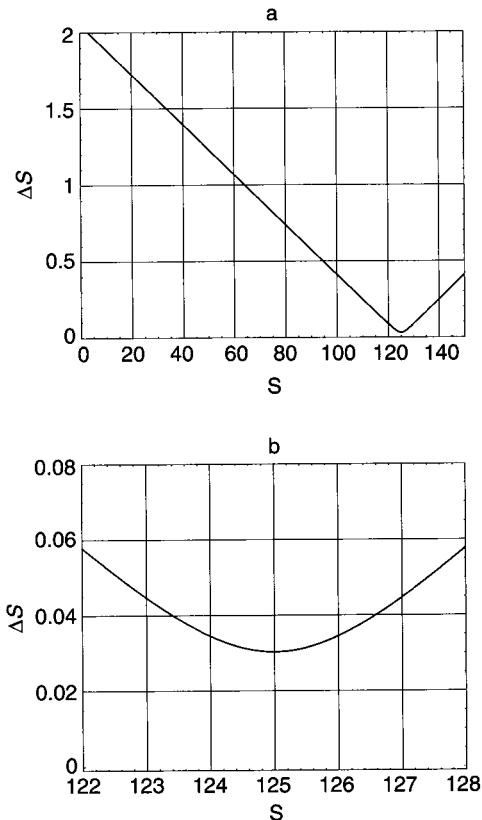
present option value near the barrier for Crank-Nicolson and three-level time discretizations. In both cases, there are  $N = 1500$  time steps, about 6 time steps per barrier sampling increment. The value of the option is small but finite for  $S \geq B$ , reflecting the possibility that the price will move below the barrier before the first sampling occurs. However, the three-level scheme gives a smooth solution, while with Crank-Nicolson, a spike appears near the barrier. This spike does not reflect numerical instability, but rather that the very sharp gradients



**Figure 6.4** Results for a discretely sampled European knockout call. (a) Present option value with a 500-point nonuniform grid ( $\alpha = 2$ ). (b) Detail of present option value with Crank-Nicolson and the three-level time discretization. Parameters are:  $T = 1$ ,  $K = 100$ ,  $D_0 = 0.02$ ,  $r = 0.07$ ,  $\sigma = 0.2$ ,  $S_{\max} = 150$ ,  $B = 125$ ,  $N = 1500$ .

that are periodically produced by setting the option value to zero for  $S \geq B$  on barrier sample dates cannot decay fast enough in the Crank-Nicolson solution. The spike remains localized, but even so, it reduces diffusion through the barrier and results in an option value that is everywhere slightly too large.

The effective grid spacing,  $\Delta S = (\partial S / \partial \xi) \Delta \xi$ , is displayed in Figure 6.5(a). Near the barrier at  $B = 125$ , the grid spacing is nearly flat, but for  $|S - B| \gg \alpha$  it becomes proportional  $|S - B|$ . There is a ratio of



**Figure 6.5** Results for a discretely sampled European knockout call. (a) Effective grid spacing  $\Delta S$  of the nonuniform grid. (b) Detail of effective grid spacing  $\Delta S$  of the nonuniform grid. Parameters are:  $T = 1$ ,  $K = 100$ ,  $D_0 = 0.02$ ,  $r = 0.07$ ,  $\sigma = 0.2$ ,  $S_{\max} = 150$ ,  $B = 125$ ,  $N = 1500$ .

about 60 between the largest and smallest spacing on the grid. The grid spacing near the barrier is about one-tenth that of a uniform grid of the same dimension. Figure 6.5(b) is a detailed plot of the effective grid spacing near the barrier.

As shown in Chapter 5, the position of barriers relative to the finite difference grid has a profound effect on the accuracy and convergence rate of the numerical solutions. Table 6.1 displays present option value and the error in option value at a series of spot prices as a function of the number of grid points  $I$  for  $\epsilon = 2000$  (a nearly

**Table 6.1 Knockout Call: Uniform Grid**

<i>S/I</i>	250	500	1000	2000	4000	$\infty$
<i>Present Option Value, Barrier at Grid Point</i>						
100	2.25793	2.28584	2.30142	2.30946	2.31356	2.31766
110	1.79641	1.82742	1.84484	1.85386	1.85846	1.86307
120	0.69263	0.71584	0.72895	0.73575	0.73922	0.74270
125	0.10970	0.12251	0.12965	0.13334	0.13523	0.13712
<i>Error in Present Value <math>\times 10^4</math>, Barrier at Grid Point</i>						
100	-597.242	-318.198	-162.336	-81.937	-40.989	
110	-666.562	-356.460	-182.220	-92.067	-46.083	
120	-500.744	-268.603	-137.554	-69.564	-34.838	
125	-274.147	-146.037	-74.6539	-37.743	-18.894	
<i>Present Option Value, Barrier at Midpoint</i>						
100	2.32436	2.31937	2.31812	2.31782	2.31774	2.31771
110	1.87081	1.86505	1.86360	1.86325	1.86316	1.86313
120	0.74853	0.74420	0.74311	0.74284	0.74277	0.74275
125	0.13891	0.13758	0.13725	0.13717	0.13715	0.13715
<i>Error in Present Value <math>\times 10^4</math>, Barrier at Midpoint</i>						
100	66.467	16.597	4.093	1.020	0.254	
110	76.798	19.199	4.727	1.177	0.293	
120	57.752	14.446	3.551	0.884	0.220	
125	17.665	4.377	1.071	0.267	0.066	

Present option value and the error in option value at a series of spot prices as a function of the number of grid points  $I$  for a nearly uniform grid  $\alpha = 2000$ . Parameters are:  $T = 1$ ,  $K = 100$ ,  $D_0 = 0.02$ ,  $r = 0.07$ ,  $\sigma = 0.20$ ,  $S_{\max} = 150$ ,  $B = 125$ ,  $N = 1500$ .

uniform grid), while Table 6.2 displays the same results but for  $\alpha = 2$  (the grid used for the calculation of Figure 6.4). For option values, the column labeled  $\infty$  is the extrapolated infinite grid result obtained by a best fit of the values for  $I = \{250, 500, \dots, 4000\}$  to the equation  $V(I) = V(\infty) - C/I^q$ , where  $C$  is a constant, and  $q$  is the convergence rate exponent. Similarly, the errors in option value are obtained by subtracting this infinite grid result from calculated present option value (since there is no analytic result with which to compare).

In each table, results are shown when the barrier lies on a grid point, and when it lies midway between grid points. The grid

**Table 6.2 Knockout Call: Nonuniform Grid**

S/I	250	500	1000	2000	4000	$\infty$
<i>Present Option Value, Barrier at Grid Point</i>						
100	2.31187	2.31446	2.31600	2.31684	2.31727	2.31771
110	1.85610	1.85935	1.86118	1.86214	1.86263	1.86313
120	0.73705	0.73980	0.74125	0.74199	0.74237	0.74275
125	0.13392	0.13551	0.13632	0.13673	0.13694	0.13715
<i>Error in Present Value <math>\times 10^4</math>, Barrier at Grid Point</i>						
100	-58.444	-32.585	-17.137	-8.782	-4.443	
110	-70.277	-37.782	-19.555	-9.945	-5.013	
120	-57.028	-29.541	-15.030	-7.580	-3.806	
125	-32.260	-16.366	-8.243	-4.137	-2.072	
<i>Present Option Value, Barrier at Midpoint</i>						
100	2.31903	2.31805	2.31780	2.31773	2.31772	2.31771
110	1.86417	1.86339	1.86320	1.86315	1.86314	1.86313
120	0.74315	0.74285	0.74278	0.74276	0.74275	0.74275
125	0.13722	0.13716	0.13715	0.13714	0.13715	0.13715
<i>Error in Present Value <math>\times 10^4</math>, Barrier at Midpoint</i>						
100	13.168	3.316	0.836	0.210	0.052	
110	10.341	2.602	0.656	0.165	0.041	
120	3.963	0.997	0.251	0.063	0.016	
125	0.722	0.181	0.046	0.012	0.003	

Present option value and the error in option value at a series of spot prices as a function of the number of grid points  $I$  for a nonuniform grid  $\alpha = 2$ . Parameters are:  $T = 1$ ,  $K = 100$ ,  $D_0 = 0.02$ ,  $r = 0.07$ ,  $\sigma = 0.20$ ,  $S_{\max} = 150$ ,  $B = 125$ ,  $N = 1500$ .

stretching procedures discussed in Chapter 5 are used to achieve this. When the barrier lies on a grid point, the error decreases inversely with grid size,  $q = 1$ . This is an unhappy circumstance because the intrinsic truncation error of the finite difference scheme is  $O(\Delta S^2)$ , while we are achieving only  $O(\Delta S)$ . However, when the grid is arranged so that the barrier falls midway between grid points, error is reduced drastically, and the theoretical  $O(\Delta S^2)$  scaling of error is achieved. The data clearly show that  $q = 2$  for both uniform and nonuniform grids.

When the barrier is on a grid point, the errors obtained on the nonuniform grid are almost uniformly smaller by a factor of 10 than

those on the uniform grid. This is expected since the  $\Delta S$  in the neighborhood of the barrier is reduced by about a factor of 10 and we have shown that error scales as  $O(\Delta S)$ . When the barrier is midway between grid points, the relative error reduction obtained using a nonuniform grid is a function of spot price, ranging from a factor of 5 to 20 in the example data. We conclude that while the clustering coordinate transformation adds only marginally to the execution time of the computational model (about 10% in this example), it results in about a factor 10 increase in accuracy. Secondly, placing a discretely sampled barrier midway between grid points increases the convergence rate of the finite difference calculation from  $O(\Delta S)$  to  $O(\Delta S^2)$ .

In Chapter 5 and earlier in this chapter, we priced several types of *continuously* sampled barrier options by placing the barrier prices exactly on finite difference grid points. In this last example, we showed that putting a *discretely* sampled barrier midway between grid points leads to increased accuracy and improved convergence. Of course, as the sampling interval is decreased, a discretely sampled barrier becomes a continuously sampled one. We therefore need a characteristic time to which to compare the sampling rate that can tell us where to place barriers on the grid for optimal accuracy, at grid points or midway between.

An obvious choice is the characteristic grid diffusion time near the barrier

$$\tau_d(B) = \Delta S^2(B) / (\sigma B)^2 \quad (6.9)$$

When the sampling interval is much less than this characteristic time  $\delta t_s \ll \tau_d(B)$ , then the discretely sampled barrier is continuously sampled to the resolution of the grid, and we should place the barrier prices at grid points. When the sampling interval is much greater than this characteristic time  $\delta t_s \gg \tau_d(B)$ , then the barrier prices should be midway between grid points for optimal accuracy.

We can use this  $\tau_d(B)$  versus  $\delta t_s$  criterion to choose where to place a discretely sampled barrier for optimal accuracy. However, it is easy to see that for almost any discretely sampled barrier option we should put the barriers midway between grid points. To this end it is instructive to calculate  $\tau_d(B)$  for some typical parameters.

If we choose the uniform grid with  $I = 1000$  in Table 6.1, we find  $\tau_d(B) \approx 10^{-4}$  year. Since the approximate daily sampling interval is  $\delta t_s = 0.004 \approx 40\tau_d(B)$ , we achieve best results with the barrier midway between grid points. For the nonuniform grid of Table 6.2,  $\delta t_s \approx 4000\tau_d(B)$ , an even stronger case for putting  $B$  between grid points.

If we do the same calculation but with a sampling interval of  $\delta t_s = 10^{-5}$  year, then the behavior we find in Tables 6.1 and 6.2 is actually reversed. The barrier becomes continuously sampled to the resolution of the uniform grid, and quadratic  $O(\Delta S^2)$  convergence is achieved with the barrier exactly on a grid point. When the barrier is midway between grid points, we are effectively calculating the value of a continuously sampled barrier option with a barrier at  $B + \Delta S(B)/2$ . Hence, the accuracy is reduced and the convergence rate is reduced to  $O(\Delta S)$ . For the nonuniform grid of Table 6.2, the sampling rate must be reduced below  $\delta t_s = 10^{-7}$  year for the optimal position of the barrier to be at a grid point.

Thus, for typical parameters, the sampling interval for which placing barriers on grid points becomes optimal is so small that simply modeling the option as continuously sampled to begin with is probably sufficiently accurate. It is surely much more efficient, because the time steps can be much larger than  $\delta t_s$ .

#### DISCRETELY SAMPLED PARISIAN OPTIONS

Barrier options are subject to short-term price manipulation of the underlying in order to trigger or avoid a knockin or knockout event. In addition, they are very difficult to hedge when spot price is near a barrier due to the large *l's* and *r's*. The Parisian option provides a remedy for both problems. In this type of option, knockin or knockout only occurs if the spot price of the underlying has been above or below the barrier price for a prescribed continuous length of time (Chesney, Cornwall, Jeanblanc-Picque, Kentwell, & Yor, 1997; Haber, Schonbucher, & Wilmott, 1999), or in the case of a discretely sampled Parisian option, for a prescribed number of *contiguous* samples.

Therefore, the Parisian option adds a new element of path dependency to the barrier option. Haber et al. (1999), describe a partial

differential equation pricing model for the continuously sampled Parisian option. In this section, we describe the corresponding discretely sampled model. Again we solve the one-dimensional Black-Scholes pricing equation:

$$\frac{\partial V}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} + (r - D_0) S \frac{\partial V}{\partial S} - rV = 0 \quad (6.10)$$

but with an additional parameter,  $L$ , the number of contiguous samples for which spot price has been in violation of the barrier condition. On sampling dates, the jump condition

$$V(S, L, t_i^-) = V(S, L_{\text{new}}, t_i) \quad (6.11)$$

is enforced, where for a Parisian knockout option

$$\begin{aligned} L_{\text{new}} &= \max(L + 1, L_{\max}) & S \geq B \\ L_{\text{new}} &= 0 & S < B \end{aligned} \quad (6.12)$$

Thus, if the underlying spot price is in violation of the barrier condition at a sample time (e.g., if  $S \geq B$  for a knockout), the counter  $L$  is incremented by one. However, if  $L$  ever reaches  $L_{\max}$ , the number of contiguous samples that causes a knockout, then it remains there. Since the option is knocked out, there is no point in accumulation of further barrier violations. If the spot price is below the barrier, then the counter is reset to 0—any existing string of contiguous violations has been broken. This feature acts in concert with the expiration condition

$$\begin{aligned} V(S, L, T) &= \max(S - K, 0) & L < L_{\max} \\ V(S, L, T) &= 0 & L = L_{\max} \end{aligned} \quad (6.13)$$

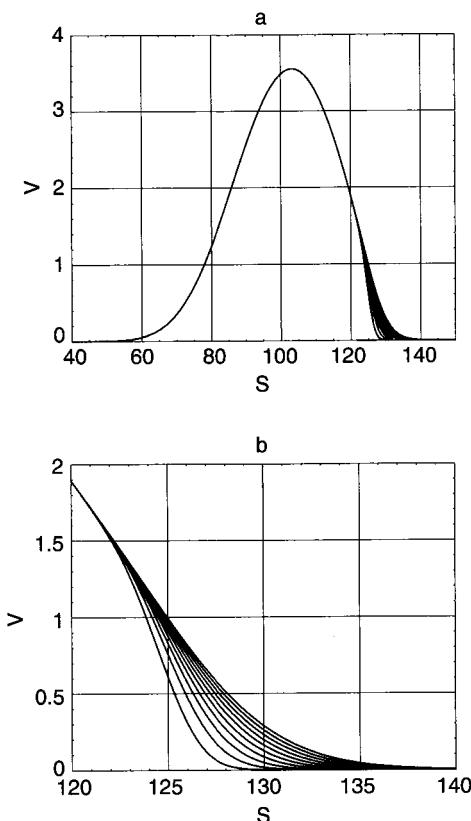
A glance at Equation 6.13 shows that the expiration condition is independent of  $L$  for  $L < L_{\max}$ , then drops to zero for  $L = L_{\max}$ . There is no interpolation scheme that we can use in the implementation of the jump condition that will adequately handle this discontinuity of slope. Linear interpolation will result in excessive smoothing, while

cubic splines will give rise to over- and undershoot ripples. Therefore, we simply make  $J = L_{\max}$ , where  $J$  is the number of grid points in the  $L$  direction. Then the jump condition will not require any interpolation at all, but will result in simple shifts of option value by integral numbers of grid points along the  $L$  direction. The error resulting from such shifts is zero, regardless of the  $L$  dependence of the expiration condition.

On those sampling dates when  $L$  is incremented to  $L_{\max}$ , the current option value will be replaced with  $\alpha$ , as can be seen from Equation 6.13, periodically creating the same type of discontinuities as in the previous discretely sampled barrier option example. Therefore, we use the same coordinate transformation in 5, Equation 5.15 to resolve these discontinuities.

Figure 6.6 shows results for a discretely sampled Parisian knockout call option for which  $L_{\max} = 10$  samples are required for knockout. The option is struck at  $K = 10a$ , expires in  $T = 1$  year, and spot price is sampled at 250 equally spaced intervals, ending with expiration. Figure 6.6(a) is the present option value,  $V(5, L, t=0)$  for options from  $L = \alpha$  to  $L = 9$  current barrier violations. Because knockout requires 10 contiguous samples with  $5 \sim B$ , the option value is larger than that for the simple discretely sampled knockout call shown in Figure 6.4(a). For spot prices far below the barrier, the option value is independent of  $L$ . In this region, there is only a minuscule probability that spot price will have moved above the barrier in one day's time. Therefore, it is nearly certain that  $L$  will get reset to  $\alpha$  on the next sample, and an option with  $L = 9$  has the same value as one with  $L = 0$ . For spot prices near or above the barrier, this is not true, and there is a strong dependence of option value on current  $L$ . Figure 6.6(b) shows this region near the barrier in detail. The option with lowest value is that with  $L = 9$ , which will be knocked out if the next sampled spot price violates the barrier condition  $5 \sim B = 125$ .

Table 6.3 displays present option value for a Parisian call option that is knocked out at  $L_{\max} = 10$  as a function of the current number of contiguous barrier violations  $L$  for a series of spot prices. The calculations were done with the same coordinate transformation used for the previous discretely sampled knockout call example, and with  $1 = 1000$  grid points. For  $5 = 10a$ , option value is independent of  $L$  to seven digits. However, with spot price exactly on the barrier,  $5 = 125$ , option value is a strong function of  $L$ . In this case, the probability of



**Figure 6.6** Results for a discretely sampled Parisian call that is knocked out after  $L_{\max} = 10$  contiguous samples of  $S \leq B$  have been recorded. (a) Present option value for  $L = \{0, 1, \dots, 9\}$  plotted over large range of spot price. (b) Detail near the barrier  $B = 125$ . Parameters are:  $T = 1$ ,  $K = 100$ ,  $D_0 = 0.02$ ,  $r = 0.07$ ,  $\sigma = 0.2$ ,  $S_{\min} = 0$ ,  $S_{\max} = 150$ ,  $B = 125$ ,  $I = 1000$ ,  $J = L_{\max} = 10$ ,  $N = 1500$ ,  $\alpha = 2$ .

spot price being above the barrier on the next sample is about 50%. Thus option value for  $L = 9$ , which will be knocked out in this circumstance, is about half that for  $L = 0$ .

Even so, this  $L = 9$  Parisian option, is still about four times more valuable near the barrier than the simple discretely sampled knock-out call shown in Figure 6.4(b). The reason is that if the underlying spot price of the Parisian option were to fall below the barrier on the

**Table 6.3 Parisian Knockout Call**

L/S	Present Option Value vs. Knockout Days			
	100	120	125	130
0	3.48481	1.89271	0.99370	0.288118
1	3.48481	1.89269	0.98241	0.259734
2	3.48481	1.89268	0.96919	0.229076
3	3.48481	1.89265	0.95341	0.195995
4	3.48481	1.89262	0.93407	0.160474
5	3.48481	1.89259	0.90957	0.122803
6	3.48481	1.89254	0.87703	0.083986
7	3.48481	1.89246	0.83071	0.046572
8	3.48481	1.89231	0.75676	0.016152
9	3.48481	1.89190	0.60862	0.001339
10	0	0	0	0

Present option value of a discretely sampled Parisian knockout call option ( $L_{\max} = 10$ ) at a series of spot prices as a function of the current number of contiguous barrier violation dates  $L$ . Parameters are:  $T = 1$ ,  $K = 100$ ,  $D_0 = 0.02$ ,  $r = 0.07$ ,  $\sigma = 0.20$ ,  $S_{\max} = 150$ ,  $B = 125$ ,  $I = 1000$ ,  $N = 1500$ ,  $\alpha = 2$ .

next sample date, the counter is reset, and the option cannot be knocked out until  $L_{\max} = 10$  more contiguous samples are recorded for which  $S \geq B$ . The simple discretely sampled barrier option is also spared from knockout if the next sample falls below the barrier, but its value is much lower because it can subsequently be knocked out by a single sample above the barrier. One can think of a discretely sampled knockout option as a Parisian option with  $L_{\max} = 1$ . Finally, far above the barrier, at  $S = 130$ , the probability that the next sample will be below the barrier, resetting  $L$  to 0, is minuscule and the value of a nearly knocked out option,  $L = 9$  is about 1/200 that of one with no current barrier violations.

Table 6.4 displays present option value at a series of spot prices for Parisian knockout call options with no current barrier violations,  $L = 0$ , as a function of  $L_{\max}$ , the number required for knockout. Because the calculations were done with the same coordinate transformation used for the previous discretely sampled knockout call example and with  $I = 1000$  grid points, the option values for  $L_{\max} = 1$  agree exactly with those in the appropriate column in Table 6.2. (The barrier is midway between two grid points for maximum accuracy.)

**Table 6.4 Parisian Knockout Call**

$S/L_{\max}$	Present Option Value vs. Knockout Days					
	1	2	3	4	5	10
100	2.31780	2.53206	2.69584	2.83658	2.96313	3.48481
110	1.86320	2.10496	2.29409	2.45953	2.61056	3.25517
120	0.74278	0.92685	1.07589	1.20968	1.33445	1.89271
125	0.13715	0.25636	0.36165	0.45989	0.55383	0.99370

Present option value of a discretely sampled Parisian knockout call option just beginning ( $L = 0$ ) at a series of spot prices as a function of the number of contiguous barrier violations required for knockout  $L_{\max}$ . Parameters are:  $T = 1$ ,  $K = 100$ ,  $D_0 = 0.02$ ,  $r = 0.07$ ,  $\sigma = 0.20$ ,  $S_{\max} = 150$ ,  $B = 125$ ,  $I = 1000$ ,  $N = 1500$ ,  $\alpha = 2$ .

Option value well below the barrier increases slowly with  $L_{\max}$ . At the barrier, the scaling with  $L_{\max}$  is roughly linear.

If we make a small change to the Parisian option jump condition Equation 6.12

$$L_{\text{new}} = L + 1 \quad S \geq B \quad (6.14)$$

then we change the nature of the option significantly. When we eliminate the resetting of  $L$  to zero when the knockout condition is not met, then  $L$  simply counts the number of samples for which the condition has been met over the life of the option, contiguous or not. If the payoff condition remains

$$\begin{aligned} V(S, L, T) &= \max(S - K, 0) & L < L_{\max} \\ V(S, L, T) &= 0 & L = L_{\max} \end{aligned} \quad (6.15)$$

then the option pays according to the cumulative number of samples  $L$  meeting the knockout condition. Chesney et al. (1997) call this type of option a "Parasian" option since it combines features of the Parisian option with the averaging characteristics of an Asian option. Additional "path dimensions" can be added, for example  $M$ , a variable that counts cumulative or contiguous numbers of samples above or below a second barrier. The path dimensions  $L$  and  $M$  can be combined in arbitrary ways with  $S$  and  $K$  to provide a rich variety of payoffs.

### A LEVERAGED KNOCKIN PUT

In our next example, a path dimension  $L$ , which counts cumulative discrete samples for which a knockin condition is met is used to leverage the payoff of a standard continuously sampled knockin put. This leveraged version has a continuously monitored knockin barrier  $X_2$  and, in addition, a second, discretely sampled knockin barrier  $X_1$ . The standard knockin payoff is multiplied by the cumulative number of samples,  $L$ , for which the spot price is below the lower barrier  $X_1$ .

$$\begin{aligned} V(S, L, T) &= L \max(K - S, 0) & S^* \leq X_2 \\ V(S, L, T) &= 0 & S^* > X_2 \end{aligned} \quad (6.16)$$

where  $S^*$  is the minimum continuously monitored spot price recorded during the life of the option. That is, if  $S^* \leq X_2$ , the option is knocked in and its payoff becomes that of a standard put option (leveraged by  $L$ ). If  $S^* > X_2$ , the option is never knocked in and expires worthless. Although such an option may seem at first glance difficult to price using PDE methods, it is actually straightforward. First consider the continuous knockin feature associated with the barrier  $X_2$ , ignoring the leveraging feature for the moment.

A standard knockin barrier option has no value until the spot price touches the barrier  $X_2$ , at which time the option becomes a standard put option. If the barrier is continuously sampled, then this condition is enforced at each instant of time. Theoretically then, the barrier  $X_2$  is a boundary, with the boundary condition

$$V_{ki}(X_2, t) = V_{sp}(X_2, t) \quad (6.17)$$

where  $V_{ki}(S, t)$  is the value of the knockin option and  $V_{sp}(S, t)$  is the value of the standard put option.

In principle, we could use the Black-Scholes formula for the value of the standard put option as a boundary condition in our knockin barrier option. However, this approach lacks generality. What happens when we want to add discrete dividends of variable amounts, interest rate or dividend yield term structure, a volatility surface  $\sigma(S, t)$ , or any number of other complicating features?

In these common situations, the required analytic formulas do not exist.

A much more general approach for knockins is to simply add another pricing equation. The additional equation gives the value of the standard option that the knockin option becomes when knocked in. This additional pricing equation includes any of the complicating features previously mentioned. For a knockin put, we would have

$$\begin{aligned}\frac{\partial V_{\text{sp}}}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 V_{\text{sp}}}{\partial S^2} + (r - D_0) S \frac{\partial V_{\text{sp}}}{\partial S} - rV_{\text{sp}} &= 0 \\ \frac{\partial V_{\text{ki}}}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 V_{\text{ki}}}{\partial S^2} + (r - D_0) S \frac{\partial V_{\text{ki}}}{\partial S} - rV_{\text{ki}} &= 0\end{aligned}\quad (6.18)$$

with expiration conditions

$$\begin{aligned}V_{\text{sp}}(S, T) &= \max(K - S, 0) \\ V_{\text{ki}}(S, T) &= V_{\text{sp}}(S, T) \quad S \leq X_2 \\ V_{\text{ki}}(S, T) &= 0 \quad S > X_2\end{aligned}\quad (6.19)$$

and knockin condition

$$V_{\text{ki}}(S, t) = V_{\text{sp}}(S, t) \quad S \leq X_2 \quad (6.20)$$

We can use the BC1 linearity boundary conditions discussed in Chapter 4 at  $S_{\min}$ , assuming we have made  $S_{\min}$  sufficiently small so that linearity holds. A simple  $V_{\text{sp}} = V_{\text{ki}} = 0$  boundary condition suffices at  $S_{\max}$  again assuming it is large enough. Simple rules of thumb for "large enough" and "small enough" follow from the pertinent boundary condition discussions in Chapter 4.

Although we are now solving two equations, substantial savings result and the computational cost is less than the factor of two that one might first suppose. Because the pricing equations are identical, the matrix  $A$  that embodies the discretized pricing equation need only be calculated once, for the standard put  $V_{\text{sp}}$ , which is updated first. That is, we solve

$$AV_{\text{sp}}^{n+1} = b_{\text{sp}} \quad (6.21)$$

Then the standard put option value required for the knockin calculation is available. Next, a modified version of the same matrix  $\hat{A}$  is used for the knockin option value

$$\hat{A}V_{ki}^{n+1} = \hat{b}_{ki} \quad (6.22)$$

The modifications that transform  $A$  into  $\hat{A}$  and  $b_{ki}$  into  $\hat{b}_{ki}$  are simple enough. We simply make all rows of  $A$  corresponding to  $S \leq X_2$ , equal to the corresponding rows of the identity matrix.

$$\begin{aligned}\hat{A}_{ii} &= 1 & S_i \leq X_2 \\ \hat{A}_{ij} &= 0 \quad (i \neq j) & S_i \leq X_2 \\ \hat{A}_{ij} &= A_{ij} & S_i > X_2\end{aligned}\quad (6.23)$$

Similarly, we simply make all elements of  $\hat{b}_{ki}$  corresponding to  $S \leq X_2$ , equal to  $V_{sp}^{n+1}$ .

$$b_{ki,i} = V_{sp,i}^{n+1} \quad S_i \leq X_2 \quad (6.24)$$

These modifications are the discrete embodiment of the knockin condition Equation 6.20:

$$V_{ki}^{n+1} = V_{sp}^{n+1} \quad S_i \leq X_2 \quad (6.25)$$

A completely equivalent approach to that described here would be to solve for both the standard option and the corresponding knockout option as described earlier in this chapter, calculating the value of the knockin option as the difference of these two. This approach also could be implemented to share arithmetic between the two option calculations. Finally, when there is a rebate, a third equation for the discounted rebate value also can be solved, again sharing computations to reduce arithmetic.

We can now add the "leveraging" path dependency feature to the knockin solution. This is done by modifying the expiration conditions

$$\begin{aligned}
 V_{sp}(S, L, T) &= L \max(K - S, 0) \\
 V_{ki}(S, L, T) &= V_{sp}(S, L, T) \quad S \sim X_1 \\
 V_{ki}(S, L, T) &= 0 \quad S > X_2
 \end{aligned} \tag{6.26}$$

and imposing the jump conditions

$$\begin{aligned}
 V_{sp}(S, L, t_i) &= V_{sp}(S, L + I, t_i) \quad S \sim X_1 \\
 V_{ki}(S, L, t_i^-) &= V_{ki}(S, L + I, t_i) \quad S \sim X_1
 \end{aligned} \tag{6.27}$$

for both knockin and standard put option values.

An interesting nuance of this particular example is that the value of the leveraged option is actually linear in  $L$ . This is easy to see; the expiration condition is linear in  $L$  and the jumps in  $L$  that occur at sampling dates are of a fixed size, preserving the initial linearity. As a consequence, one can calculate the value of this option with only two grid points in the  $L$  direction, two points being sufficient to define any linear function. A linear interpolation is sufficient to implement the jump conditions in this case.

Another common feature of the jump condition for this option is that it requires information, specifically option values, from regions that are not covered by the finite difference grid itself. Thus, the interpolation scheme used in implementing the jump condition will sometimes be asked to do extrapolation.

We define a finite difference grid  $S_{\min} \sim S \sim S_{\max}$  and  $L_{\min} \sim L \sim L_{\max}$  and assign an expiration option value to each point on the grid. We integrate backward in time from expiration  $T$  to a sampling time,  $t_i$ . After completing the time step that brings the calculation to  $t_i$  we execute the jump condition. The jump conditions, Equation 6.27 will attempt to make the following replacements:

$$\begin{aligned}
 V_{sp}(S, L_{\max}, t_i) &= V_{sp}(S, L_{\max} + I, t_i) \quad S \sim X_1 \\
 V_{ki}(S, L_{\max}, t_i^-) &= V_{ki}(S, L_{\max} + I, t_i) \quad S \sim X_1
 \end{aligned} \tag{6.28}$$

Since, as we have already shown, the option value is linear in  $L$ , we can simply obtain the required option value by linear extrapolation:

$$\begin{aligned} V_{\text{sp}}(S, L_{\max}, t_i^-) &= 2V_{\text{sp}}(S, L_{\max}, t_i) - V_{\text{sp}}(S, L_{\max} - 1, t_i) \\ V_{\text{ki}}(S, L_{\max}, t_i^-) &= 2V_{\text{ki}}(S, L_{\max}, t_i) - V_{\text{ki}}(S, L_{\max} - 1, t_i) \end{aligned} \quad (6.29)$$

In doing this extrapolation, we have synthesized information from outside the finite difference grid, by making an assumption about its behavior near the  $L_{\max}$  boundary. This is the discrete analog of the boundary condition we would have to provide at the  $L_{\max}$  boundary in a continuous sampling model.

Figure 6.7(a) shows the present values of leveraged and standard knockin puts. Both options expire in  $T = 1$  year and are struck at  $K = 100$ . The knockin barrier  $X_2 = 90$  is sampled continuously while the leveraging barrier  $X_1 = 75$  is sampled at 256 equally spaced sampling times, with the last sample at option expiration. We set  $L_{\min} = 0$  and  $L_{\max} = 256$ . These are our two grid points in the  $L$  direction. The standard knockin option value has been multiplied by the total number of samples,  $n = 256$ , for comparison purposes. For  $S > X_1$ , the value of the leveraged knockin reflects the strong effect of the leverage factor  $L$ . The ratio of the present values of the leveraged and standard knockin put is shown in Figure 6.7(b). For  $S < X_1$ , the ratio approaches the number of samples, while for  $S > X_1$  it decreases rapidly.

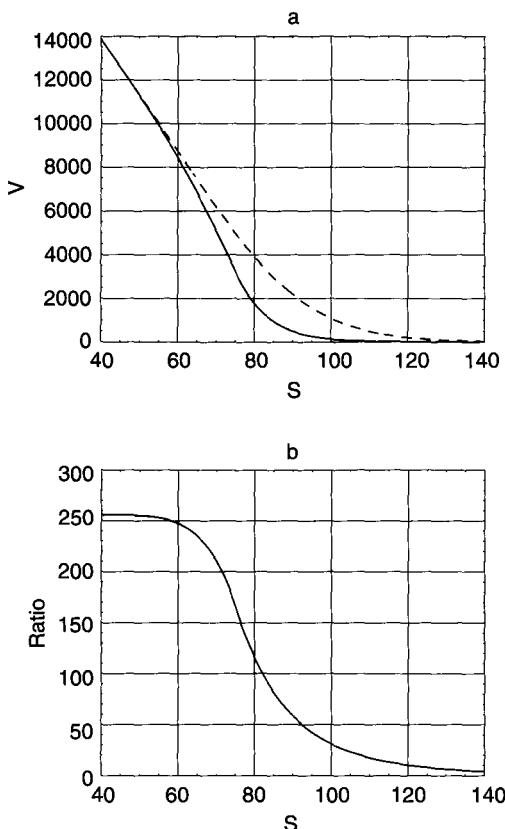
#### DISCRETELY SAMPLED ASIAN OPTIONS

In Chapter 4, the pricing of discretely sampled Asian options was discussed briefly and numerical results shown. In the earlier sections of this chapter, we have discussed in detail the implementation of numerical models of several other discretely sampled path-dependent options. In this section, we briefly describe the numerical implementation for Asian options and provide more numerical results.

We again solve the one-dimensional Black-Scholes equation with spot price variable  $S$  but on a two-dimensional  $(S,g)$  grid. The second dimension  $g$  is the running sum of spot prices. The option payoff is

$$V(S, g, T) = \max(g/n_{\text{samp}} - K, 0) \quad (6.30)$$

The average price that enters into the payoff function is simply  $g/n_{\text{samp}}$ , where  $n_{\text{samp}}$  is the number of samples at expiration and



**Figure 6.7** Results for leveraged knockin put. (a) Present option values of leveraged (solid curve) and standard (dotted curve) knockin puts. The standard knockin option value has been multiplied by the total number of samples  $n = 256$  for comparison. (b) Ratio of present values of leveraged and standard knockin puts. Parameters used:  $K = 100$ ,  $T = 1$ ,  $D_0 = 0$ ,  $\sigma = 0.2$ ,  $r = 0.06$ ,  $X_1 = 75$ ,  $X_2 = 90$ .

where  $K$  is the strike price. To price an option that is just starting, the minimum value of  $g$  on the grid is set to  $g_{\min} = 0$  and the present option values of interest are  $V(S, g = 0)$  (i.e., no sampling has taken place). An option that is already running will have a present average spot price,  $A_0$ , and a number of samples already taken,  $n_0$ . Then, the present option values of interest are  $V(S, g_0)$ , where  $g_0 = n_0 A_0$ . The remaining  $n_{\text{rem}} = n_{\text{samp}} - n_0$  samples will occur during the finite

difference calculation itself. At sampling times, we enforce the jump condition

$$V(S, g, t_i^-) = V(S, g + S, t_i) \quad (6.31)$$

In this discretely sampled Asian model, we will be interpolating a smooth, but nonlinear function, so we use cubic spline interpolation, which was shown in Chapter 4 to be superior to linear interpolation within the context of discrete dividends. In fact, the use of a high-order interpolator is often more important for sampling of discrete path dependencies because there will typically be many sample times, perhaps several hundred or more, over the life of the option. Since the option value has curvature in the  $g$  direction, a linear interpolator can result in a large spurious numerical diffusion that causes excessive smoothing of the solution in the  $g$  dimension. Since option value is typically upward convex, the smoothing will result in overpricing. In this case, the interpolation scheme used to implement this jump condition will again be asked to do extrapolation. At any point on the grid for which  $S > g_{\max} - g$ , Equation 6.31 will attempt to replace the option value at  $S, g$  as follows:

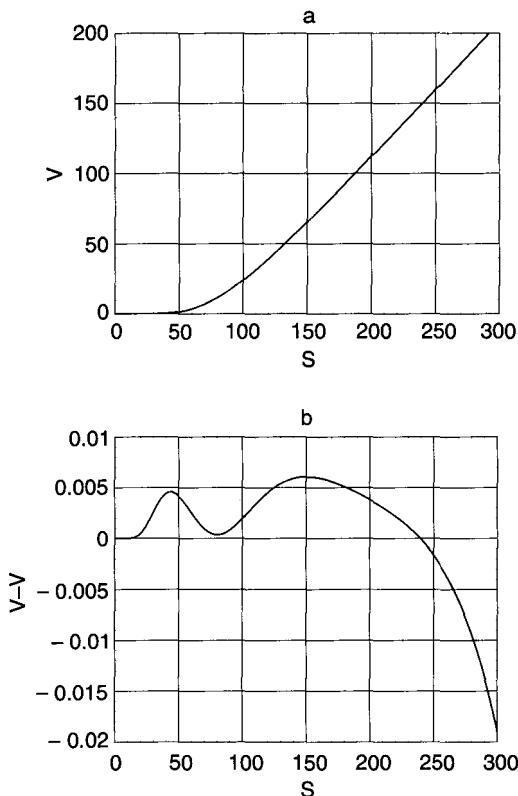
$$V(S, g, t_n^-) = V(S, g_{\text{new}}, t_n) \quad (6.32)$$

where

$$g_{\text{new}} = g + S > g_{\max} \quad (6.33)$$

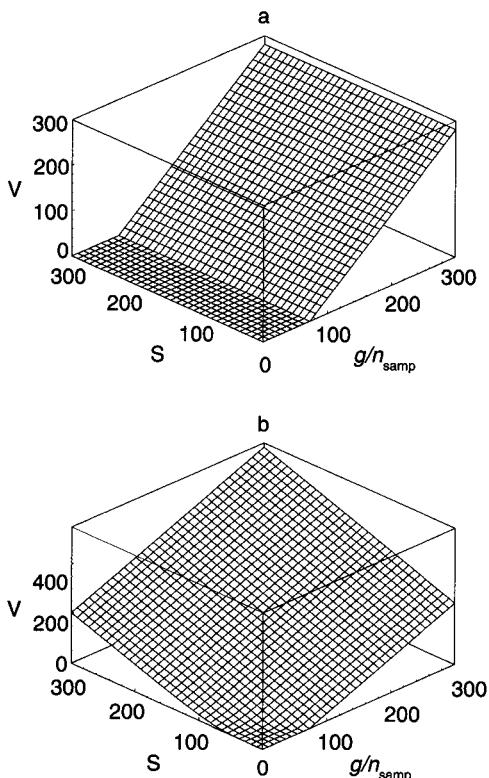
A simple analysis shows that linear extrapolation is again the appropriate solution. At expiration, the option value has no  $S$  dependence and is linear in  $g$  near the  $g_{\max}$  boundary. That is, it is planar near the boundary. The jump condition at each sample time shifts the option value by an amount that is linear in  $S$ . Therefore, these shifts produce a simple rotation of the surface  $V(S, g)$  so that it varies both with  $S$  and  $g$  (as shown in Figure 6.9, later in this chapter). But a rotation of a plane just produces another plane, and so the jump conditions will preserve the linearity of the option value surface in the  $g$  direction. This simple analysis neglects the effects of volatility and drifts in the  $S$  direction, but as long as the  $g_{\max}$  boundary is sufficiently large, linear extrapolation is an excellent approximation.

In this Asian option model, as well as in most discretely sampled path-dependent option models, there will be many grid points in the path dimension (e.g.,  $L$  or  $g$ ). One can extrapolate from the last two grid points, or do a least squares fit over some neighborhood near the boundary. In practice, we have found the former to be sufficient. Finally, if one makes a coordinate transformation of the path dimension, for example, a simple exponential transformation, or one of the more complex ones discussed in Chapter 5, then the extrapolation used in the jump condition must be suitably transformed as well to compensate.



**Figure 6.8** Results for a European arithmetically averaged fixed strike call. (a) Present value at  $g = 0$ . (b) Comparison of the finite difference option value with that from a geometrical conditioning approximation. Parameters used:  $K = 90$ ,  $T = 72/52$ ,  $g_{\max} = 20000$ ,  $S_{\max} = 300$ ,  $D_0 = 0$ ,  $\sigma = 0.5$ ,  $r = 0.09$ .

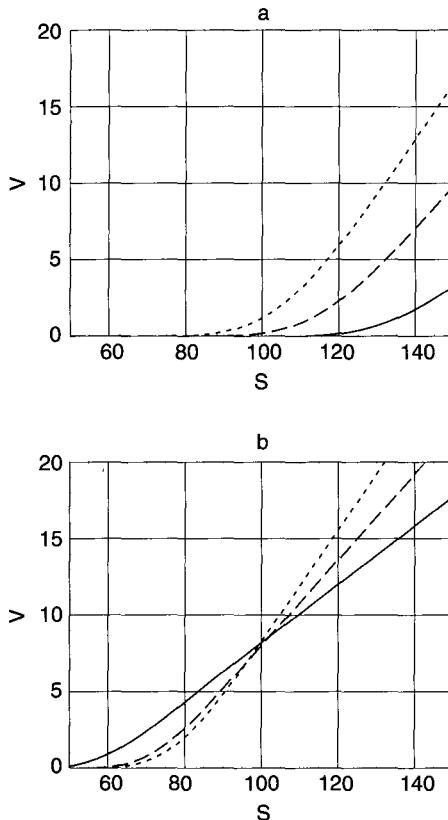
Figure 6.8(a) shows the value of a European arithmetically averaged fixed strike call. The option has 72 weeks to run. There will be 53 samples at weekly intervals, beginning with week 20 and ending at expiry. Other parameters are given in the figure caption. Figure 6.8(b) shows the difference between the finite difference results and those resulting from a geometrical conditioning approximation (Curran, 1995). Figures 6.9(a) and (b) show the expiration and present values respectively of the option over the entire two-dimensional  $S, g/n_{\text{samp}}$  region. The value at 72 weeks before expiry shown in Figure 6.8(a) corresponds to  $V(S, g = 0)$  in



**Figure 6.9** Results for a European arithmetically averaged fixed strike call. (a) Expiration and (b) present values as a function of  $S$  and  $g/n_{\text{samp}}$ . Plot of present value has been truncated to preserve scaling. Parameters used:  $K = 90$ ,  $T = 72/52$ ,  $g_{\text{max}} = 20000$ ,  $S_{\text{max}} = 300$ ,  $D_0 = 0$ ,  $\sigma = 0.5$ ,  $r = 0.09$ .

Figure 6.9(b). The computed present value can be larger than the expiration value because the jump conditions bring in data from beyond the artificial computational boundaries by linear extrapolation, as discussed.

Finally, Figure 6.10 shows the present value of a European, arithmetically averaged fixed strike call, with sampling already running. The option is a 5-year, weekly sampled option, with either



**Figure 6.10** Present value of a European, arithmetically averaged fixed strike call, with sampling already running. The option is a 5-year, weekly sampled option, with 2, 1.5, or 1 years left to run (dotted, dashed, and solid curves, respectively). The sampling began at the start of the option, which is struck at  $K = 100$ . (a)  $A_0 = 90$ . (b)  $A_0 = 110$ . Other parameters used:  $T = (2, 1.5, 1)$ ,  $D_0 = 0$ ,  $\sigma = 0.15$ ,  $r = 0.09$ ,  $S_{\max} = 300$ .

## 212 Numerical Examples

2, 1.5, or 1 years to run. The sampling began at the start of the option, which is struck at  $K = 100$ . When the present average is less than the strike, for example  $A_0 = 90$  as in Figure 6.10(a), then option value increases with the time remaining to expiration for all spot prices. However, when the present average is greater than the strike, as in Figure 6.10(b) where  $A_0 = 110$ , then option value increases with time remaining to expiration for spot prices above the strike, but decreases with time remaining for spot prices below the strike. As expected, the option  $\Delta$  becomes smaller as the time remaining to expiration decreases whatever the present average. In the limit of no samples remaining,  $\Delta$  must become zero.

The Asian jump conditions previously described are easily modified to price other varieties of Asian options. For example, to price a capped Asian option, in which the spot prices contributing to the running average are limited to some maximum value,  $C$ , the jump conditions become

$$V(S, g, t_i^-) = V(S, g + \min(S, C), t_i) \quad (6.34)$$

Whereas to price a weighted average Asian option, we would have

$$V(S, g, t_i^-) = V(S, g + w_i S, t_i) \quad (6.35)$$

## STOCHASTIC VOLATILITY

In this section, we price a European call with stochastic volatility. Finding a self-consistent approach for relaxing the Black-Scholes assumption of constant spot price variance is the subject of much research, for example Derman, Kani, and Chriss (1996). A local volatility,  $a(S, t)$ , can be obtained from market data through solution of an inverse problem using PDEs (Bodurtha & Jermakyan, 1999; Andersen & Brotherton-Ratcliffe, 1997), and using implied lattices (Derman et al., 1996). Volatility may be also modeled as a stochastic process, leading to two-factor models for equity options. In this example, we price European options with a stochastic volatility model proposed by Heston (1993). While one of several stochastic volatility

models, this example provides an application of the PDE approach to an interesting multidimensional problem, for which an analytic solution is available for comparison.

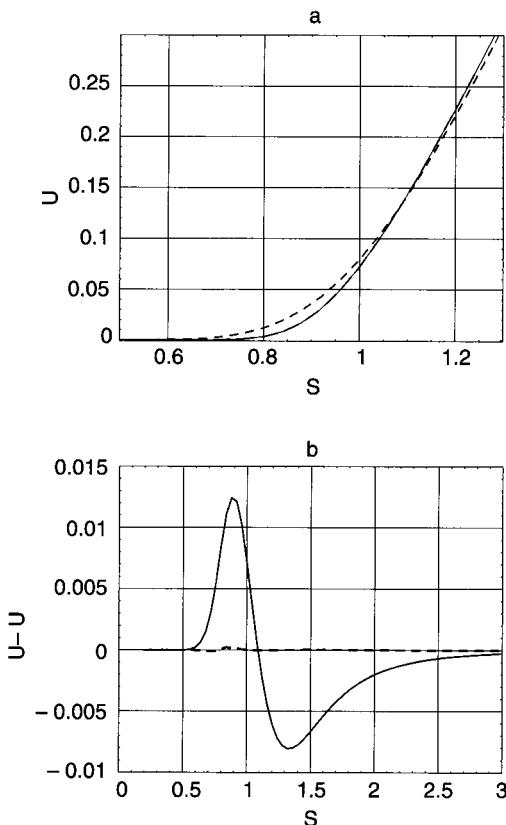
The equation to be solved for the option value  $U$  is

$$\begin{aligned} \frac{\partial U}{\partial t} + \frac{1}{2} v S^2 \frac{\partial^2 U}{\partial S^2} + \rho \sigma v S \frac{\partial^2 U}{\partial S \partial v} + \frac{1}{2} \sigma^2 v \frac{\partial^2 U}{\partial v^2} \\ + r S \frac{\partial U}{\partial S} + [\kappa(\theta - v) - \lambda(S, v, t)] \frac{\partial U}{\partial v} - r U = 0 \end{aligned} \quad (6.36)$$

where the variance of spot price is  $v$ , and  $\sigma$  is the volatility of the variance. We have used  $U$  here to denote option value to avoid confusion with the variance  $v$ .  $\theta$  is the mean variance,  $\kappa$  is the variance mean reversion rate, and  $\rho$  is the correlation of volatility to spot price. For simplicity, we take  $\theta$ ,  $\kappa$ , and  $\rho$  to be constants although they could easily be made functions of time. Heston writes the market price of volatility risk as  $\lambda(S, v, t) = \lambda v$ , where  $\lambda$  is a constant.

We use a Crank-Nicolson time discretization and BC1 boundary conditions on all boundaries, including the transverse diffusion and correlation terms. The discrete system of equations is solved using a biconjugate gradient-stabilized solver. In an American version of the model, an SOR solver can be applied, using a biconjugate gradient solution as an initial guess as described in Chapter 4. The expiration condition is the cell-averaged payoff.

Figure 6.11(a) shows the present value of the European call calculated by the finite difference code (solid curve), and for comparison, the Black-Scholes result for a constant variance equal to  $v = 0.04$  (dashed curve). The correlation between spot price and variance is  $\rho = -0.7$ , the mean variance is  $\theta = 0.04$ , the variance mean reversion rate is  $\kappa = 6$ , and the volatility of variance is  $\sigma = 1$ . The option is struck at  $K = 1$ . Because of the strong negative correlation of variance with spot price, the stochastic volatility model option value is far less than the Black-Scholes value at low spot prices. The difference between the two models, the solid curve in Figure 6.11(b), is about 10% at the strike price, and proportionately larger below the strike. For large spot prices, the stochastic volatility option value is as much as 2% to 3% larger than the Black-Scholes value. The dotted curve, just barely visible in Figure 6.11(b) is the difference between the finite difference and the analytic solution for the stochastic volatility model.



**Figure 6.11** (a) Present value of a European call calculated by the finite difference code (solid curve), and for comparison, the Black-Scholes result for a constant variance equal to 0.04 (dashed curve). (b) Difference between finite difference and analytic results for present option value (dotted curve) and the difference between finite difference and Black-Scholes constant volatility present option value (solid curve). Parameters used:  $D_0 = 0$ ,  $K = 1$ ,  $\kappa = 6$ ,  $\lambda = 0$ ,  $r = 0.0$ ,  $\rho = -0.7$ ,  $\sigma = 1$ ,  $\theta = 0.04$ ,  $T = 1$ .

#### CONVERTIBLE BOND

In this example, we price a convertible bond that has both put and call features, discrete coupon payments, and continuous conversion opportunity. For early but particularly clear examples of finite difference solutions for convertible bonds, the reader may consult

Brennan and Schwartz (1977, 1980). Again, we use a Cox-Ingersoll-Ross stochastic interest rate model. The pricing equation is

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + \frac{1}{2} \sigma^2 r \frac{\partial^2 V}{\partial r^2} + \rho \sigma \sigma_r S \sqrt{r} \frac{\partial^2 V}{\partial S \partial r} + (r - D_0) S \frac{\partial V}{\partial S} + (a - br) \frac{\partial V}{\partial r} - rV = 0 \quad (6.37)$$

The maturity condition for the bond is

$$V(S, r, T) = \min [\max [N_s S, PP(T) + AccI(T), Par], CP(T) + AccI(T)] \quad (6.38)$$

where  $CP(t)$  and  $PP(t)$  are the call and put prices, respectively, which may be functions of time.  $N_s$  is the conversion ratio.  $AccI(t)$  is the accrued interest on the pending coupon payment.

$$AccI(t) = K_i \frac{t - t_{i-1}}{t_i - t_{i-1}} \quad (6.39)$$

where  $t$  is the current time,  $t_i$  and  $t_{i-1}$  are times of the next pending and previous coupon payment dates, respectively, and where  $t_{i-1} \leq t < t_i$ .  $K_i$  is the amount of the  $i^{\text{th}}$  coupon payment. Thus,  $AccI(t)$  has a "sawtooth" shape, growing from zero at coupon payment times to  $K_i$  at the instant before the next payment.

The continuous conversion, puttable, and callable features are enforced with time-dependent constraints

$$\begin{aligned} V(S, r, t) &\geq N_s S \\ V(S, r, t) &\geq PP(t) + AccI(t) \\ V(S, r, t) &\leq CP(t) + AccI(t) \end{aligned} \quad (6.40)$$

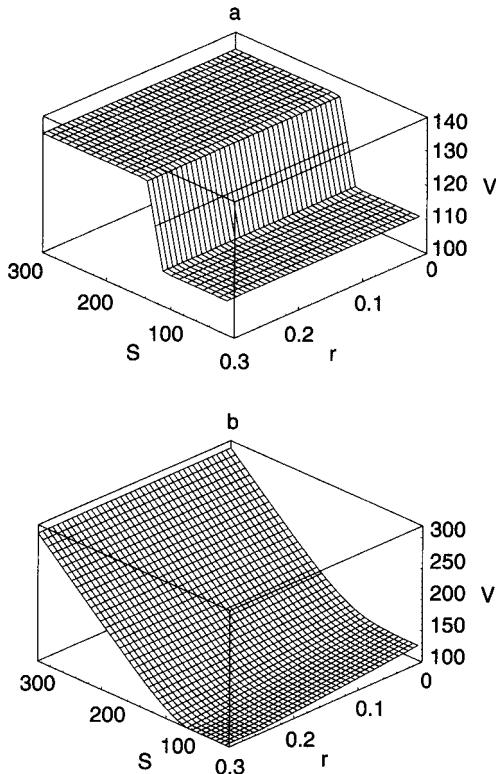
respectively, using an SOR solver following the main biconjugate gradient-stabilized main solver.

The discrete coupon payment is implemented with the jump conditions

$$V(S, r, t_i^-) = V(S, r, t_i) + K_i \quad (6.41)$$

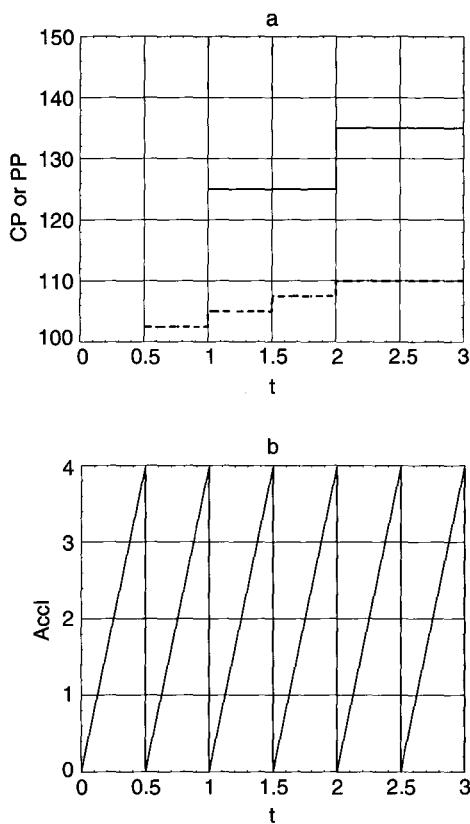
## 216 Numerical Examples

During the solution of the pricing equation, the time step is adjusted so that the coupon payment times  $t_i$  are hit exactly, along with any other times,  $t_{ci}$  and  $t_{pi}$ , at which  $CP(t)$  or  $PP(t)$  have discontinuities in either value or slope. Failure to hit these times exactly during the course of the computation means that the put or call prices are incorrect during some portion of the time step that includes them and hence reduces the order of accuracy of the calculation to  $O(\Delta t)$ . If the stock were to pay a discrete dividend, then the dividend payment times would have to be added to this list also.



**Figure 6.12** (a) Expiration value of a convertible bond with put and call features, discrete coupons, and continuous conversion opportunity under a eIR interest rate model. (b) Present value of the bond as a function of stock price.

Figure 6.12(a) displays the maturity value of a puttable and callable convertible bond as a function of spot stock price  $S$  and spot interest rate  $r$ . The bond matures in 3 years and pays a semi-annual coupon  $K_i = K = \$4$ . The par value is \$100. It is callable after the first year and puttable after the sixth month with call  $CP(t)$  and put  $PP(t)$  prices as shown in Figure 6.13(a). Figure 6.13(b) shows the accrued interest as a function of time. The volatility is  $\sigma = 0.35$  and the correlation of stock price and interest rates is  $\rho = -0.3$ . Figure 6.12(b) displays the bond's present value. Since the bond is neither callable nor



**Figure 6.13** (a) Call  $CP(t)$  and put  $PP(t)$  prices as a function of time. (b) Accrued interest on the pending coupon payment as a function of time.

putt able at the present, the continuous conversion constraint  $V \sim NsS$  forces bond value much higher than the maturity value.

Figure 6.14 displays the present value of four bonds as a function of stock price for a spot interest rate of  $r = 0.05$ . Shown are bonds that are: both puttable and callable, only putt able, only callable, and neither puttable nor callable.

#### SIMPLE FIXED INCOME INSTRUMENTS: FORWARD SWAP

In this section, we price a vanilla forward swap using PDE methods. We again employ a simple Cox-Ingersoll-Ross interest rate model, although the techniques presented can be easily generalized to more complex single factor and multifactor interest rate models. The extension of the techniques to more complex instruments such as swaptions, flexi-caps, and the like is also possible, but we shall not address them here.

In a vanilla forward swap, parties agree to periodically exchange cash flows corresponding to interest payments on a certain notional amount. The swap may start at some future time  $t_s$ . There follows  $n$  payment periods each of length  $\Delta_i$ . One party pays a fixed

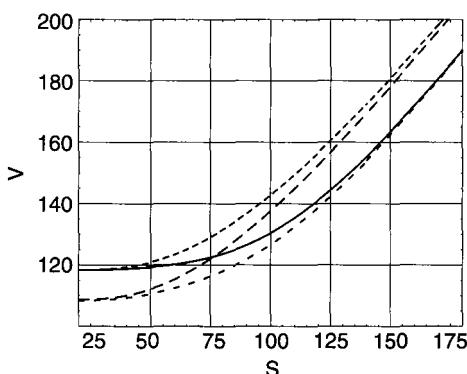


Figure 6.14 Present value of four bonds as a function of stock price for a spot interest rate of  $r = 0.05$ . Shown are bonds that are: both puttable and callable (solid curve), only putt able (dotted curve), only callable (short dashed curve), and neither puttable nor callable (long dashed curve).

rate of interest,  $r_f$ , while the other pays a floating rate, say six-month LIBOR. In a vanilla forward swap, the floating rate is set at the beginning of each payment period and the payments occur at the end of each period. The fixed interest rate is set so that the swap has no present value.

To price a forward swap, we solve two pricing equations simultaneously, one for the swap value, and one for a zero coupon bond:

$$\begin{aligned}\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 r \frac{\partial^2 V}{\partial r^2} + (a - br) \frac{\partial V}{\partial r} - rV &= 0 \\ \frac{\partial B}{\partial t} + \frac{1}{2} \sigma^2 r \frac{\partial^2 B}{\partial r^2} + (a - br) \frac{\partial B}{\partial r} - rB &= 0\end{aligned}\tag{6.42}$$

For simplicity, we let all the periods be of equal length  $\Delta$ . Then  $T = t_s + n\Delta$  is the expiration of the forward swap. We shall calculate the value of the swap to the party that pays the fixed rate (i.e., a “payer swap”). At  $t = T$ , the pricing variables are given the values

$$\begin{aligned}V(r, T) &= 0 \\ B(r, T) &= 1\end{aligned}\tag{6.43}$$

Then at the beginning of each payment period  $t_i^- = t_s + i\Delta$ , where  $0 \leq i \leq n - 1$ , the value of the swap undergoes a jump corresponding to the exchange of payments

$$V(r, t_i^-) = V(r, t_i) + \left[ \left( \frac{1}{B(r, t_i)} - 1 \right) - \Delta r_f \right] B(r, t_i)\tag{6.44}$$

where  $t_i^-$  is an instant of time before the payment exchange. After this jump in swap value, the value of the bond is reset:

$$B(r, t_i^-) = 1\tag{6.45}$$

In Equation 6.44, the term  $\left( \frac{1}{B(r, t_i)} - 1 \right)$  represents the floating leg payment at the beginning of a payment period.  $\Delta r_f$  is obviously the

fixed leg payment. Since these cash flows will actually occur at the end of the payment period, they are multiplied by  $B(r, t_i)$  to discount them to the beginning of the payment period. Once the difference of these cash flows is added to the swap value, the pricing equation itself takes care of the addition discounting to the present time. The bond value is reset to 1 after accounting for the cash flows so that the discounting factor for the next payment period can be calculated (“next payment period” in terms of the calculation, “previous payment period,” in calendar time). For an interest rate model with constant parameters, such as we use here, the bond calculation need only be done over a single period and can then be reused. However, the calculation that has been outlined is general enough to be used with more complex interest rate models that have time-dependent coefficients.

The fixed interest rate  $r_f$  that makes the forward swap valueless at the present (the forward swap rate) can be easily calculated using a numerical root finding algorithm such as Newton’s method (Press, Flannery, Teukolsky, & Vetterling, 1986). This technique, which can be applied to much more complex instruments, will converge quite rapidly in this example because  $V$  is nearly linear in  $r_f$ . However, for the simple vanilla forward swap an analytic formula for  $r_f$  may be derived (Rebonato, 1997):

$$r_f = \frac{1}{\Delta} \frac{B(0, t_s) - B(0, T)}{\sum_{i=1}^n B(0, t_i)} \quad (6.46)$$

where  $B(t, T)$  is the value at time  $t$  of a zero coupon bond that pays one dollar at  $T$ . To evaluate this expression, we can solve the bond pricing equation  $n + 1$  times to calculate the required sum, or more economically, we can solve a set of three PDEs for the sum of bond prices in Equation 6.46:

$$\begin{aligned} \frac{\partial B}{\partial t} + \frac{1}{2} \sigma^2 r \frac{\partial^2 B}{\partial r^2} + (a - br) \frac{\partial B}{\partial r} - rB &= 0 \\ \frac{\partial \Sigma}{\partial t} + \frac{1}{2} \sigma^2 r \frac{\partial^2 \Sigma}{\partial r^2} + (a - br) \frac{\partial \Sigma}{\partial r} - r \Sigma &= 0 \\ \frac{\partial B_s}{\partial t} + \frac{1}{2} \sigma^2 r \frac{\partial^2 B_s}{\partial r^2} + (a - br) \frac{\partial B_s}{\partial r} - rB_s &= 0 \end{aligned} \quad (6.47)$$

along with initial and jump conditions

$$\begin{aligned} B(r, T) &= 1 \\ \Sigma(r, T) &= 1 \\ B_s(r, t_s) &= 1 \\ B(r, t_i^-) &= 1 \\ \Sigma(r, t_i^-) &= \Sigma(r, t_i) + 1 \end{aligned} \tag{6.48}$$

Then

$$r_f(r) = \frac{1}{\Delta} \frac{B_s(r, 0) - B(r, 0)}{\Sigma(r, 0)} \tag{6.49}$$

gives the forward swap rate  $r_f(r)$  for any spot interest rate  $r$ .

As has been described for bond pricing examples in Chapter 4, we employ the pricing equation itself as boundary conditions at  $r = r_{\min}$  and  $r = r_{\max}$ , discretized with one-sided finite difference operators. We use a Crank-Nicolson time discretization scheme. However, the time step is adjusted as required so the calculation lands exactly on the payment times  $t_i$  to maintain second order accuracy in time.

In Table 6.5, we display results for the forward swap at a spot interest rate of  $r = 0.05$  as a function of the number of time steps  $N$  and grid points  $I$ . We have set  $I = N$  for convenience. The second column is the error in the calculated  $r_f(0.05)$  relative to the analytic result  $r_f(0.05) = 0.0483010766$  given by Equation 6.46. Furthermore, the bond values in the analytic formula are themselves calculated by the analytic formula available for CIR bonds (Cox, Ingersoll, & Ross, 1985). The forward swap begins in one year and runs for three years with six payment periods. The error in the forward swap rate is less than a tenth of a basis point even for  $I = N = 10$  and converges roughly as the inverse cube of  $I$ , until reaching a lower bound. This lower bound on error arises from the artificial boundary at  $r_{\max}$  and our imperfect boundary condition there. It can be eliminated by increasing  $r_{\max}$ , here 0.5.

The third column of Table 6.4 is the value of the swap in dollars at the chosen spot rate for a notional amount of \$100 million. The

**Table 6.5 Forward Swap**

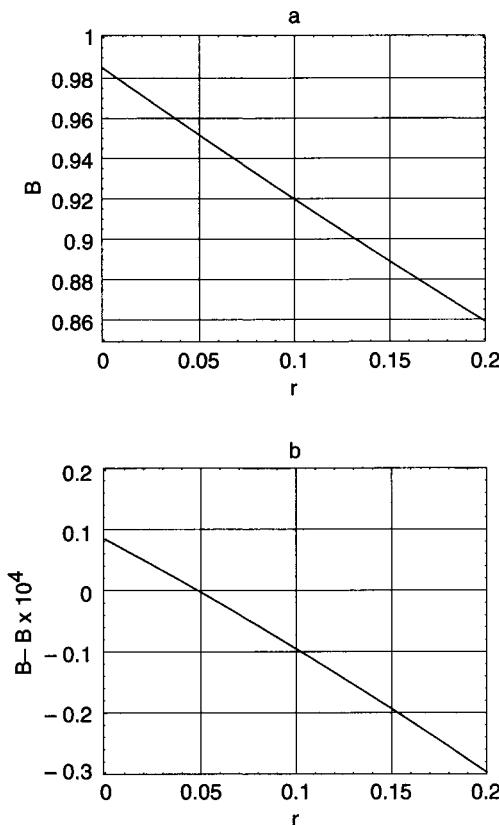
$I = N$	error $r_f(0.05)$	$V(0.05)$
10	-0.0000130440	0.64
20	-0.0000016701	-0.95
30	-0.0000004272	-0.19
40	-0.0000001453	0.29
50	-0.0000000692	-0.71
60	-0.0000000402	-0.10
70	-0.0000000245	0.52
80	-0.0000000166	-0.48
90	-0.0000000149	0.42
100	-0.0000000144	0.53

Error in forward swap rate relative to the analytic result and present value of the forward swap for a notional amount of \$100 million, at spot interest rate 0.05, as a function of  $I = N$ . The swap starts in one year and runs for three years with six equal payment periods. Other parameters used:  $a = 0.8$ ,  $b = 0.04$ ,  $c = 0.3$ ,  $r_{\min} = 0$ ,  $r_{\max} = 0.5$ .

theoretical value of the forward swap at the chosen spot rate should be zero. Because  $r_f$  and  $V$  are calculated self-consistently, by identical finite difference algorithms, the departure from zero value is negligible even for  $I = N = 10$ . It shows no systematic decrease as grid resolution is improved.

Figure 6.15(a) shows present value of the zero coupon bond B used in the forward swap calculation, as a function of spot interest rate. This bond is reset to its maturity value at the beginning of every payment period. The error in present value of the zero coupon bond (x 104)relative to the analytic value is shown in Figure 6.15(b) with  $I = 50$  grid points. Finally, Figure 6.16 displays the present value of the forward swap as a function of spot interest rate. The forward swap rate has been chosen to make the swap valueless at a present spot interest rate of 0.05.

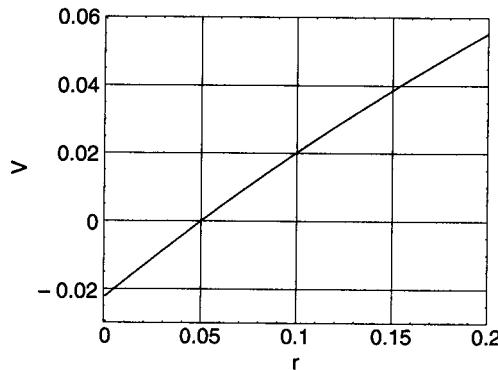
While the forward swap valuation problem just presented can easily be done analytically for this simple interest rate model, the methods presented are applicable to more complex instruments and interest rate models.



**Figure 6.15** (a) Present value of the zero coupon bond  $B$  used in the forward swap calculation, as a function of spot interest rate. The swap starts in one year and runs for three years with six equal payment periods. The bond is reset to its maturity value at the beginning of every payment period. (b) Error in present value of the zero coupon bond ( $\times 10^4$ ) relative to the analytic value. Other parameters used:  $a = 0.8$ ,  $b = 0.04$ ,  $c = 0.3$ ,  $r_{\min} = 0$ ,  $r_{\max} = 0.5$ .

#### CREDIT DERIVATIVES

In this section, we discuss implementation of a numerical model for the full protection credit put discussed in Chapter 3. In this example calculation, credit events (defaults) are modeled as a pure jump process whose intensity  $h$  is itself a lognormal stochastic process. The



**Figure 6.16** Results for a vanilla forward swap. Present value of the forward swap as a function of spot interest rate. The forward swap rate has been chosen to make the swap valueless at a present spot interest rate of 0.05. The swap starts in one year and runs for three years with six equal payment periods. Other parameters used:  $a = 0.8$ ,  $b = 0.04$ ,  $c = 0.3$ ,  $r_{\min} = 0$ ,  $r_{\max} = 0.5$ .

specific interest rate model chosen is Cox-Ingersoll-Ross. Recapitulating the derivation of Chapter 3: The derivative that pays the holder the following amount if default occurs at time  $t$ , where  $0 < t < T_P$

$$KB(t, T_B) - R(t)B_d(t, T_D) \quad (6.50)$$

where  $R(t)$  is the recovery rate,  $B(t, T_B)$  is a risk-free bond maturing at time  $T_B$ ,  $B_d(t, T_D)$  is a defaultable bond maturing at time  $T_D$ , and  $K$  is a constant. The contract terminates in case of default. If no payments triggered by default have occurred at maturity  $T_P$ , then the derivative pays

$$\max[0, KB(T_P, T_B) - B_d(T_P, T_D)] \quad (6.51)$$

This is a full-protection credit put because it protects both against defaults and drop of value as a result of increased yields. The pricing equation can be written

$$\frac{\partial V}{\partial t} + \mu_r \frac{\partial V}{\partial r} + \mu_h \frac{\partial V}{\partial h} + \frac{\sigma_r^2}{2} \frac{\partial^2 V}{\partial r^2} + \sigma_r \sigma_h \rho_{rh} \frac{\partial^2 V}{\partial r \partial h} + \frac{\sigma_h^2}{2} \frac{\partial^2 V}{\partial h^2} = (r + h)V - h[KB(t, T_B) - R(t)B(t, T_D)] \quad (6.52)$$

with payoff

$$V(T) = \max[0, KB(T, T_B) - B_d(T, T_D)] \quad (6.53)$$

where  $\mu_r$  and  $\mu_h$  are the drifts of the interest rate and credit event intensity, and  $\sigma_r$  and  $\sigma_h$  are the stochastic parts of the interest rate and credit event intensity processes, respectively. We need to know  $B(t, T_B)$  and  $B_d(t, T_D)$  at every point in the solution space. Thus, we need to simultaneously solve pricing equations for the values of the riskless and defaultable bonds. For the riskless bond

$$\frac{\partial B}{\partial t} + \mu_r \frac{\partial B}{\partial r} + \frac{\sigma_r^2}{2} \frac{\partial^2 B}{\partial r^2} = rB \quad (6.54)$$

while for the defaultable bond,

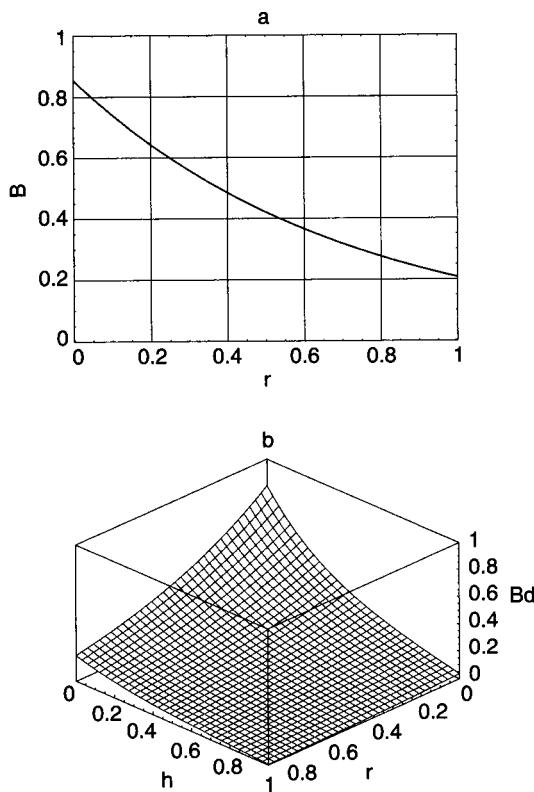
$$\begin{aligned} \frac{\partial B_d}{\partial t} + \mu_r \frac{\partial B_d}{\partial r} + \mu_h \frac{\partial B_d}{\partial h} + \frac{\sigma_r^2}{2} \frac{\partial^2 B_d}{\partial r^2} + \\ \sigma_r \sigma_h \rho_{rh} \frac{\partial^2 B_d}{\partial r \partial h} + \frac{\sigma_h^2}{2} \frac{\partial^2 B_d}{\partial h^2} = [r + (1 - R)h]B_d \end{aligned} \quad (6.55)$$

We now specialize to a Cox-Ingersoll-Ross interest rate model:  $\mu_r \rightarrow \alpha_r - \beta_r r$ ,  $\sigma_r \rightarrow \sigma_r \sqrt{r}$  and a lognormal credit intensity process,  $\mu_h \rightarrow \mu_h h$ ,  $\sigma_h \rightarrow \sigma_h h$ . We assume that the recovery factor is a constant.

In the interior of the  $r - h$  region, we solve the three equations for  $V$ ; the value of the full protection credit put,  $B_d$ ; the value of the defaultable bond; and  $B$  the value of the riskless bond. We use the pricing equations themselves as boundary conditions, using the BC2 scheme, that is, employing first-order accurate one-sided difference operators for the second derivative terms with derivatives normal to

the boundary under consideration. For first derivative terms normal to the boundary, we employ a second order accurate one-sided difference discretization. Derivatives transverse to the boundary are differenced with the usual second order accurate central difference discretization. Correlation terms are discarded at the corner points.

We set  $T = \max(T_B, T_D) \geq T_P$  and initialize the longer dated bond to its maturity condition, either  $B = 1$  or  $B_d = 1$ . The solution for the longer dated bond is evolved in time until we reach  $t = \min(T_B, T_D)$ . At this time, the shorter dated bond is initialized to its maturity

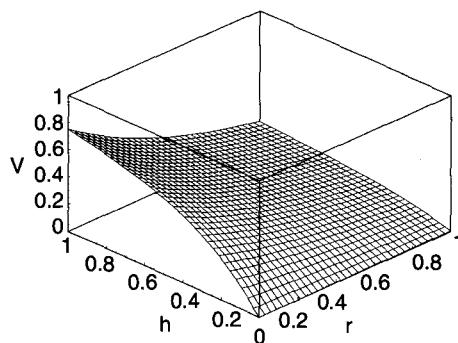


**Figure 6.17** Present bond values for full protection credit put. The defaultable bond has a maturity of  $T_D = 5$  years, while the riskless bond matures in  $T_B = 4$  years. (a) Present value of the riskless bond  $B$ . (b) Present value of the defaultable bond  $B_d$ . Parameters used:  $\alpha_r = 0.04$ ,  $\beta_r = 0.6$ ,  $h_{\max} = 1.0$ ,  $K = 0.93$ ,  $\mu_h = 0.1$ ,  $R = 0.6$ ,  $\rho = 0$ ,  $r_{\max} = 1.0$ ,  $\sigma_h = 0.2$ ,  $\sigma_r = 0.3$ .

condition, again either  $B = 1$  or  $B_d = 1$  and both bond solutions are evolved until we reach  $t = T_p$ . At this time the value of the put is initialized to its payoff given by Equation 6.53. As in all situations involving discrete sampling, or discontinuous behavior in time, it is very important that during the calculation, the time step be adjusted as necessary to hit the relevant times, here  $T_B$ ,  $T_D$ , and  $T_p$ , exactly. We use a preconditioned biconjugate gradient-stabilized solver for all three pricing equations, a one-dimensional version for the  $B$  equation, and two-dimensional versions for the  $B_d$  and  $V$  equations. Fifty grid points are used in both the  $r$  and  $h$  dimensions, and there are 50 time steps.

Figure 6.17(a) displays the present value of the riskless bond  $B$  as a function of spot interest rate  $r$ . Agreement with the analytic CIR bond formula is a few parts in  $10^5$ . Figure 6.17(b) shows the present value of the defaultable bond  $B_d$  as a function of spot interest rate  $r$  and spot default intensity  $h$ . As expected, the value  $B_d$  decreases rapidly with both  $h$  and  $r$ . The defaultable bond has a maturity of  $T_D = 5$  years, while the risk-free bond matures in  $T_B = 4$  years.

The put expires at  $T_p = 1$  year. The strike is  $K = 0.93$  reflecting the difference in riskless and defaultable bond maturities. In Figure 6.18, we display the present value of the full protection credit put  $V$ . As expected, the put value is nearly zero as  $h \rightarrow 0$  and increases with  $h$  as the probability of default before expiration



**Figure 6.18** Present value of the full protection credit put  $V$ . Parameters used:  $\alpha_r = 0.04$ ,  $\beta_r = 0.6$ ,  $h_{\max} = 1.0$ ,  $K = 0.93$ ,  $\mu_h = 0.1$ ,  $R = 0.6$ ,  $\rho = 0$ ,  $r_{\max} = 1.0$ ,  $\sigma_h = 0.2$ ,  $\sigma_r = 0.3$ .

grows. The put value decreases with  $r$ , reflecting the decrease in the value of the underlying bonds. The present value of the put and defaultable bond at spot interest rate  $r = 0.05$  and spot credit event intensity  $h = 0.02$ , are  $V(0.05, 0.02) = 0.0282$  and  $B_d(0.05, 0.02) = 0.7112$ .

#### REFERENCES

- Ahn, D.-H., Figlewski, S., & Gao, B. (1999). Pricing Discrete Barrier Options with an Adaptive Mesh. *Journal of Derivatives*, 6, 33-44.
- Andersen, L.B.G., & Brotherton-Ratcliffe, R (1997). The Equity Option Volatility Smile: An Implicit Finite-Difference Approach. *Journal of Computational Finance*, 1, 5-38.
- Bodurtha, J.N., & Jermakyan, M. (1999). Nonparametric Estimation of an Implied Volatility Surface. *Journal of Computational Finance*, 2, 29-60.
- Brennan, M.J., & Schwartz, B.S. (1977). Convertible Bonds: Valuation and Optimal Strategies for Call and Conversion. *Journal of Finance*, 32, 1699-1715.
- Brennan, M.J., & Schwartz, B.S. (1980). *Journal of Financial and Quantitative Analysis, Proceedings Issue*, 15, (4), 907-929.
- Chesney, M., Cornwall, J., Jeanblanc-Picque, M., Kentwell, G., & Yor, M. (1997, January). Parisian Pricing. *Risk*, 10, 77-79.
- Cheuk, T.H.F., & Vorst, T.C.F. (1995). Complex Barrier Options. *Journal of Derivatives*, 4, 8-22.
- Clewlow, L., & Strickland, C (1998). *Implementing Derivative Models*. Chichester, England: Wiley.
- Cox, J.C, Ingersoll, J.E., & Ross, S.A. (1985). A Theory of the Term Structure of Interest Rates. *Econom.*, 53, 363.
- Curran, M. (1995). Beyond Average Intelligence. In R Jarrow (Ed.), *Over the Rainbow: Developments in Exotic Options and Complex Swaps*. London: Risk Publications.
- Derman, B., Kani, I., & Chriss, N. (1996). Implied Trinomial Trees of the Volatility Smile. *Journal of Derivatives*, 3, 7-22.
- Dupire, B. (1997, October 21). *A Unified Theory of Volatility*. Presented at Derivcon 97, Ft. Lauderdale, FL.
- Haber, R.J., SchOnbucher, P.J., & Wilmott, P. (1999, Spring). Pricing Parisian Options. *Journal of Derivatives*, 1, 71-79.
- Heston, S. (1992). *Review of Financial Studies* 6, 327-343.

- Press, W., Flannery, B., Teukolsky, S., & Vetterling, W. (1986). *Numerical Recipes*. Cambridge, England: Cambridge University Press.
- Rebonato, R. (1997). *Interest-Rate Option Models*. Chichester, England: Wiley.
- Rogers, L.c.G., & Zane, O. (1997). Valuing Moving Barrier Options. *Journal of Computational Finance*, 1, 5-11.

# Index

- Alternating Direction Implicit (ADI) solvers, 95, 177
- American derivatives/options, 2, 23,39-45,59,100-105
- conditions at exercise boundaries, 43-44
- defined, 2
- derivation of pricing equation for, 39-45
- as dynamic optimization problems, 42-43
- finite difference approach to, 100-105
- linear complementarity problem (LCP), 44-45,100-105
- relationship between European and, 40-41
- Arbitrage pricing principle, 1,3, 13-21
- and change of measure, 16-21, 48-50
- introduced by Black/Scholes/Merton, 1
- Asian options, 3, 49, 114, 132, 135, 136,137,138,139,206-212
- continuously sampled, 132, 135, 136
- discretely sampled, 136-137, 138, 206-212
- jump conditions, 212
- Banded systems, 94
- Barrier options, 183-196
- and coordinate transformations, 156, 163, 165, 171
- knockout call: nonuniform grid (Table 6.2),194
- knockout call: uniform grid (Table 6.1),193
- nonuniform grids and discrete sampling, 187-196
- time-dependent barriers, 183-187
- Basket options, 4,141,143,144, 146,178,179,180,181
- Bellman principle of dynamic programming, 42,43
- Bermudan options, 2
- Biconjugate gradient stabilized (BCGST) solver, 142, 144, 145, 146,213,227
- BCGST-ILU,145
- Black-Scholes model/equation, 2, 26
- assumptions, 121, 183
- eigenvalue analysis of, 86-90
- examples, 33, 57-58, 113, 119, 121, 122, 125, 137, 139, 154, 156, 157, 172, 173, 183, 197, 202,206,213,214
- hedge parameters, 59
- single-asset, 57-58
- Bonds:
- convertible (numerical example), 214-218
- defaultable,37-38
- interest rate model (*see* Cox-Ingersoll-Ross (CIR) interest rate model)
- Boundary conditions, 39, 43, 120-132

- Boundary conditions (*Continued*)  
 BC1 defined, 123  
 BC2 defined, 124  
 in multiple dimensions, 130-132  
 in one dimension, 121-130
- Brownian motion, 14, 17, 19-20, 23,  
 49,52,55, 147
- CG (conjugate gradient) solver, 144,  
 145, 146
- Change of measure, 16-21,48-50
- Chapmann- Kolmogorov equation,  
 6
- CIR. *See* Cox-Ingersoll-Ross (CIR)  
 interest rate model
- Complete markets, 13,40
- Conjugate gradient (CG) solver,  
 144, 145, 146
- Consistency, 5-6, 66
- Continuity condition, 48. *See also*  
 Jump(s)
- Convergence, 66, 110-114, 142
- Convertible bond (numerical  
 example), 214-218
- Convolution integrals, 33, 147, 148,  
 150
- Convolution theorem, Fourier, 153
- Coordinate transformations, 90,  
 113,156-182  
 multidimensional, linear, 177-181  
 multidimensional, time-  
 independent, 173-181  
 factored, 174-175  
 general, 175-177
- one-dimensional, time-  
 dependent, 172-173
- one-dimensional, time-  
 independent, 157-173  
 concentrating grid points,  
 167-171  
 placing grid points at selected  
 positions, 160-167
- Cox-Ingersoll-Ross (CIR) interest  
 rate model, 126, 128, 129, 130,  
 215,218,221,224,225,227
- Crank-Nicolson scheme, 80-82, 86,  
 87,88, 89, 93-94, 101, 110, 123,  
 126, 134, 135, 143, 149, 150,213,  
 221  
 and barrier options, 184, 189,  
 190, 191  
 defined, 80-82
- Credit derivatives, 30, 34-37,  
 38-39, 107,223-228  
 numerical example, 223-228
- Cubic spline interpolation, 115,  
 117,118,120,161,166,198
- Currency/foreign  
 exchange/markets, 16, 18-21,  
 50-55
- Digital call option, 112, 114, 115
- Dimensionality reduction, 3, 21,  
 48-55  
 currency translated options (four  
 types), 50-55  
 reformulating underlying  
 processes in different  
 measure, 49-50
- Discretization:  
 distortions induced by, 105-107  
 schemes, finite difference  
 (commonly used) (Table 3.1),  
 69  
 space discretizations, 63, 71-73,  
 107  
 time discretization, 65, 73-76,  
 93,107 (*see also* Crank-  
 Nicolson scheme)  
 upwind, 61, 120, 133, 135
- Dissipation/ dispersion, 105-106
- Dividends, 56,114
- Drift, 8, 13-14, 19  
 risk-neutral, 14, 16,27

- Drift-dominated pricing equations, 134-135
- Eigenvalue analysis of Black-Scholes equation, 86-90
- Euler scheme:  
 explicit, 77-79, 83, 84, 87, 88, 125, 126  
 implicit, 79-80, 87, 88, 125, 126, 150
- European derivatives/options, 2, 23, 24-29, 40-41, 61-67  
 defined, 2  
 examples, 112, 117, 150, 151, 152, 164, 183, 210, 211, 212  
 relationship between American and, 40-41
- European derivatives/options:  
 pricing equation for, 23, 24-29, 61-67
- Feynman-Kac approach, 27-29  
*(see also Feynman-Kac theorem)*
- hedging portfolio approach, 24-27
- European options: pricing  
 equation for (in presence of jumps), 30-39  
 bonds, defaultable, 37-38  
 credit derivatives, 30, 34-37  
 full protection credit put, 38-39  
 minimum variance pricing equation, 30-33
- Exercise boundaries, 41, 43-44
- Exercise strategy, 2, 40-42, 98  
 exercise-don't exercise, 41, 42  
 optimal, 2, 40, 41, 42, 98
- Explicit Euler. *See* Euler scheme, explicit
- Feynman-Kac approach, pricing  
 European options, 27-29
- Feynman-Kac theorem, 5, 16, 52, 53, 55
- FFT algorithm, 153
- Financial instruments (two fundamental types), 2  
 American (*see* American derivatives /options)  
 European (*see* European derivatives /options)
- Financial instruments, pricing:  
 coordinate transformations, 156-182  
 equations, derivation of, 23-60  
 finite difference methods, I, 61-109  
 numerical examples, 183-228  
 overview/introduction, 1-22  
 special issues, 110-155
- Financial instruments, pricing (derivation of equations), 23-60  
 American derivatives, 39-45  
 dimensionality reduction, 48-55  
 European options, 24-39  
 hedging parameters, 56-59  
 path dependency, 45-48
- Financial instruments, pricing (derivation of equations, European options in presence of jumps), 30-39  
 bonds, defaultable, 37-38  
 credit derivatives, 30, 34-37  
 full protection credit put, 38-39  
 minimum variance pricing equation, 30-33
- Financial instruments, pricing (numerical examples), 183-228  
 barrier options, 183-196  
 convertible bond, 214-218  
 credit derivatives, 223-228  
 discretely sampled Asian options, 206-212

- Financial instruments, pricing  
(numerical examples)  
*(Continued)*
- discretely sampled Parisian options, 196-201
  - leveraged knockin put, 202-206, 207
  - simple-fixed income instruments: forward swap, 218-223
  - stochastic volatility model, 212-214
- Financial instruments, pricing (special issues), 110-114
- boundary conditions, 120-132
  - in multiple dimensions, 130-132
  - in one dimension, 121-130
  - effect of payoff discontinuities on convergence, 110-114
  - jump conditions, 114-120
  - numerical solution of PIDEs: jump-diffusion and pure jump models, 147-154
  - path-dependent options, 132-141
  - continuous sampling, 132-136
  - discrete sampling, 136-141
  - performance of solvers for multidimensional problems, 141-146 (*see also* Solvers)
- Finite difference equations (FDEs), 71
- Finite difference methods, 1, 61-109
- American options, finite difference approach to (analysis chapter), 100-105
  - commonly used finite difference discretization schemes (Table 3.1), 69
  - constructing finite difference approximations, 67-70
  - distortions induced by discretization, 105-107
  - error, two fundamental sources of, 66
  - issues (consistency / convergence/stability), 66
  - motivation, 61-67
  - strategies for complex derivative structures, 107-108
  - time advancement, 93-100
  - vs. trees/simulation, 1
- Finite difference methods:
- algorithms, 77-86
  - Crank-Nicolson scheme, 80-82
  - explicit Euler scheme, 77-79
  - implicit Euler scheme, 79-80
  - multi-level methods, 84-86
  - predictor-corrector methods, 82-84
- Finite difference methods: solving sparse systems of linear equations, 94-100
- common preconditioners (Table 3.2), 97
  - common solvers (Table 3.3), 98
  - direct solvers, 94-96
  - Gauss-Seidel method, 97, 99-100
  - iterative solvers, 96-99
  - Jacobi iterative method, 97, 98, 99, 104
  - successive overrelaxation (SOR) method, 100 (*see also* Successive overrelaxation (SOR) iterative solver)
- Finite difference methods: stability analysis, 70-93
- Fourier approach, 90-93
  - multilevel scheme, 92-93
  - theta method, 91-92
  - matrix approach, 70-90, 124

- eigenvalue analysis of Black-Scholes equation, 86-90, 124
- space discretization, 71-73
- time discretization, 73-76
- Fokker-Planck equation, 5,6,7,8,9
- Foreign exchange/markets, 16, 18-21,50-55
- Forward swap (numerical example), 218-223
- Fourier approach, stability analysis, 70,90-93, 105, 106, 133, 153
- multilevel scheme, 92-93
- theta method, 91-92
- Gaussian elimination, 9,95, 98
- Gauss-Seidel method, 97, 99-100
- Girsanov's theorem, 17, 18
- Greeks,56-59,88,135
  - computing, 56-59
  - defined, 56
- Hedging parameters, equations for, 5, 56-59
- Hedging portfolio approach, pricing European options, 24-27
- Implicit Euler. *See* Euler scheme, implicit
- Interpolation methods, higher-order, 115-120
- Iterative solvers. *See* Solvers
- Ito's formula/lemma, 5, 9-13
  - applied, 18,20,25,27,35,46,47, 49
  - defined, 5, 9-13
  - for processes with jumps, 10-13
- Jacobi iterative method, 97, 98, 99, 104
- Joint probability density function (JPOF),4
- Jump(s),5, 10-13,30-39,48, 114-120,147-154,212
  - Asian jump conditions, 212
  - continuity condition referred to as,48
  - credit derivatives, 34-37
  - European options: pricing equation in presence, 30-39
  - implementing jump conditions, 114-120
  - Ito's formula, 10-13
  - jump diffusion model, 147, 151
  - numerical solution of PIOEs, 147-154
  - Poisson, 5, 10-11, 147
  - pricing equation in presence of, 30-39
  - pure jump models, 147-154
- Knockin put, leveraged, 202-206, 207
- Knockout call, 191-194
- Lagging, 149
- Lax Equivalence Theorem, 66-67
- Leveraged knockin put, 202-206, 207
- Leveraging path dependency, 202
- Linear complementarity problems (LCP), 3, 44-45, 100
- Lookback options, 136, 140
- Market, complete, 13
- Markovian processes, 5-8
- Matrix approach. *See* Finite difference methods: stability analysis
- Measure transformation, 16-21, 48-50

- Minimum variance pricing equation, 30-33
- Neumann, 86, 124, 125
- Optimal exercise strategy, 2, 40, 41, 42, 98
- Ordinary differential equations (ODEs), 71, 169
- Parisian options, discretely sampled, 196-201
- Partial differential equations (PDEs), 1,23
- Partial-inte gro-differentiation equations (PIDEs), 33, 147-154
- Path dependency, 45-48, 122, 132-141, 196, 202, 204
- continuous sampling, 132-136
- discrete sampling, 47-48, 136-141
- leveraging, 202, 204
- Parisian option, 196
- Payoff discontinuities, 110-114
- Poisson process/jumps, 5, 10-11, 147
- Predictor-corrector methods, 82-84
- Pricing financial instruments. *See* Financial instruments, pricing
- Quantization error, 111, 113, 161
- Quasi-minimum residual (QMR) solvers, 144, 145
- Radon-Nikodym derivative, 17
- Recovery value, 37-38
- Risk neutrality, 2, 13, 14, 16, 18, 20, 26-27, 30, 49
- SDE. *See* Stochastic differential equation (SDE)
- Simulation, I, 3
- Smooth pasting, 43, 44
- Solvers, 94-100, 141-146
- ADI (alternating direction implicit), 95, 177
- BCGST (biconjugate gradient stabilized), 142, 144, 145, 146, 213, 227
- CG (conjugate gradient) solver, 144, 145, 146
- common (Table 3.3), 98
- common preconditioners (Table 3.2), 97
- convergence rate, and coordinate transformations, 158
- direct, 94-96
- Gauss-Seidel method, 99-100
- iterative, 96-99, 124
- Jacobi method, 99
- performance of, for multidimensional problems, 141-146
- QMR (quasi-minimum residual), 144, 145
- SOR (successive overrelaxation), 97, 98, 99, 100, 124, 143, 144, 145, 146, 163, 213
- for sparse systems of linear equations, 94-100, 124
- SSOR (symmetric successive overrelaxation), 97, 98
- stationary / nonstationary methods, 97
- tridiagonal, 94-95, 98, 124, 185
- SOR. *See* Successive overrelaxation (SOR) iterative solver
- Space discretizations, 63, 71-73, 106
- Sparse matrices. *See* Finite difference methods: solving sparse systems of linear equations

- Spectral methods, 93
- SSOR (symmetric successive overrelaxation), 97, 98
- Stability, 66, 70-93, 158
  - analysis (*see* Finite difference methods: stability analysis)
  - and coordinate transformations, 158
- Stochastic differential equation (SDE), 4, 5, 8-9
- Stochastic processes, 3-13
  - Ito's formula/lemma, 9-13
  - for processes with jumps, 10-13
  - Markov processes, 5-8
  - stochastic differential equations, 8-9
- Stochastic volatility model, 212-214
- Successive overrelaxation (SOR)
  - iterative solver, 97, 98, 99,100, 124,143,144,145,146,163,213
- SSOR (symmetric), 97, 98
- Taylor series, 10, 44, 62, 67, 76, 106, 154, 159
- Time, backward, 62, 156
- Time advancement, implementation of, 93-100
- Time discretization, 65, 73-76, 93, 106. *See also* Crank-Nicolson scheme
- Total variation diminishing (TVD) algorithms, 61, 134, 135
- Trees, I, 3
- Tridiagonal scheme/solver, 94-95, 98, 124, 185
- Truncation error, 66, 67, 110
- TVD. *See* Total variation diminishing (TVD) algorithms
- Upwinding / upwind discretization, 61, 120, 133, 135
- Variance gamma process model, 153
- Wiener process, 4, 9,17