Modelling Financial Derivatives with *Mathematica*

Mathematical Models and Benchmark Algorithms

WILLIAM T. SHAW
Quantitative Analysis Group
Nomura International plc, London
and Balliol College, Oxford

CAMBRIDGE
UNIVERSITY PRESS

PUBL SHED BY THE PRESS SYND CATE OF THE UN VERS TY OF CAMBR DGE
The Pitt Building, Trumpington Street, Cambridge CB2 1RP, United Kingdom

CAMBR DGE UN VERS TY PRESS
The Edinburgh Building, Cambridge CB2 2RU, UK http://www.cup.cam.ac.uk
40 West 20th Street, New York, NY 10011-4211, USA http://www.cup.org
10 Stamford Road, Oakleigh, Melbourne 3166, Australia

First published 1998

Printed in the United Kingdom at the University Press, Cambridge

Typeset in *Mathematica* 3 and TEX

A catalogue record of this book is available from the British Library

ISBN 0 521 59233 X hardback (with CD-ROM)

NOMURA

for
Susan Mary Wallace
[1946–1997]
and
Sarah-Jane

Contents

Preface

This text has a number of aims. The first is to show how *Mathematica* (version 3 in particular), can be used as a derivatives modelling tool. Second, it presents a complete if concise development of the mathematical approach to the valuation and hedging of a large class of derivative securities. Third, although the basic mathematical development is oriented towards dynamic hedging and partial differential equations, this book aims to present a balanced approach to algorithm development, in which analytical, finite-difference, tree and Monte Carlo methods are each applied in the appropriate context, without any forced adherence to any particular method. Fourth, it is intended that this text collects together and highlights many of the mathematical pathologies that exist in derivatives modelling problems. This last point is all too frequently ignored, so a discussion here may be appropriate.

Financial analysts use often-complex mathematical models to guide their decisions when trading derivative financial instruments. However, derivative securities are capable of exhibiting some diverse forms of mathematical pathology that confound our intuition and play havoc with standard or even state-of-the-art algorithms. The potential traps fall into two categories. The first category contains problems arising from the complexity of some models, leading to their being seriously error-prone in their implementation, even if not intrinsically flawed. The second category contains algorithms that are intrinsically flawed. Let's take a look at some problems in each category.

An obvious example of a type-one problem relates to the computation of hedge parameters, or "Greeks." These are the partial derivatives of the option value with respect to the underlying price and other variables such as time and interest rates. For all but the simplest vanilla options, the pen-and-paper computation of such entities is very complex and therefore error-prone, leading to the potential of errors in coding. The estimation of such quantities by purely numerical methods (differencing) leads to other types of problems associated with inaccuracies in the estimate of the analytical derivative. Such difficulties can often be eliminated in one swoop with the *Mathematica* system, which is able to compute the symbolic derivatives – and hence the hedge parameters – exactly by analytical differentiation of the option-pricing formula.

A more subtle type-one difficulty relates to the computation of implied volatility, which is a favourite parameter of traders. Implied volatility makes sense only for the simplest vanilla options. In other cases, the implied volatility may be unstable, double-valued, or triple-valued, or may even possess infinitely many values. The implementation must check that the price is a strictly increasing or a strictly decreasing function of volatility; otherwise, nonsense can and will be obtained for the implied volatility. In *Mathematica* the graphical tools can be used to test this very quickly.

Some quite well-known algorithms are intrinsically flawed. Problems which we might identify as a type-two issue can be found in the following models.

(i) Binomial models
(ii) Implicit finite-difference models
(iii) Monte Carlo simulation models

These are essentially numerical methods, and this book looks in detail at them in comparison with exact solutions for known cases. This is straightforward in a system such as *Mathematica*, where complex, exact solutions can be expressed exactly and worked out to any degree of precision. As numerical methods, they involve an essential discretization of time and other relevant variables such as the underlying asset price. A common theme is what happens when the time-step is taken to be large, which is very tempting in an implementation in order to obtain results quickly.

For example, several of the standard binomial models suffer from the well-known difficulty that as the time-step becomes large, the probabilities associated with the underlying tree model may become negative, which is manifest nonsense. In other types of models, the asset prices can become negative. Both of these effects are well known. What appears not to be understood is that the reason for these difficulties has a common root in the fact that tree models are typically underspecified from a mathematical point of view. A number of constraints can be written down that should apply to a tree. The solution of a full set can be quite hard, so in practice the authors of tree models have worked with a subset and made up one or more missing conditions in order to solve for the tree structure. This leads to the problems with negative probabilities or negative asset prices. When one is armed with *Mathematica*'s symbolic equation-solving capabilities, the solution of a full set of tree constraints is a straightforward matter – and in fact leads to a model where neither the up-and-down tree probabilities nor the asset price can become negative. Other problems with trees, discovered by others in relation to barrier and cap effects, are also discussed.

One of the most surprising and deeply rooted difficulties relates to the use of implicit finite-difference schemes. In principle, these allow a larger numerical time-step to be used than in treelike models and are becoming increasingly popular. When properly used, they combine accuracy with efficiency. There is, however, a major difficulty with them that appears not to have fully migrated in its appreciation from the academic numerical analysis community to the market practitioners. When the initial conditions for the associated partial differential equation (in financial terms, the option payoff) are nice and smooth (in loose terms, continuous with continuous slopes), one can get away with almost any implicit finite-difference scheme. This is emphatically not the case in option-pricing problems, where the payoffs are typically non-smooth and frequently discontinuous. Such "glitches" in the payoff will propagate through the solution, and while they do not necessarily cause a large error in the option value, they can cause significant errors in the Greeks such as Delta, Gamma, and Theta. This will occur with some of the most common schemes in current use for larger time-steps. It can be avoided only with a certain subset of implicit schemes. Which subset works and which does not is in fact well known to the numerical analysis community. In the text this is made crystal clear by comparison with some exact solutions; and the good, but infrequently used, schemes are contrasted with the bad, but widely used, schemes.

Monte Carlo simulation is a popular method for the valuation of options that are European in style but path-dependent. The manner in which simulated solutions converge to the correct answer is investigated for some cases where the exact solution is known. This reveals several difficulties with such numerical simulation methods, and in particular the very slow convergence associated with certain classes of options. We give suggestions for control variates in a number of useful cases but highlight the difference between getting the variance down – but possibly converging to the wrong answer – and getting the right answer.

However, it would be wrong to assume that the purpose of writing this book was merely to discuss what can go wrong! The illumination of pathology is only one of the abilities of *Mathematica*. For example, in addition to being able to do calculus, *Mathematica* has other advantages over traditional modelling environments such as spreadsheets and C/C++. For example, the presence of a vast library of special functions, coupled with the ability to do differentiation and integration, means that novel, exact solutions can be implemented with ease. A beautiful example of this is the exact solution for the Asian option with arithmetic averaging, which requires that one invert the Laplace transform of a hypergeometric function. This requires just a few lines in *Mathematica* and can be directly differentiated to obtain the Greeks. Other areas in which *Mathematica* can be fruitfully applied include novel analytical techniques for double-Barrier options and accurate analytical approximations for American options.

How the Text is Organized

This book is divided into six groups of chapters. The first group establishes the preliminaries in terms of the use of *Mathematica*, the basics of stochastic calculus and the derivation of partial differential equations, and the basic technique for solving the Black-Scholes PDE family. The next group of chapters explores a wide variety of analytical models, from simple vanilla options, through a range of by-now standard “exotics”, and also develops more complex analytical models for Asian and American options. Next we take a long hard look at the finite-difference models, including the standard approaches and also novel methods with much better numerical characteristics. This block makes particularly good use of the new features of *Mathematica* 3.0, and it is shown how to use the *Mathematica* compiler to build numerical solutions of the PDEs in an efficient manner.

The fourth group of chapters explores the fundamentals and implementation aspects of binomial and trinomial tree models, using *Mathematica* both to define new tree models, and to implement traditional and novel tree models using the compiler. Group five looks in detail at Monte Carlo simulation and applications in particular to path-dependent and Basket options. Finally we take a brief look at some simpler interest-rate models and related non-log-normal equity models.

Some History

The origins of this text are diverse. Many years ago I began running courses for modelling professionals under the auspices of my consulting firm, Oxford System Solutions. Inspired by Ross Miller’s work in *The Mathematica Journal*, I began to look at developing a programme tailored to financial applications, and gave it to several London financial organizations. This course focused largely on the analytical aspects – the limited compilation features of version 2.X of *Mathematica* did not then allow complex numerical models to be developed in an efficient fashion. Later, when employed as a consultant to Nomura Research Institute Europe Ltd., the question of how to carefully test the integrity of the models then being employed by Nomura arose. Although the existing models had been developed and tested with considerable care, I proposed that a systematic sweep through all the existing models be done, using *Mathematica* to independently build all the models, using the basic published mathematical research as a starting point. Furthermore, with one eye on the features of the then forthcoming *Mathematica* 3.0, I realized that one could begin to use *Mathematica* to perform detailed numerical computation, so that the project need not be limited to the simpler models admitting exact solutions. The project scope was then expanded not just to include the existing in-house models, but to explore numerous other models in the literature, with a view to assessing the desirability of their implementation. That extended project led to this text, and continues move to forward now that I am on the staff of Nomura International plc, and involved in the specification, prototyping and testing of a wide range of derivative models.

Technology Aspects

The chapters of this book exist in their entirety as a collection of *Mathematica* 3 Notebooks. All chapter material, including *Mathematica* code, text, graphics and typeset mathematical material, is native to *Mathematica* 3. The front- and end-matter (this preface, contents and index etc.) were prepared in LaTeX using Textures 1.8. The book was produced in final form on Power Macintoshes, in the form of an 8500/120 upgraded with a 266 MHz G3, and a further G3/266, with Notebooks being printed to disk as PostScript files, which were then used by the publisher to produce the final printed version. Timing results are based on the G3/266, running *Mathematica* 3.0.0, which in general is slightly faster on average than a Pentium II at 300 MHz running NT4 (if you are a Windows user make sure that you are using *Mathematica* 3.01 or later, because that version is fully Pentium optimized, and note that NT is significantly more efficient that '95). The timings should therefore be typical of desktop computers in production at the intended publication date of mid-1998. The printed version made use of a few features of the 3.1 or 3.5 system with regard to page layout only. The kernel code is targeted at *Mathematica* 3, though most of the non-compiled material is V2.X friendly. Work in progress in the numerical optimization of future versions of *Mathematica* may modify some of the conclusions regarding numerical efficiency issues.

Accuracy and Errors

In a project of this size and scope it is impossible to guarantee the absolute correctness of all the material and its implementation. I have made significant efforts to check the models contained herein against basic research results and other model implementations, but can make no guarantees regarding these implementations. I have prepared this material both for its educational value, and to provide a set of implementations of valuation models for comparison with other systems. This material should emphatically not be used in isolation for pricing and hedging in real-world applications (see the disclaimer also). Note also that some of the algorithms are highly experimental. Furthermore, it should be noted that all results printed here are those obtained on Apple Power Macintosh systems. A substantial number of the calculations (but not necessarily all) have been re-run on Intel Pentium systems running Microsoft Windows 95 and NT4, and on various UNIX systems from SUN, and have been found to give identical results. However, the author cannot guarantee complete hardware independence. Wolfram Research Inc. make their own best efforts to ensure that the *Mathematica* system operates in a consistent fashion, but there are inevitable minor differences, usually when machine-precision arithmetic is employed.

Stylistic Issues

The coding contained herein is for the most part based on my own efforts, except as explicitly acknowledged within the text. My efforts have focused on accuracy and speed, and I have deemed elegance and compactness to be secondary to transparency of function. In financial applications, for checking purposes, transparency of function is critical, and I hope the code contained here is legible and easy to understand and check. I make no apologies for allegedly ugly code! All that matters to me is getting an accurate answer and getting it efficiently.

Typesetting Issues

Mathematica 3.0 and later versions have a variety of styles for the display of *Mathematica* code and mathematical equations. Except in the early tutorial chapters of this book, where consistency has been the goal in order to avoid confusing the reader, I have been fairly liberal in switching between styles, where it appears to be useful to select a particular style for displaying material. Most *Mathematica* input uses the old version 2.X input form that is pure text, but occasionally, in order, for example, to make it easier to compare input with published research, I have converted input cells to "Standard Form" so that they look more like ordinary mathematics. Similarly, most of the output is in Standard Form, but occasionally it has been converted to "Traditional Form" so that it looks *exactly* like ordinary mathematical notation. Some of the Traditional Form outputs have in addition been typeset as numbered equations. Where there is mathematical material without any related *Mathematica* input or output it is almost all Traditional Form, usually created from Input Form, styled as numbered equations.

One notational point needs to be made here. Mathematica 3 Traditional Form uses a partially double-struck font for symbols such as i and e, and for the d in dS in integrals. I have avoided using this when creating my own equations, e.g. in the stochastic calculus material, but equations that are converted Mathematica output use the default typefaces employed by the software system. Typographical purists may dislike this notation, but I have tried to avoid editing Mathematica-created output wherever possible, in order that "what you see is what *Mathematica* made" or, as we shall remark quickly in the text to remind the reader that something strange and unfamiliar may be about to appear: "WYSIWMAMA".

One decision on presentation was to suppress all the "In" and "Out" numbered statements. This has the benefit of tidyness, but also has the potential for confusion as to what is input and what is output. In the printed form, I have used indentation on most of the outputs to try to indicate their character, but if there is any confusion as to the types or styles of cells, this can be resolved by reference to the electronic form.

Conventions

There are may different issues of convention that plague this subject. For example, how shold Delta be quoted? We could quote the raw partial derivative; the same expressed as a percentage; the same expressed in terms of a one per cent change in the underlying, and so on. The following are the rules, except as explicitly stated in the text:

- All variables are in natural units:
  - the interest-rate and continuous dividend yield are continuously compounded, and expressed in absolute terms, i.e., an interest-rate of 10 per cent continuously compounded corresponds to $r = 0.10$;
  - the time is in years;
  - the volatility is in absolute annual terms, and will normally (but not always) be a number less than unity, so that $\sigma = 0.25$ corresponds to 25 per cent annualized volatility;
- All Greeks are based on the raw partial derivatives with respect to absolute quantities in natural units, so that, e.g.,
  - Delta corresponds to the instantaneous rate of change of option value with respect to the underlying price, with the latter expressed in currency terms – for a vanilla Call Delta lies between zero and one;
  - Rho is rate of change with respect to absolute continuously compounded interest rates;
  - Vega is rate of change with respect to absolute volatility;
  - Theta is rate of change with respect to time in years.

These are most convenient for the mathematical description, as it means there are very few occurences of factors of $100, 365, 1/365$ and so on. In making comparisons with your own on-desk systems, this may require various conversion factors to be applied. Note that if you have numerical differencing algorithms in place, you may have made a choice to calculate actual changes rather than rates of change.

Feedback

Comments are actively sought on this material, especially if material errors are discovered. I also wish to hear about how things could have been done better, particularly with regard to speed and/or accuracy. I am not representing this text as necessarily the best way of implementing models in *Mathematica*, and have not doubt that many others will be able to improve on the material here.

Feedback to: `william.shaw@nomura.co.uk`

All trademarks are acknowledged.

Acknowledgements

I have to begin this list by apologizing to anyone I leave out. Over the past few years, I have had numerous discussions with many colleagues inside and outside Nomura regarding the use of *Mathematica* in both financial and non-financial applications, and I am not going to be able to remember everybody! I will therefore keep this list short. Within the Quantitative Analysis Group in London, my special thanks go to Reza Ghassemieh for his unflagging support throughout the project and to Roger Wilson for helping to solve numerous implementation problems. In the derivatives team, I have to acknowledge the infinite patience of David Kelly, Ben Mohamed and Dominic Pang, for their diverse contributions in the various testing and prototyping phases of the project. Marta Garcia has consistently brought me down to earth with reminders of the complex real world of convertibles and of the limitations of mathematics (and mathematicians). A special thanks goes to James Hutton, for many useful discussions on general points, and for making available early copies of the research on LP methods. Numerous members of the Risk Management teams have provided valuable feedback on model test reports that formed the basis for early drafts of this work. Valuable comments on draft chapters at various stages of development have been received from colleagues inside and outside Nomura, including: Martin Baxter, Ian Buckley, Asif Khan, Jason Tigg, Rachel Pownall, Hideki Shimamoto and my anonymous reviewers. My relatively recent education in finance has benefited from countless discussions with other colleagues at Nomura, and Nick Knight and Allison Southey deserve a special mention, along with numerous past and present members of the equity and strategy teams.

At Wolfram research in the US and the UK, Stephen Wolfram, Conrad Wolfram, Magnus Germandson, Theodore Gray, Rachel Leaver, Claire Miller, Tom Wickham-Jones and many others have provided a mixture of support including enthusiastic noises, organizing presentations, fixing my page layout headaches, fixing my code, and telling bad jokes to warm up my audience before presentations on aspects of this material.

With regard to the book production aspects, David Tranah and the Cambridge University Press team displayed chronic enthusiasm and tolerance.

While this book was being edited for final production, I learnt of the sudden death of my eldest sister Susan. This book is dedicated to her memory and to my niece Sarah-Jane.

Index

Note: this is not a comprehensive index of *Mathematica* commands built in to *Mathematica* – see *The Mathematica Book* also.