Numerical Methods for Nonlinear Equations in Option Pricing

by

David Pooley

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2003

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis explores numerical methods for solving nonlinear partial differential equations (PDEs) that arise in option pricing problems. The goal is to develop or identify robust and efficient techniques that converge to the financially relevant solution for both one and two factor problems. To illustrate the underlying concepts, two nonlinear models are examined in detail: uncertain volatility and passport options.

For any nonlinear model, implicit timestepping techniques lead to a set of discrete nonlinear equations which must be solved at each timestep. Several iterative methods for solving these equations are tested. In the cases of uncertain volatility and passport options, it is shown that the frozen coefficient method outperforms two different Newton-type methods. Further, it is proven that the frozen coefficient method is guaranteed to converge for a wide class of one factor problems.

A major issue when solving nonlinear PDEs is the possibility of multiple solutions. In a financial context, convergence to the viscosity solution is desired. Conditions under which the one factor uncertain volatility equations are guaranteed to converge to the viscosity solution are derived. Unfortunately, the techniques used do not apply to passport options, primarily because a positive coefficient discretization is shown to not always be achievable.

For both uncertain volatility and passport options, much work has already been done for one factor problems. In this thesis, extensions are made for two factor problems. The importance of treating derivative estimates consistently between the discretization and an optimization procedure is discussed.

For option pricing problems in general, non-smooth data can cause convergence difficulties for classical timestepping techniques. In particular, quadratic convergence may not be achieved. Techniques for restoring quadratic convergence for linear problems are examined. Via numerical examples, these techniques are also shown to improve the stability of the nonlinear uncertain volatility and passport option problems.

Finally, two applications are briefly explored. The first application involves static hedging to reduce the bid-ask spread implied by uncertain volatility pricing. While static hedging has been carried out previously for one factor models, examples for two factor models are provided. The second application uses passport option theory to examine trader compensation strategies. By changing the payoff, it is shown how the expected distribution of trading account balances can be modified to reflect trader or bank preferences.

Acknowledgements

First, sincere gratitude is extended to my supervisor, Peter Forsyth. I have benefited greatly from his considerable help and guidance.

Thanks also to my committee members: Andrew Heunis, George Labahn, Cornelis Oosterlee and Ken Vetzal. Important feedback and advice has also been provided by Bruce Simpson, Justin Wan and Ken Vetzal (again) at various stages of this thesis.

Of course, people too numerous to mention have made my time at Waterloo both productive and enjoyable. Members of the scicom lab, both past and present, have always been there when needed. Many fond memories remain of the graduate student research conference, with its diverse, friendly and talented organizing committees. Various other friends have helped me to conquer problems, both real and imagined. Particular mention must be made of Yann, Howard, Heath, Amélie, Dana and Reinhold, who have provided wonderful company on the courts and fields, for idle banter, at the card table, over the computer, in the theaters, and around the dinner table.

Last, and certainly not least, I thank my wife Elizabeth for her support and patience over the past many years. Perhaps now I'll be around for more dinners, and will be able to plan vacations more than two weeks ahead. May we grow closer together as I finally(!) move past the student phase of life.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

The classical no-arbitrage approach for option pricing leads to the celebrated Black-Scholes linear partial differential equation (PDE) for the fair price of an option [10, 39]. While the Black-Scholes equation is still the industry standard, the underlying theory has been extended in at least two ways. First, practitioners and theoreticians alike have been extending the basic assumptions to better reflect observed market behaviour. For example, the underlying stochastic price process can be augmented by jumps [19], or different volatility models can be employed [43].

Second, the Black-Scholes or no-arbitrage approach to option pricing has been applied to a wider range of contingent claim contracts. In theory, contracts can be defined on any measurable quantity. In practice, options based on energy prices, weather patterns or even sporting events are currently being sold [30]. In the context of real options, the no-arbitrage analysis has been applied to investment decisions [20]. Even for traditional options based on stock prices, a multitude of new features have been introduced, such as barriers, guaranteed return levels or reset features.

As option contracts become increasingly complex, the need for robust, efficient and flexible pricing methods becomes more important. In this thesis, numerical techniques for

solving nonlinear option pricing PDEs are studied. To motivate the discussion, two different applications will be examined in detail: the uncertain volatility model and passport options.

Uncertain volatility can be classified as a modification to the basic Black-Scholes model. Instead of assuming that asset volatilities are constant, assume instead that they are bounded between extreme values [31]. When pricing, the seller of the contract is interested in the worst-case scenario for volatility selection (exactly what worst-case means will be discussed later). It will be shown that the resulting pricing equation has a nonlinear diffusion term.

Passport options differ from traditional options in the sense that the payoff is based on the value of a trading account. The option holder trades in an underlying asset for a certain period of time, keeping any net trading gain, but having any net loss covered [6, 36]. The trading strategy of the option holder is not known at contract inception. From a hedging point of view, the pricing equation must anticipate the worst-case trading strategy. As with uncertain volatility, the resulting pricing equation is nonlinear.

In both sample problems, the pricing PDE is nonlinear due to a dependence of equation coefficients on solution derivatives. This form of nonlinearity is common for financial models because of the emphasis on hedging. Recall that in a no-arbitrage setting, the value of a contract is found by constructing a risk-free portfolio (the hedge) that exactly replicates the contingent claim payoff [10]. With uncertain volatility, or for passport options, such a portfolio is impossible to find. However, during the lifetime of the contract the seller can assume that parameters follow the worst possible path for the value of the hedge (uncertain volatility), or that the investor behaves in the worst possible manner for the value of the hedge (passport option). By pricing with these assumptions, the seller can construct a hedge that covers the contingent claim payout in the worst-case scenario, and leaves a positive balance in all other cases. It should be stressed that the worst-case is in relation to the value of the hedge, and is not necessarily related to a maximal payout from the option.

Of course, a best-case pricing can also be performed, and the definition of worst-case and best-case depends on whether you are the buyer or seller of the contract.

When examining the hedge at a particular point in time, an approximation of its evolution can be determined. By assuming some type of optimal behaviour from one instant to the next, the principal of dynamic programming leads to a Hamilton-Jacobi-Bellman equation for the price of the option [22]. While this viewpoint (i.e. stochastic optimal control) is only briefly mentioned in chapter 2, it must be kept in mind when reading the literature for nonlinear financial PDEs. Ultimately, enforcing worst-case or best-case scenarios results in a nonlinear pricing PDE in which the coefficients depend on solution derivative values. This is to be expected, as hedge construction depends on solution derivative values.

During the numerical solution of nonlinear PDEs, there is always the question of solution uniqueness. For example, it is well known that nonlinear conservation law hyperbolic PDEs do not have unique solutions once shocks form. In this case, the physically correct solution satisfies the E-condition [35]. In financial applications, the relevant solution is the viscosity solution [22]. Provided that the nonlinear PDE satisfies certain technical conditions (the strong comparison principle [8]), it is known that a stable, consistent and monotone discretization of financial (non-conservative) PDEs converges to the viscosity solution [8]. These issues will be discussed in some detail in chapter 2.

Unfortunately, monotonicity is a somewhat restrictive condition for numerical schemes. We have been successful in deriving monotonicity conditions for one factor uncertain volatility problems. However, the techniques used do not easily extend to higher dimensional problems. For example, if a linear two factor option pricing problem is discretized using linear basis functions, then in the case of non-constant correlation, the discretized coefficient matrix is not in general an M-matrix [66]. Hence, the discretization in the nonlinear case cannot be monotone. We conjecture that the monotonicity condition may be weakened in certain cases, but proofs of this result will have to await further advances in the theory of

viscosity solutions. In the meantime, the one factor uncertain volatility results can help determine expected behaviour for more difficult problems.

In addition to global convergence to the viscosity solution, there is an issue of convergence for the discrete equations at each timestep. Recall that the use of an implicit discretization method results in a set of nonlinear algebraic equations which must be solved at each timestep. We show that a frozen coefficient iterative scheme is globally convergent for a certain class of problem. This class includes one factor uncertain volatility problems, but does not include passport options. Nevertheless, numerical experiments indicate that the frozen coefficient scheme is rapidly convergent for passport options. The convergence properties of two other Newton iteration type methods are also tested.

Option valuation problems commonly feature discontinuities in either the payoff function or its derivatives. For example, the payoffs of standard vanilla options have discontinuous first derivatives. Contracts where an advance notice of exercise is required (e.g. most callable bonds) also have discontinuous derivative values. Discretely monitored barriers introduce discontinuities at observation dates, while the payoff itself is discontinuous for digital options. This non-smooth data causes *quantization error* [50], and can lead to serious degradation in the convergence rates of numerical schemes. Another consequence can be poor estimates of the solution derivatives (the first derivative w.r.t asset price – delta – and the second derivative w.r.t asset price – gamma) even though the prices appear to be correct. This is especially problematic for nonlinear problems which require accurate derivative estimates in order to optimize certain parameters. To further complicate matters, discontinuities often occur in regions of high interest, such as near the strike. It is often critical to obtain prices and hedging parameters near these asset values. Several methods to alleviate the numerical difficulties caused by non-smooth data will be examined. The methods involve a modified timestepping procedure, and various techniques to smooth the data.

Finally, the discrete equations for the sample problems to be tested have an embedded optimization problem for each node at each timestep. The analytic form of the PDE indicates how the optimization problem needs to be solved. However, in practice, it is the discrete form of the equations that needs to be optimized. It is therefore important to treat the analytic and numerical forms consistently, particularly the discrete derivative values that may be used for the optimization. Details of this issue will be provided, especially for two factor uncertain volatility problems.

Although uncertain volatility and passport options are examined in detail in this thesis, it should be stressed that the techniques and issues raised can be applied to any nonlinear PDE whose coefficients depend on the solution or its derivatives. While some encouraging results have been obtained, many open problems remain.

1.1 Contributions

The main contributions of this thesis are as follows:

- Nonsmooth initial conditions can cause Crank-Nicolson timestepping methods to revert to linear convergence. Techniques to restore quadratic convergence are discussed (see chapter 2).

- When solving a nonlinear PDE implicitly, an iterative method must be used at each timestep to solve the nonlinear discrete equations. Conditions under which a frozen coefficient iterative method is guaranteed to converge at each timestep for a one factor finite difference discretization are derived (see chapter 2).

- In a financial context, the viscosity solution is desired when solving nonlinear PDE models. We prove that implicit timestepping for one factor uncertain volatility problems converges unconditionally to the viscosity solution. Conditions under which

Crank-Nicolson timestepping converges to the viscosity solution are also derived (see chapter 3).

- A procedure to numerically solve two factor uncertain volatility problems, with uncertainty in both underlying factors, is developed. The importance of numerically optimizing the discrete equations, instead of relying on the theoretical form, is discussed. To the author's knowledge, no previous study has examined two factor uncertain volatility (see chapter 4).

- The application of static hedging for two factor uncertain volatility problems is discussed (see chapter 4).

- It is known that positive coefficient discretizations are unconditionally stable (for fully implicit timestepping) and can avoid spurious oscillations in the solution. For convex payoff structures, it is shown that a positive coefficient discretization can be achieved for one factor passport options. However, it is also shown that positive coefficients are not guaranteed for one factor passport option discretizations with general payoffs when using forward, backward and central differencing (see chapter 5).

- The application of passport option theory to the problem of trader compensation is discussed (see chapter 5).

- Via numerical examples, it is shown that the frozen coefficient iterative method [13] has faster convergence than the full (numerical Jacobian) Newton method, or even the non-smooth Newton method [44]. In fact, the Newton methods (full and non-smooth) often failed to converge for both two factor uncertain volatility problems and for passport options (see chapters 4 and 5).

1.2 Outline

The remainder of this thesis is organized as follows. In chapter 2, the basics of the Black-Scholes model are reviewed. A general discretization for one factor and two factor problems, and a discussion of the numerical issues which arise when solving the nonlinear discrete equations, are also presented. Chapters 3, 4 and 5 present the details of the sample problems: one factor uncertain volatility, two factor uncertain volatility, and passport options, respectively. Each of these chapters begins with an overview of the problem to be solved, followed by the problem specific discretization. Theoretical results and numerical results follow. In the case of two factor uncertain volatility, an application to optimal static hedging is provided. For passport options, an application to trader compensation is discussed. Finally, conclusions are drawn in chapter 6.

Chapter 2

Background

The modern analysis of option pricing began with the work of Black and Scholes [10] and Merton [39] in the early 1970s. The resulting model, often called the Black-Scholes model, culminated in a partial differential equation whose solution gives the fair price of a contingent claim. This model is briefly reviewed in section 2.1.

Once a pricing equation has been obtained, it still needs to be solved. Analytic solutions only exist for simple and highly constrained scenarios. In general, the pricing PDE needs to be solved numerically. For one factor problems the finite difference method is used. For two factor problems, a finite volume method is used. Details of these discretization techniques are given in section 2.2. Methods for solving the linear systems of equations that can result from these discretizations are provided in section 2.3.

When solving the final set of discrete equations, one is obviously interested in the accuracy of the solution. To obtain highly accurate solutions, the convergence rate of the methods being used is important. The non-smooth initial conditions present for many option pricing problems can cause difficulties for standard discretization methods. These problems, and methods to restore optimal convergence rates, are discussed in detail in section 2.4.

The original Black-Scholes equation is a linear, parabolic PDE. Some recent financial models have led to nonlinear parabolic pricing equations. Nonlinearities introduce a range of numerical issues. For example, an iterative method must be used to solve the discrete equations. A variety of iterative methods are presented in section 2.5. Of course, it is desirable that the chosen iterative scheme is convergent for the problem being solved. A proof of convergence for the frozen coefficient method for a particular class of nonlinear models is given in section 2.6. Further, nonlinear equations can have more than one solution in general. In a financial context, the viscosity solution is sought. Viscosity solutions are discussed in section 2.7.

2.1 General Structure of Pricing PDEs

In this thesis, both one factor and two factor option pricing problems will be solved. For two factor problems, the value U of the option depends on two stochastic variables – S_1 and S_2 – as well as time t. Following the standard no-arbitrage approach [59], it is assumed that the underlying factors evolve through time under standard geometric Brownian motion:

$$\begin{aligned} dS_1 &= a(S_1, t)dt + b(S_1, t)dW_1 \\ dS_2 &= a(S_2, t)dt + b(S_2, t)dW_2, \end{aligned} \tag{2.1}$$

where $a(S_i, t)$ is the drift, $b(S_i, t)$ is the diffusion, and W_i is a Wiener process for underlying factor i. The two Wiener processes are assumed to have correlation factor ρ.

Based on these stochastic processes, a partial differential equation can be derived whose solution gives the fair price of the option. Details can be found in [67] and the references therein. The result is

$$U_t - \mathbf{V} \cdot \nabla U + (\mathbf{D}\nabla) \cdot \nabla U - rU = 0, \tag{2.2}$$

where $\mathbf{V}$ is the velocity/convection vector, $\mathbf{D}$ is the diffusion tensor, and r is the risk-free

interest rate. Since the value of the option is usually known at expiry time $t = T$, it is standard to let $\tau = T - t$ and to rewrite equation (2.2) backward in time as

$$U_\tau = -\mathbf{V} \cdot \nabla U + (\mathbf{D}\nabla) \cdot \nabla U - rU. \tag{2.3}$$

When written this way, the option payoff becomes the PDE initial condition. Boundary conditions must also be specified, but will depend on the particular option being priced.

For options which depend on only one factor, equation (2.3) simplifies to

$$U_\tau = v(S,t)U_S + d(S,t)U_{SS} - rU. \tag{2.4}$$

For future reference, $v(S,t)U_S$ is called the convection term, $d(S,t)U_{SS}$ is the diffusion term, while $-rU$ is the decay term. Note also that in the financial literature, U_S is called the option delta, and U_{SS} is the option gamma.

As a specific example, consider an option written on an underlying asset S that follows the process

$$dS = \mu S dt + \sigma S dW, \tag{2.5}$$

where μ is the expected drift rate and σ is the volatility of the asset. Equation (2.4) becomes

$$U_\tau = rSU_S + \frac{1}{2}\sigma^2 S^2 U_{SS} - rU, \tag{2.6}$$

which is the classical Black-Scholes equation for an option on a single underlying asset.

So far, all PDEs presented have been linear. However, as mentioned in the introduction, several problems in finance require a worst-case analysis, or involve some kind of optimal behaviour from one timestep to the next. In the cases to be considered, the one factor

pricing PDE (2.4) assumes the form

$$\frac{\partial U}{\partial \tau} = -rU + \max_q \left\{ v(q,S,\tau)\frac{\partial U}{\partial S} + d(q,S,\tau)\frac{\partial^2 U}{\partial S^2} \right\} \tag{2.7}$$

or

$$\frac{\partial U}{\partial \tau} = -rU + \min_q \left\{ v(q,S,\tau)\frac{\partial U}{\partial S} + d(q,S,\tau)\frac{\partial^2 U}{\partial S^2} \right\}, \tag{2.8}$$

where $q = q(U, U_S, U_{SS})$ represents a variable to be optimized. In other words, the convection or diffusion coefficients depend on the solution or its derivatives (making a nonlinear PDE), and the convection and/or diffusion terms need to be maximized or minimized. For example, in the one factor uncertain volatility case to be considered in chapter 3, $v(q,S,\tau)$ is simply rS while $d(q,S,\tau)$ becomes $[\sigma(U_{SS})]^2 S^2/2$. Whether or not equation (2.7) or (2.8) is being solved depends on whether or not the investor is long or short, and whether or not a best-case or worst-case analysis is being performed.

For two factor problems, the option pricing PDE (2.3) assumes the form

$$U_\tau = \sup\left\{ -\mathbf{V}\cdot\nabla U + ((\mathbf{D}\nabla)\cdot\nabla U) \right\} - rU \tag{2.9}$$

or

$$U_\tau = \inf\left\{ -\mathbf{V}\cdot\nabla U + ((\mathbf{D}\nabla)\cdot\nabla U) \right\} - rU, \tag{2.10}$$

where $\mathbf{V}$ and $\mathbf{D}$ depend on the solution or its derivatives, as well as the underlying factors and time. Theoretically, this is exactly the same as the one factor case, although the numerical maximizing or minimizing of the convection and/or diffusion terms is more challenging.

When the PDE is nonlinear, the equations must be solved numerically using an iterative scheme, and more than one solution is possible. These issues will be discussed below. Before this, however, discretization techniques for the linear equations (2.3) and (2.4) will

be reviewed. In the nonlinear case (equations (2.7),(2.8) and (2.9),(2.10)), the discretized equations are identical in form to the linear equations, except that the discrete coefficients need further specification.

2.2 Numerical Discretization

Assuming a European style option, equation (2.4) can be discretized by a standard one factor finite difference method with variable timeweighting to give

$$\begin{aligned} U_i^{n+1} - U_i^n = (1-\theta)\left[(-\alpha_i^{n+1} - \beta_i^{n+1} - r\Delta\tau)U_i^{n+1} + \alpha_i^{n+1}U_{i-1}^{n+1} + \beta_i^{n+1}U_{i+1}^{n+1}\right] \\ + \theta\left[(-\alpha_i^n - \beta_i^n - r\Delta\tau)U_i^n + \alpha_i^n U_{i-1}^n + \beta_i^n U_{i+1}^n\right]. \end{aligned} \tag{2.11}$$

Examples in this thesis are restricted to fully implicit ($\theta = 0$) or Crank-Nicolson ($\theta = 1/2$) timestepping schemes. Although second order BDF methods have also been used in financial applications [40], work in [61] demonstrated that second order BDF methods may behave poorly for American options with time dependent exercise features. Hence, second order BDF methods will not be considered, although this remains as an avenue for future research. The forms of α_i and β_i depend on the choice of finite difference stencil, and the particular contract being priced. Exact representations will be provided when models are fully developed.

Equation (2.11) can also be rewritten in the form

$$U_i^{n+1} = c_i^n U_i^n + \sum_{j \in i-1, i+1} c_j^n U_j^n + \sum_{j \in i-1, i+1} c_j^{n+1} U_j^{n+1}. \tag{2.12}$$

According to [32], a positive coefficient discretization is one for which the coefficients of $\{U_i^n, U_j^n, U_j^{n+1}\}$ in equation (2.12) are non-negative and sum to one. Positive coefficient discretizations have the desirable property of not generating spurious oscillations in the

solution. Under this definition, equation (2.11) satisfies the properties for a discounted positive coefficient scheme (the coefficients sum to less than one due to the discounting term [24]) if

- $\alpha_i \geq 0$ and $\beta_i \geq 0$
- $\theta = 0$, or $\theta > 0$ and the timestep is sufficiently small so that $[1-\theta(\alpha_i^n+\beta_i^n+r\Delta\tau)] \geq 0$.

Nevertheless, for the purposes of showing stability or obtaining an M-matrix, such a definition is overly strict. For example, it is easy to show using a maximum analysis that if $\alpha_i > 0$ and $\beta_i > 0$ in equation (2.11), then a fully implicit timestepping method is unconditionally stable, even though the equation is nonlinear. The following definition is therefore used in the remainder of this thesis:

Definition 2.1 (Positive Coefficient Discretization). *Equation* (2.11) *is a positive coefficient discretization if* $\alpha_i \geq 0$ *and* $\beta_i \geq 0$.

In an attempt to attain positive coefficients, the first derivative term of equation (2.4) is discretized by a central, forward, or backward difference approximation as appropriate. Let $\Delta S_i^+ = S_{i+1} - S_i$ and $\Delta S_i^- = S_i - S_{i-1}$. The different first derivative approximations include

$$\begin{aligned}(U_S)^n_{i,central} &= \frac{U^n_{i+1} - U^n_{i-1}}{\Delta S_i^+ + \Delta S_i^-} \\ (U_S)^n_{i,forward} &= \frac{U^n_{i+1} - U^n_i}{\Delta S_i^+} \\ (U_S)^n_{i,backward} &= \frac{U^n_i - U^n_{i-1}}{\Delta S_i^-}.\end{aligned} \tag{2.13}$$

For the second derivative term a central difference approximation is always used:

$$(U_{SS})^n_i = \Gamma^n_i = \sum_{j \in i-1,i+1} \frac{2(U^n_j - U^n_i)}{(S_{i+1} - S_{i-1})|S_j - S_i|}. \tag{2.14}$$

Note that a finite volume approach as in [67] would lead to the same form of discretization as equation (2.11). Wherever forward/backward differencing is used, a finite volume approach would use upstream weighting. On a uniform grid in one space dimension, the finite volume method and finite difference method give identical discretizations. On non-uniform grids, upstream weighting and forward/backward differencing give slight differences, although the form of the discretizations remains identical. One advantage of finite volume methods is the easy extension to higher order flux limiters. However, for typical financial parameters and grid spacings, forward/backward differencing is rarely required for single factor options. On the other hand, a flux limiter can be highly beneficial for multi-factor options, especially in the case of convection dominated problems such as Asian options [67]. Since a prime focus of this thesis is model nonlinearity, matters are not complicated by adding nonlinear flux limiters, although these methods may be beneficial in certain cases.

For ease of notation in single factor analyses, a finite difference discretization has been used. Note again that forward/backward differencing is only used at nodes where $\alpha_{i,central} < 0$ or $\beta_{i,central} < 0$. In practice, since this occurs at only a small number of nodes remote from the region of interest, the limited use of a low order scheme does not result in poor convergence as the mesh is refined for the types of problem considered in this thesis. Verification of convergence rates will be done via numerical experiments. As will be seen in chapter 3, requiring that all α_i and β_i are non-negative can have important theoretical ramifications.

For two factor problems, a finite volume discretization has been used [67]. Specifically, consider a discrete two dimensional computational domain $\mathcal{D}$ which is tiled by triangles.

Let N_i be the usual C^0 Lagrange basis functions defined on triangles. Then,

$$\begin{aligned} N_i &= 1 \text{ at node } i \\ &= 0 \text{ at all other nodes} \\ \sum_j N_j &= 1 \text{ everywhere in the solution domain.} \end{aligned} \tag{2.15}$$

If $U^n = \sum_j U_j^n N_j$, where $U_j^n = U(S_{1j}, S_{2j}, \tau^n)$ is the value of U at $(S_{1j}, S_{2j}, \tau = n\Delta\tau)$, then a discretization of equation (2.3) is given by:

$$\begin{aligned} A_i \left(\frac{U_i^{n+1} - U_i^n}{\Delta\tau} \right) &= (1-\theta) \left(\sum_{j \in \eta_i} \gamma_{ij} (U_j^{n+1} - U_i^{n+1}) + \mathbf{V}_i \cdot \sum_{j \in \eta_i} \vec{L_{ij}} U_{ij+1/2}^{n+1} - A_i r U_i^{n+1} \right) \\ &\quad + \theta \left(\sum_{j \in \eta_i} \gamma_{ij} (U_j^n - U_i^n) + \mathbf{V}_i \cdot \sum_{j \in \eta_i} \vec{L_{ij}} U_{ij+1/2}^{n} - A_i r U_i^{n} \right), \end{aligned} \tag{2.16}$$

where

$$\begin{aligned} A_i &= \int_{\mathcal{D}} N_i d\mathcal{D} \\ \Delta\tau &= \text{timestep} \\ \theta &= \text{timeweighting} \\ &\quad \theta = 0 \text{ fully implicit} \\ &\quad \theta = 1/2 \text{ Crank-Nicolson} \\ U_i^{n+1} &= U(S_{1i}, S_{2i}, \tau^{n+1}) \\ \gamma_{ij} &= -\int_{\mathcal{D}} \nabla N_i \cdot \mathbf{D}_i \cdot \nabla N_j d\mathcal{D} \\ \eta_i &= \text{set of neighbours of node } i \\ \vec{L_{ij}} &= \int_a^b \hat{n} ds \\ U_{ij+1/2}^{n+1} &= \text{value of } U \text{ at the face between node } i \text{ and node } j. \end{aligned}$$

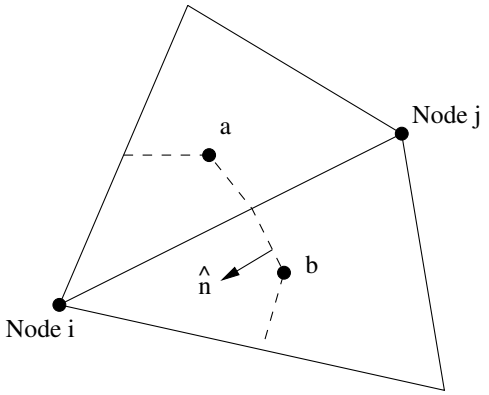


FIGURE 2.1: *Part of the finite volume surrounding node i. Points a and b are at triangle centroids. Connections are made to/from edge mid-points.*

The remaining variables are shown graphically in Figure 2.1. Central weighting is used for the $U^{n+1}_{ij+1/2}$ term. Note that upstream weighting was never required (or had negligible impact on the overall solution) for the problems to be tested in this thesis. Mass lumping (A_i is the sum of row i in the normal 'mass' matrix, see [63]) was used for the time derivative term. For further details of this discretization, and a discussion of the advantages of this approach compared to a simpler two factor finite difference discretization, see [67].

Analyzing the general form of the two factor equations is quite difficult. Further, even if the discretized equations are diffusion dominated, if the cross-derivative terms are non-zero, a positive coefficient scheme may be impossible to obtain [68]. Nevertheless, we can proceed with numerical experiments to determine how the equations behave in practice.

2.3 Solution of Linear Systems of Equations

When implicit timestepping methods are used, a system of equations must be solved at each timestep. For linear PDEs, the resulting system is linear. For nonlinear PDEs, a nonlinear system of discrete equations arises. If an iterative method (e.g. Newton's method) is used to solve the nonlinear system, a set of linear equations occurs at each iteration. Hence, for

either linear or nonlinear PDEs, a series of linear systems need to be solved.

For one factor problems, a direct method is used to solve the linear systems. In fact, since the matrix coming from discretization (2.11) will be tri-diagonal, solving the matrix is relatively quick and easy.

For two factor problems, an iterative solver is used. Specifically, a level(1) ILU preconditioner with Bi-CGSTAB acceleration [55] is employed. Nodes are ordered with RCM ordering [25]. The convergence tolerance for the Bi-CGSTAB iteration is based on

$$\frac{||r^k||_2}{||r^0||_2} < tol, \tag{2.17}$$

where $||r^k||_2$ is the L_2 residual at the k^{th} iteration. Unless otherwise stated, the tolerance factor tol is set to 10^{-6}.

2.4 Smoothing Initial Conditions

As mentioned in the introduction, many financial problems feature non-smooth initial data which can cause convergence problems for classical numerical methods. Several techniques have been proposed to address these problems. In the context of timestepping, Rannacher [46] proposed a scheme in which Crank-Nicolson timestepping is preceded by a finite number of implicit steps. The rationale is that high frequency error components will be dampened by the implicit steps, leading to smooth convergence. The expected rate of convergence remains quadratic since only a finite number of implicit steps are taken. Furthermore, this type of timestepping can help eliminate oscillations in the solution derivative values. Effective estimation of the hedging portfolio for the underlying contract is then easier to compute.

Unfortunately, simply modifying the timestepping procedure is not enough to restore expected convergence rates. Some type of *smoothing* of the data must also be performed.

Perhaps the first study to look at this was [51], which used an averaging method to improve convergence. In [50], it was suggested that shifting the grid such that discontinuities occur midway between grid points can increase accuracy. Note that this idea is similar to placing nodes equidistantly from a discretely observed barrier [12]. Perhaps the most general method for handling discontinuities was given in [58]. Under this method, the initial conditions undergo an L_2 projection onto the space spanned by a given set of basis functions.

Although the idea of smoothing the payoff has been discussed before in the finance literature (see [28]), this was in the context of vanilla payoffs using explicit type numerical methods which are generally only first order accurate in time. Higher order rates of convergence can sometimes be obtained [28], but require special circumstances (i.e. constant volatility). By contrast, we are interested in cases with discontinuous payoffs, and in the use of Crank-Nicolson type methods which can theoretically attain second order convergence for any volatility structure. We show that this higher order rate can be achieved by employing a combination of some form of smoothing *and* the Rannacher timestepping method. As a byproduct, we show how known partial differential equation (PDE) evaluation and analysis techniques can explain and improve upon the observed behaviour of other pricing methods (such as lattices).

For ease of exposition, focus is maintained on one asset digital call options and one asset supershare binary call options written on an underlying asset S (extensions for multi-factor contracts are briefly discussed in section 2.4.3). However, it is important to note that these methods can be used for any problem with non-smooth initial conditions.

The payoff of a digital option is defined by

$$V(t=T) = \begin{cases} 1.0 & \text{if } S \geq K \\ 0.0 & \text{otherwise,} \end{cases} \tag{2.18}$$

where K is the strike price. Instead of having a single discontinuity, a supershare payoff has two, and is specified by

$$V(t=T) = \begin{cases} 0.0 & \text{if } S < K \\ 1.0/d & \text{if } K \le S \le K+d \\ 0.0 & \text{if } S > K+d, \end{cases} \tag{2.19}$$

where K plays the role of a strike price and d gives the width of the (positive) payoff region.

The value of both digital and supershare options satisfies the usual Black-Scholes PDE (2.6) with appropriate initial and boundary conditions. To be consistent with the one factor and two factor examples, and anticipating the theoretical requirements of the smoothing methods, equation (2.6) is discretized using the finite volume method as in equation (2.16). In the one factor case, the N_i linear Lagrange basis functions are simply the usual hat functions. Further, the following simplifications to equation (2.16) can be made:

$$\begin{aligned} A_i &= \int_{\mathcal{D}} N_i dS = \frac{S_{i+1} - S_{i-1}}{2} \\ V_i &= rS_i \\ \gamma_{ij} &= -\int_{\mathcal{D}} \nabla N_i \cdot \frac{1}{2}\sigma^2 S_i^2 \cdot \nabla N_j dS = \frac{\sigma^2 S_i^2}{2|S_i - S_j|} \\ L_{ij} &= \begin{cases} +1 & \text{if } j = i+1 \\ -1 & \text{if } j = i-1. \end{cases} \end{aligned}$$

The value for $U^{n+1}_{ij+1/2}$ is computed by central differencing or upstream weighting as appropriate to ensure positive coefficients (although upstream weighting is rarely required).

This discretization may appear overly cumbersome for the simple one factor problems to be studied. Nevertheless, on an equally spaced grid in one space dimension, the standard finite difference, finite element (with mass lumping), and finite volume methods all give

identical discretizations for constant coefficient equations. Constant coefficients can be obtained for equation (2.6) by using the transformation $x = \log S$. For unequally spaced grids, slight differences may occur if upstream weighting (forward/backward differencing) is used for the convection term. Nevertheless, the forms of the discretizations remain the same. In higher dimensions, a finite volume discretization can be considered to be a Galerkin finite element method with a special quadrature rule [23]. Hence, the finite element results to which we appeal later hold for any standard discretization method in one dimension (equally spaced grid, constant coefficients), and are expected to hold for any standard discretization in the more general settings.

The three methods for smoothing the non-smooth data are given below. In all cases, f_i denotes the initial data value at node i on the discrete grid. The function $f(S) = f$ denotes the exact initial conditions, which are assumed to be known at all spatial points.

Averaging The Initial Conditions A simple method for handling discontinuities involves averaging the initial data. Specifically, nodal values are replaced with an average value over nearby space. Mathematically, we set

$$f_i = \frac{1}{S_{i+1/2} - S_{i-1/2}} \int_{S_{i-1/2}}^{S_{i+1/2}} f(y)dy, \tag{2.20}$$

where $S_{i+1/2}$ denotes the point halfway between S_i and S_{i+1}. The validity of this method was proven in [33] and [51]. It was used in [28] to increase the rate of convergence when pricing vanilla call options using binomial trees.

Shifting The Mesh In [50], it was remarked that having the strike price occur midway between mesh nodes generally increases the accuracy of the finite difference method. For an intuitive explanation of this result, consider a digital call option with a strike of 40 dollars, and grid points 39, 40 and 41. In this case, the initial values at these nodes are 0, 1 and 1 respectively. However, the exact same initial values would occur

on this grid for any strike price in the range $(39, 40]$. In other words, this grid cannot resolve differences between digital options with strike prices just over 39 dollars and up to (and including) 40 dollars. On the other hand, consider a shifted grid with points at 39.5 and 40.5. A strike price of 40 then falls halfway between the range of resolvable prices. The maximum possible error is therefore reduced from the previous grid.

On a more mathematical side, the averaging procedure defined by equation (2.20) would have no effect on a grid in which the strike price is halfway between grid points. In a sense, the initial data has already been smoothed, and we might expect the shifted grid to give the expected rate of convergence. However, refining the grid is no longer trivial. Simply adding nodes halfway between existing nodes places a new node at the strike price. To maintain the desired properties of the grid, the refined grid must then be shifted such that the strike price once again falls between two grid nodes. Interpolation can be used to recover the option value at the strike price.

Projecting The Initial Conditions Perhaps the most systematic technique for handling discontinuities involves projecting the initial conditions onto a set of basis functions [46]. Let N_i be the basis functions used in the discretization. Typically, the N_i are the linear Lagrange basis functions defined by properties (2.15). The continuous form of a discrete function can then be written as

$$c = \sum_k c_k N_k. \tag{2.21}$$

The best (in the L_2 norm) approximation of the exact initial conditions f in the space spanned by the N_k basis functions is found by selecting c_k values to minimize

$$\int_{\mathcal{D}} (c - f)^2 dS, \tag{2.22}$$

where $\mathcal{D}$ is still the computational domain. Let $\langle x, y\rangle = \int_{\mathcal{D}} xy\, dS$ denote the inner product between functions x and y. The solution to problem (2.22) is also the solution to the linear system $Mc = F$ where

$$M = \begin{bmatrix} \langle N_1, N_1\rangle & \langle N_1, N_2\rangle & \cdots & \langle N_1, N_n\rangle \\ \langle N_2, N_1\rangle & \langle N_2, N_2\rangle & \cdots & \langle N_2, N_n\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle N_n, N_1\rangle & \langle N_n, N_2\rangle & \cdots & \langle N_n, N_n\rangle \end{bmatrix} \quad \text{and} \quad F = \begin{bmatrix} \langle N_1, f\rangle \\ \langle N_2, f\rangle \\ \vdots \\ \langle N_n, f\rangle \end{bmatrix}. \tag{2.23}$$

For the simple linear Lagrange basis functions, the entries of M can be computed analytically. The result is

$$\frac{1}{6}\begin{bmatrix} 2(S_2 - S_1) & (S_2 - S_1) & 0 & 0 & \cdots & 0 \\ (S_2 - S_1) & 2(S_3 - S_1) & (S_3 - S_2) & 0 & \cdots & 0 \\ 0 & (S_3 - S_2) & 2(S_4 - S_2) & (S_4 - S_3) & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & (S_n - S_{n-1}) & 2(S_n - S_{n-1}) \end{bmatrix}. \tag{2.24}$$

Matrix M is obviously symmetric and diagonally dominant. It is also positive definite and therefore non-singular. Further, by virtue of being tridiagonal, it is relatively quick and simple to solve the resulting linear system.

Entries of F will of course depend on the payoff function f. For a known f, the entries can be computed analytically. However, it is advantageous to code a black-box routine that can accept any payoff form. If f is piecewise linear, then a second order integration scheme will be exact when using linear basis functions. For example,

Simpson's rule can be used over each grid interval as follows:

$$\begin{aligned}\langle N_i, f\rangle &= \int_{S_{i-1}}^{S_i} N_i f\,dx + \int_{S_i}^{S_{i+1}} N_i f\,dx \\ &= \frac{S_i - S_{i-1}}{2}\left[\frac{4}{6}f(S_{i-1/2}) + \frac{1}{3}f(S_i)\right] \\ &\quad + \frac{S_{i+1} - S_i}{2}\left[\frac{4}{6}f(S_{i+1/2}) + \frac{1}{3}f(S_i)\right], \end{aligned} \tag{2.25}$$

where $f(S_{i-1/2})$ is the value of the payoff evaluated halfway between S_i and S_{i-1}. In practice, for this scheme to be exact for piecewise linear payoff functions, there are two further restrictions:

(i) All discontinuities and/or kinks in the payoff must be aligned with grid nodes.

(ii) To properly handle discontinuities, one must evaluate $f(S_i - \epsilon)$ for the left integral, and $f(S_i + \epsilon)$ for the right integral, instead of $f(S_i)$ in equation (2.25). Here, ϵ is some small (positive) value.

From a theoretical point of view, a projection operator P applied to the initial data f has the property that $P^2 f = Pf$. In other words, one application of the projection operator makes the data as smooth as possible in the L_2 norm. On the other hand, the data resulting from the averaging operator (2.20) may be oscillatory. One may be tempted to average the new data (assuming linear interpolation between the data values), but then it is not clear when this process should end.

The projection method will also be used for smoothing two factor problems. In this case, the linear basis functions are defined on triangles, as in discretization (2.16). The integrals in expression (2.23) are evaluated using quadrature rules that are exact for quadratic functions.

2.4.1 Non-smooth Convergence Results From Finite Element Analysis

It is a commonplace idea that finite difference methods have theoretical difficulties when presented with non-smooth payoffs. However, as previously mentioned, the standard discrete finite difference equations for constant coefficient one dimensional PDEs are identical to a finite element discretization with linear basis functions and mass-lumping for the time derivative term. Consequently, we can appeal to a finite element convergence analysis, rather than relying on the classical finite difference analysis which assumes smooth data. We can expect that finite element methods will require less continuity than finite difference methods, since finite element methods are based on the weak or integrated form of the PDE. Strictly speaking, the results to which we appeal do not directly apply to the general finite difference discretization of equation (2.11), or to the finite volume discretization in equation (2.16). Nevertheless, problems are not anticipated when using these discretizations, and numerical examples have not encountered any deviation from the finite element analysis.

The main relevant results (see [46] for the proof) are as follows. Consider a linear parabolic PDE, with a non-smooth initial condition (which may contain a finite number of discontinuities). Assuming that the discretization uses linear basis functions and Crank-Nicolson timestepping, second order convergence to the exact solution is obtained for any finite value of $\tau = \tau_0 > 0$, if

(i) the initial condition (i.e. the payoff) is L_2 projected onto the space of basis functions; and

(ii) two timesteps of fully implicit timestepping are used, followed by Crank-Nicolson (which we refer to as *Rannacher* timestepping).

The projection and Rannacher timestepping must occur at all non-smooth states, including whenever a discontinuity is introduced. For digital options, this just means at the

start, but in the case of a discretely monitored barrier option, it would be at each monitoring date.

For vanilla payoffs, which are continuous but have discontinuous derivatives, *no smoothing is required* if there is a node at the strike. This is contrary to the suggestions in [28]. In this case, since the payoff can be represented by piecewise linear functions consistent with the C^0 basis functions implied by a finite difference method, the projection operation does not alter the initial data (of course, in the case of a discontinuous payoff, projection does change the initial data).

The above results explain the erratic convergence of the standard binomial lattice method [30], which is simply an explicit finite difference method on a log spaced grid. When pricing at-the-money options, there is a node at the strike only if the number of timesteps is even. For an odd number of timesteps, the payoff is not in the space of linear basis functions implied by the grid.

It is also important to note that these results only hold for linear PDEs. For the nonlinear PDEs to be studied in this thesis, quadratic convergence is not guaranteed. However, to have any hope of higher order convergence, these techniques must be used. Further, as future examples will demonstrate, Rannacher timestepping (and to a lesser extent the smoothing techniques) can help the convergence of the nonlinear equations.

2.4.2 Convergence Results for Non-smooth Data

Digital Call Options

The parameters used for the digital option tests (see equation (2.18)for the payoff) are listed in Table 2.1. Convergence results are presented for a sequence of grids starting with 41 (non-uniform) nodes, and uniformly refined to 641 nodes. Constant timestepping is used, with a step-size of 0.02 for the original grid. The step-size was then cut in half for subsequent refined grids. Note that the ratio of changes in the solution for successive grid refinements

Type	Call
Time to expiry	0.5 years
r	0.05
K	40
σ	0.3
Payoff amount	1.0

TABLE 2.1: *Model parameters for the binary call option.*

	Implicit			Crank-Nicolson			Rannacher		
Nodes	Value	Difference	Ratio	Value	Difference	Ratio	Value	Difference	Ratio
41	0.5044549			0.5027482			0.5042929		
81	0.4981564	0.0062985		0.4996981	0.0030501		0.4981224	0.0061705	
161	0.4951500	0.0030064	2.10	0.4967380	0.0029601	1.03	0.4951438	0.0029786	2.07
321	0.4936830	0.0014670	2.05	0.4952843	0.0014537	2.04	0.4936827	0.0014612	2.04
641	0.4929575	0.0007255	2.02	0.4945815	0.0007028	2.07	0.4929580	0.0007247	2.02

TABLE 2.2: *Convergence results for an at-the-money ($S = 40$) digital call option using various timestepping methods. Data as in Table 2.1. The exact value is 0.4922403. Difference is the absolute value of the change in the solution as the grid is refined. Ratio is the ratio of successive differences.*

will be 2 for a linearly convergent scheme; by contrast with quadratic convergence this ratio will be 4.

To begin, we test the effect of the initial discontinuity on standard timestepping methods. In Table 2.2, we see that implicit timestepping leads to consistent first order convergence. This is to be expected given the known convergence properties of implicit timestepping. On the other hand, while Crank-Nicolson timestepping has expected quadratic convergence, erratic linear convergence is observed. Using the Rannacher method restores smooth convergence, but only at a linear rate.

In Table 2.3, we show the results of the three methods for smoothing discontinuities, combined with pure Crank-Nicolson timestepping. All methods appear to have higher convergence, but the rates jump around to such a degree that the convergence is unreliable. In particular, it would be impossible to use the results to extrapolate to a better solution.

Given that the Rannacher timestepping method appears to be beneficial, the tests were repeated with the three smoothing methods combined with Rannacher timestepping. Ta-

	Averaging			Shifting Mesh			Projection		
Nodes	Value	Difference	Ratio	Value	Difference	Ratio	Value	Difference	Ratio
41	0.4928843			0.4926810			0.4928886		
81	0.4923661	0.0005182		0.4923915	0.0002895		0.4923299	0.0005587	
161	0.4922658	0.0001003	5.17	0.4922847	0.0001068	2.71	0.4922505	0.0000794	7.04
321	0.4922437	0.0000221	4.54	0.4922545	0.0000302	3.53	0.4922368	0.0000138	5.77
641	0.4922397	0.0000040	5.47	0.4922454	0.0000091	3.32	0.4922364	0.0000004	34.4

TABLE 2.3: *Convergence results for an at-the-money ($S = 40$) digital call option using Crank-Nicolson timestepping and various methods for smoothing the payoff. Data as in Table 2.1. The exact value is 0.4922403.*

	Averaging			Shifting Mesh			Projection		
Nodes	Value	Difference	Ratio	Value	Difference	Ratio	Value	Difference	Ratio
41	0.4928350			0.4927280			0.4927897		
81	0.4923899	0.0004451		0.4923676	0.0003604		0.4923787	0.0004111	
161	0.4922778	0.0001121	3.97	0.4922728	0.0000948	3.80	0.4922750	0.0001037	3.97
321	0.4922497	0.0000281	3.99	0.4922485	0.0000243	3.91	0.4922490	0.0000260	3.99
641	0.4922427	0.0000070	4.00	0.4922424	0.0000061	3.96	0.4922425	0.0000065	4.00

TABLE 2.4: *Convergence results for an at-the-money ($S = 40$) digital call option using Rannacher timestepping and various methods for smoothing the payoff. Data as in Table 2.1. The exact value is 0.4922403.*

ble 2.4 shows that all of the methods now give smooth quadratic convergence.

Rannacher timestepping also has a beneficial effect on computed delta and gamma values. During testing, a variety of grids and parameter values showed oscillations in delta and gamma when Crank-Nicolson timestepping was used, even if the payoff was smoothed. For example, Figure 2.2 plots the gamma of an option using both Rannacher timestepping and Crank-Nicolson timestepping when the initial values are smoothed with the averaging method. This example is based on the same data as in Table 2.1. We see that Crank-Nicolson timestepping leads to severe oscillations. However, these oscillations disappear when Rannacher timestepping is used.

The effects on the initial data of the averaging method and grid shifting are obvious. The effect of projecting the initial conditions onto the linear basis functions is shown in Figure 2.3. We see that the projected values overshoot and undershoot the exact values around the strike price. The result is similar to a truncated Fourier series of a step function. Comparing the projected values on the first mesh versus the fourth refined mesh, we see that

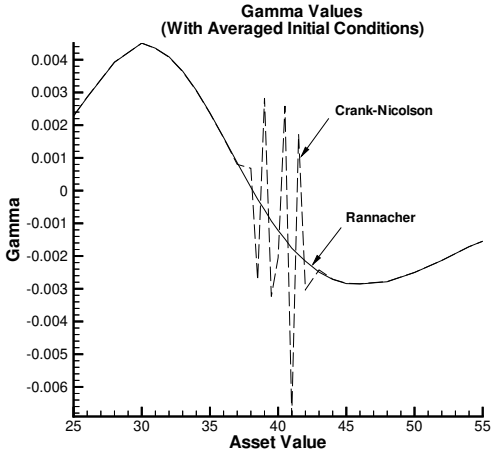


FIGURE 2.2: *Comparison of gamma values at $t = 0$ when using Crank-Nicolson timestepping (dashed line) and Rannacher timestepping (solid line) for averaged initial data. Model parameters as in Table 2.1.*

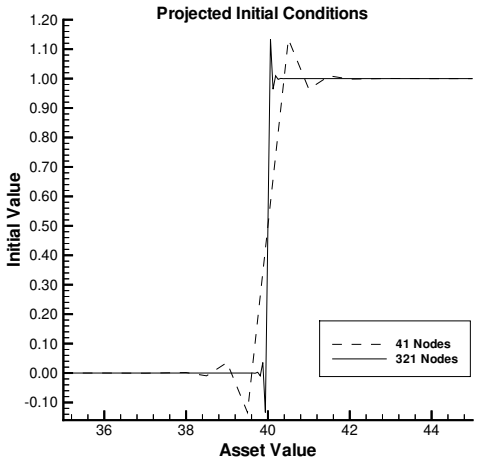


FIGURE 2.3: *Projected initial conditions for the original and fourth refined mesh for a digital payoff.*

changes only extend about five nodes beyond the strike price in either direction. The grid spacing and values are not important. While at first sight it is disturbing that the projected values undershoot and overshoot the payoff near the discontinuity, these local extrema are rapidly damped with Rannacher timestepping (as demonstrated by Figure 2.2).

Supershare Binary Call Options

To indicate that the results of the previous section are not coincidental, we also test some supershare binary call options. The payoff a supershare call was given in equation (2.19).

Parameters for the problem are taken from [18] and are summarized in Table 2.5. Given the two discontinuities (at K and $K + d$), we expect this problem to be more difficult than a regular digital call option.

Similar to the digital call option results above, solution values are presented for a sequence of grids starting with 65 (non-uniform) nodes, and uniformly refined to 1025 nodes. The constant stepsize of 0.02 for the original grid was cut in half for each grid refinement.

Type	Call
Time to expiry	1.0 year
r	0.05
K	10
σ	0.2
d	3

TABLE 2.5: *Model parameters for the supershare binary call option.*

	Implicit			Crank-Nicolson			Rannacher		
Nodes	Value	Difference	Ratio	Value	Difference	Ratio	Value	Difference	Ratio
65	0.149504			0.149237			0.149248		
129	0.144083	0.005421		0.143952	0.005284		0.143955	0.005293	
257	0.141330	0.002753	1.97	0.141266	0.002687	1.97	0.141266	0.002689	1.97
513	0.139943	0.001387	1.99	0.139911	0.001355	1.98	0.139911	0.001355	1.98
1025	0.139248	0.000695	1.99	0.139232	0.000679	1.99	0.139232	0.000680	1.99

TABLE 2.6: *Convergence results for a supershare binary call option evaluated at $S = K = 10$ using various timestepping methods. Data as in Table 2.5. The exact value is 0.1385509. Difference is the absolute value of the change in the solution as the grid is refined. Ratio is the ratio of successive differences.*

Results for implicit, Crank-Nicolson and Rannacher timestepping without any data smoothing are given in Table 2.6. Again, the initial discontinuities have reduced the convergence of both Crank-Nicolson and Rannacher timestepping to first order. Further, all methods exhibit relatively poor accuracy. Nevertheless, quadratic convergence is once again restored for Rannacher timestepping by smoothing the initial data, as shown by Table 2.7. Moreover, accuracy is greatly improved, even on the original grid.

	Averaging			Shifting Mesh			Projection		
Nodes	Value	Difference	Ratio	Value	Difference	Ratio	Value	Difference	Ratio
65	0.1383331			0.1382546			0.1384516		
129	0.1384979	0.0001648		0.1384731	0.0002185		0.1385275	0.0000759	
257	0.1385377	0.0000398	4.14	0.1385310	0.0000578	3.38	0.1385452	0.0000177	4.30
513	0.1385476	0.0000099	4.03	0.1385459	0.0000149	3.88	0.1385495	0.0000043	4.10
1025	0.1385501	0.0000025	4.00	0.1385496	0.0000038	3.93	0.1385505	0.0000011	3.98

TABLE 2.7: *Convergence results for a supershare binary call option evaluated at $S = K = 10$ using Rannacher timestepping, combined with various methods for smoothing the payoff. Data as in Table 2.5. The exact value is 0.1385509.*

2.4.3 Multi-factor Extensions

Each of the three methods for smoothing initial discontinuities can be extended to higher dimensional problems with varying degrees of ease and success. Depending on the payoff condition, it may or may not be possible to place the strike price halfway between nodes in higher dimensions. For example, this smoothing method would be impossible for curved or angled payoff discontinuities when using a structured grid. In theory, both the averaging method and the projection method generalize to higher dimensions. In practice, it may be difficult to accurately integrate the known payoff function if discontinuities are not aligned with element boundaries. This is particularly true for the averaging method, where (technically) the integrations should be done over non-overlapping control volumes surrounding each node.

For testing, we have used the projection method for a simple two-factor digital call option. The pricing equation in this case is

$$\frac{\partial U}{\partial t} + \frac{1}{2}\sigma_1^2 S_1^2 \frac{\partial^2 U}{\partial S_1^2} + rS_1 \frac{\partial U}{\partial S_1} + \rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 U}{\partial S_1 \partial S_1} + \frac{1}{2}\sigma_2^2 S_2^2 \frac{\partial^2 U}{\partial S_2^2} + rS_2 \frac{\partial U}{\partial S_2} - rU = 0. \quad (2.26)$$

The discretization given by equation (2.16) is used. The parameters are the same as in Table 2.1, with the additions of $\sigma_1 = \sigma_2 = 0.3$ and an asset correlation of $\rho = 0.3$. Further, the payoff condition is taken to be

$$V(S_1, S_2, t = T) = \begin{cases} 1.0 & \text{if } \max(S_1, S_2) \geq K \\ 0.0 & \text{otherwise.} \end{cases} \quad (2.27)$$

The effect of the L_2 projection is shown on the left side of Figure 2.4 for the original grid. Subsequent grids were created by splitting each triangle into four new triangles by adding new nodes at all edge midpoints. With each grid refinement, the initial timestep of 0.02

Nodes	Value	Difference	Ratio
41×41	0.6893597		
81×81	0.6889065	0.0004532	
161×161	0.6887937	0.0001128	4.02
321×321	0.6887654	0.0000283	3.99
641×641	0.6887583	0.0000071	4.00

TABLE 2.8: *Convergence results for an at-the-money ($S_1 = S_2 = 40$) two-factor digital call option using Rannacher timestepping and projected initial conditions. The exact value is 0.6887560.*

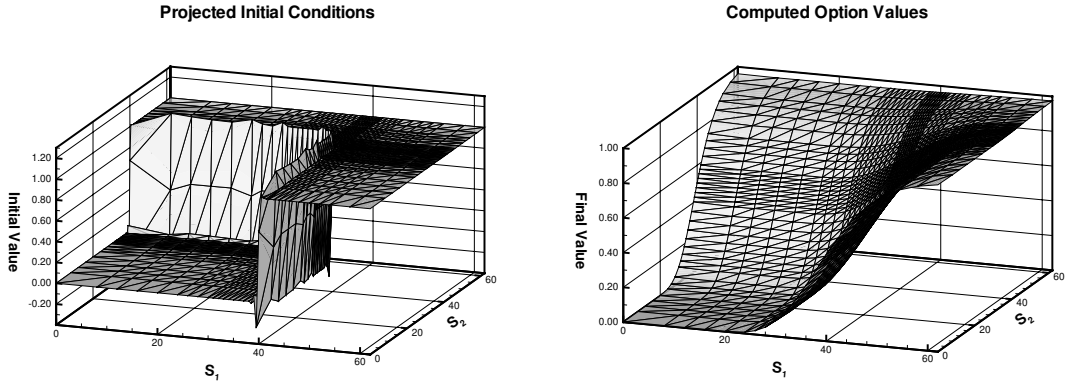


FIGURE 2.4: *Projected initial conditions for a two-dimensional mesh, and the resulting solution at $\tau = T$.*

was also cut in half. Despite the rather oscillatory nature of the projected initial conditions, Table 2.8 confirms that the numerical solution obtained by combining the projected initial data with Rannacher timestepping does in fact exhibit quadratic convergence. A plot of the solution surface is shown on the right of Figure 2.4.

2.5 Iterative Methods for Nonlinear Systems of Equations

When discretizing a nonlinear PDE, the resulting set of discrete algebraic equations is also nonlinear. Solving these discrete equations is usually done by an iterative approach. Several iterative methods will be studied below. For all cases, the basic scheme is summarized in

Let $(U^{n+1})^0 = U^n$
for $k = 0, 1, 2, \ldots$ until convergence **do**
  Solve a linear system for an update vector: ΔU^k
  Update the solution: $(U^{n+1})^{k+1} = (U^{n+1})^k + \Delta U^k$
  if $\max_i \frac{|(U_i^{n+1})^{k+1} - (U_i^{n+1})^k|}{\max(scale, |(U_i^{n+1})^{k+1}|)} < tolerance$ **then**
    Quit the iterations
  end if
end for

ALGORITHM 2.1: *Basic template for all iterative methods to be tested.*

Algorithm 2.1. The scale factor *scale* in the denominator of the tolerance check should be of the same magnitude as the expected value of the option. For all problems studied in this thesis, options are assumed to be priced in dollars, so *scale* has been set to 1.0.

The iterative schemes outlined below differ in which linear system is solved to determine the update vector from iteration k to $k+1$. The schemes to be tested include: Newton's (numerical Jacobian) method, non-smooth Newton's method, and a frozen coefficient method. To avoid algebraic complication, the iterative methods will be illustrated using the one factor discrete equations (2.11). The reader should have no difficulty generalizing these algorithms to the two factor equations. For all methods, let $\bar{U} = U^{n+1}$ be the solution at time $n+1$, and let $\bar{U}^{k+1} = (U^{n+1})^{k+1}$ be the estimation from iteration k to $k+1$ for the solution at time $n+1$.

Newton (numerical Jacobian) In a full Newton iteration, the Jacobian is constructed and used to compute the update. Specifically, let $\mathbf{F}(\bar{U}^{k+1})$ represent the system of discrete equations to be solved at iteration k (refer to equation (2.11)):

$$\begin{aligned}\mathbf{F}(\bar{U}^{k+1}) = -\bar{U}_i^{k+1} + U_i^n + (1-\theta)\Big[&(-(\alpha_i^{n+1})^{k+1} - (\beta_i^{n+1})^{k+1} - r\Delta\tau)\bar{U}_i^{k+1} \\ &+ (\alpha_i^{n+1})^{k+1}\bar{U}_{i-1}^{k+1} + (\beta_i^{n+1})^{k+1}\bar{U}_{i+1}^{k+1}\Big] \\ &+ \theta\left[(-\alpha_i^n - \beta_i^n - r\Delta\tau)U_i^n + \alpha_i^n U_{i-1}^n + \beta_i^n U_{i+1}^n\right].\end{aligned} \tag{2.28}$$

Further, let $\mathbf{J}(\bar{U}^{k+1})$ be the associated Jacobian of $\mathbf{F}(\bar{U}^{k+1})$. The update vector is then found by the solution to

$$\mathbf{J}(\bar{U}^{k+1})(\Delta\bar{U}^{k}) = -\mathbf{F}(\bar{U}^{k+1}). \tag{2.29}$$

A straightforward approach to approximating $\mathbf{J}(\bar{U}^{k+1})$ in equation (2.29) is to use numerical differentiation (forward differencing has been used in this thesis). Note that when a variable is shifted, the α and β coefficients in equation (2.28) need to be recomputed. If the equations are non-smooth due to constraints on the nonlinear coefficients (as is the case for uncertain volatility and passport options), this straightforward approach can be expected to have problems.

Non-smooth Newton Non-smooth Newton methods use a generalized Jacobian in the Newton iteration [44]. An example of a particular element of the generalized derivative of a non-smooth function is given by the following. Consider the function $f(x) = \min(x, 1)$, which is non-smooth at $x = 1$. For $x \neq 1$, use the limiting values

$$f'(x) = \begin{cases} 1 & \text{if } x < 1 \\ \\ 0 & \text{if } x > 1. \end{cases} \tag{2.30}$$

At $x = 1$, one of these values can be used arbitrarily, such as

$$f'(x) = 1 \quad \text{if } x = 1. \tag{2.31}$$

In the context of uncertain volatility and passport options, the Jacobian is constructed numerically, except that special care is taken near the max/min values of the non-smooth coefficients to ensure that the correct limit is obtained. This technique is commonly used in computational fluid dynamics (CFD) when applying upwind weighting

schemes [13].

Frozen Coefficients The last method tested can be structurally identical to Newton's method, but the values for q in equations (2.7) and (2.8) are frozen using the $\bar{U}_i^k = (U_i^{n+1})^k$ data. The discrete equations therefore have the form (k is the iteration index)

$$\begin{aligned}\mathbf{F}(\bar{U}^{k+1}) = &-\bar{U}_i^{k+1} + U_i^n \\ &+ (1-\theta)\left[(-(\alpha_i^{n+1})^k - (\beta_i^{n+1})^k - r\Delta\tau)\bar{U}_i^{k+1} + (\alpha_i^{n+1})^k\bar{U}_{i-1}^{k+1} + (\beta_i^{n+1})^k\bar{U}_{i+1}^{k+1}\right] \\ &+ \theta\left[(-\alpha_i^n - \beta_i^n - r\Delta\tau)U_i^n + \alpha_i^n U_{i-1}^n + \beta_i^n U_{i+1}^n\right]. \end{aligned} \tag{2.32}$$

Alternatively, one can say that the α and β coefficient values are not recomputed when the linearized equations are being constructed. Of course, this means that the linearized equations do not use the true Jacobian, and this method reverts to what is often called a fixed point iteration. This method is also commonly used in CFD applications [13].

For many problems, the nonlinear coefficients always obtain maximal or minimal values (bang-bang problems). For example, this occurs for one factor uncertain volatility and passport options when the payoff is convex. However, during the iterative process, an update may temporarily turn the convex data concave, especially if the original data is close to linear. In these cases, it may be more appropriate to recompute coefficients, instead of relying on a theoretical analysis which forces them to extreme values (this issue will be discussed in detail in chapter 5).

An alternative approach to the nonlinear iteration methods mentioned above is to eval-

uate the nonlinear coefficients explicitly. In this case we solve

$$U_i^{n+1} - U_i^n = (1-\theta)\left[(-\alpha_i^n - \beta_i^n - r\Delta\tau)U_i^{n+1} + \alpha_i^n U_{i-1}^{n+1} + \beta_i^n U_{i+1}^{n+1}\right] + \theta\left[(-\alpha_i^n - \beta_i^n - r\Delta\tau)U_i^n + \alpha_i^n U_{i-1}^n + \beta_i^n U_{i+1}^n\right]. \quad (2.33)$$

An explicit evaluation of the coefficients is equivalent to forcing only one nonlinear iteration per timestep for the frozen coefficient method. This method is simple to implement, but can give at most linear convergence as the timestep is reduced. As will be seen in chapter 3, in some cases it leads to sub-linear convergence.

2.6 Convergence of the Frozen Coefficient Iteration

In the previous section, several iterative methods for solving the nonlinear discrete algebraic equations that arise at each timestep of the numerical solution process were listed. In this section, the convergence of the frozen coefficient method (fixed point iteration method) is studied in detail. The analysis will be done for one factor problems of the form (2.7) or (2.8). In other words, one factor nonlinear PDEs in which the coefficients depend on the solution or its derivatives and need to be optimized at each timestep, are examined. This type of PDE arises in the one factor uncertain volatility model, and also for passport options. The results of the analysis in this section will be used in chapters 3 and 5 (with different amounts of success).

For notational convenience, it helps to rewrite the discrete equations (2.11) in matrix form. Let $U^{n+1} = [U_1^{n+1}, U_2^{n+1}, \ldots, U_{imax}^{n+1}]'$, $U^n = [U_1^n, U_2^n, \ldots, U_{imax}^n]'$ and

$$\left[\hat{M}^n U^n\right]_i = -\left[(-\alpha_i^n - \beta_i^n - r\Delta\tau)U_i^n + \alpha_i^n U_{i-1}^n + \beta_i^n U_{i+1}^n\right]. \quad (2.34)$$

The first and last rows of $\hat{M}$ are modified as needed to handle the boundary conditions. In

the examples to be tested, the boundary conditions are of Dirichlet type. The boundary condition at $S = 0$ is enforced by setting $\alpha_i = \beta_i = 0$ at $i = 1$. We approximate the infinite computational domain $S \in [0, \infty]$ by the finite domain $S \in [0, S_{max}]$. Denote the node corresponding to $S_i = S_{max}$ as $S_i = S_{imax}$. To avoid algebraic complication in the following, we will assume that the Dirichlet condition at $S = S_{imax}$ is time independent, so that

$$U(\tau, S_{max}) = U_{imax} = \ Const.$$

The boundary condition at $i = imax$ is enforced by setting $U^0_{imax} = U_{imax}$, and setting the last row of $\hat{M}$ to be identically zero. With a slight abuse of notation, we denote this last row as $(\hat{M})_{imax} \equiv 0$. In the following, it will be understood that equations of type (2.34) hold only for $i < imax$, with $(\hat{M})_{imax} \equiv 0$.

The discrete equations (2.11) can then be written in a compact matrix form:

$$\left[I + (1-\theta)\hat{M}^{n+1}\right] U^{n+1} = \left[I - \theta\hat{M}^{n}\right] U^{n}. \tag{2.35}$$

By applying forward and backward differencing as needed to ensure that α^n_i and β^n_i are positive, matrix $\left[I + (1-\theta)\hat{M}^{n+1}\right]$ is an M-matrix—a diagonally dominant matrix with positive diagonals and non-positive off-diagonals. It is known that all of the elements of the inverse of an M-matrix are non-negative.

Let $(U^{n+1})^k$ be the k^{th} estimate for U^{n+1}. The frozen coefficient method can then be written as in Algorithm 2.2. For further notational convenience, define

$$\hat{M}^k \equiv \hat{M}((U^{n+1})^k)$$

Let $(U^{n+1})^0 = U^n$
for $k = 0, 1, 2, \ldots$ until convergence **do**
  Solve $\left[I + (1-\theta)\hat{M}((U^{n+1})^k)\right](U^{n+1})^{k+1} = \left[I - \theta\hat{M}(U^n)\right]U^n$
  if $\max_i \frac{|(U_i^{n+1})^{k+1} - (U_i^{n+1})^k|}{\max(scale, |(U_i^{n+1})^{k+1}|)} < tolerance$ **then**
    Quit the iterations
  end if
end for

ALGORITHM 2.2: *Pseudo-code for the frozen coefficient method.*

and as before define

$$\bar{U}^k \equiv (U^{n+1})^k.$$

The key update in Algorithm 2.2 can then be written as

$$\left[I + (1-\theta)\hat{M}^k\right]\bar{U}^{k+1} = \left[I - \theta\hat{M}^n\right]U^n. \tag{2.36}$$

Before proving the convergence of the iterative method, we begin with a lemma. The lemma will be used in the proof of convergence below.

Lemma 2.1 (Single signed update during the iteration). *Let k be the iteration index, and assume that the discretization scheme does not change between iteration k and $k+1$ (i.e. central, forward or backward discretizations are chosen at each node independent of the iteration level k). Then for $k \geq 1$, the expression*

$$\left[(-\alpha_i^k + \alpha_i^{k-1} - \beta_i^k + \beta_i^{k-1})U_i^k + (\alpha_i^k - \alpha_i^{k-1})U_{i-1}^k + (\beta_i^k - \beta_i^{k-1})U_{i+1}^k\right] \tag{2.37}$$

is non-negative when solving equation (2.7) *and is non-positive when solving equation* (2.8)*.*

Proof. The values of α_i^k and β_i^k depend upon the value of $q_i^k = q_i^k(U^k, (U_S)_i^k, (U_{SS})_i^k)$. Otherwise, the equations would be linear and expression (2.37) is identically zero. When solving equation (2.7), the value of q_i^k is calculated such that the discrete equation (2.11)

is maximized. Hence, the term

$$(-\alpha_i^k - \beta_i^k)U_i^k + \alpha_i^k U_{i-1}^k + \beta_i^k U_{i+1}^k \tag{2.38}$$

obtains its maximum value for all possible α_i and β_i (as a function of q). Conversely, the term

$$(-\alpha_i^{k-1} - \beta_i^{k-1})U_i^k + \alpha_i^{k-1} U_{i-1}^k + \beta_i^{k-1} U_{i+1}^k \tag{2.39}$$

is less than or equal to the maximum since q_i^{k-1} is used to determine α_i^{k-1} and β_i^{k-1}, but the solution data is at iteration k. For this to be true in general, the discrete scheme used at iteration k must be the same as the scheme used at iteration $k-1$ (e.g. cannot switch from central differencing to forward differencing). Thus, expression (2.37), which is equation (2.38) minus equation (2.39), is non-negative.

On the other hand, when solving equation (2.8), the term

$$(-\alpha_i^k - \beta_i^k)U_i^k + \alpha_i^k U_{i-1}^k + \beta_i^k U_{i+1}^k \tag{2.40}$$

obtains its minimum value. Further, the term

$$(-\alpha_i^{k-1} - \beta_i^{k-1})U_i^k + \alpha_i^{k-1} U_{i-1}^k + \beta_i^{k-1} U_{i+1}^k \tag{2.41}$$

is larger than or equal to the minimum value, since the q_i and α_i, β_i values are at different iteration levels. Hence, expression (2.37) is non-positive. □

Our main result in this section can now be summarized in the following theorem:

Theorem 2.1 (Convergence of the nonlinear iterations). *If the matrix* $\left[I + (1-\theta)\hat{M}^k\right]$ *in equation* (2.36) *is an M-matrix, and if the discretization scheme does not change between iterations* k *and* $k+1$*, then the nonlinear iteration in Algorithm 2.2*

converges to the unique solution to equation (2.35)*, given any initial iterate* $\bar{U}^0$*. Moreover, the iterates converge monotonically.*

Proof. We will first prove that this algorithm is globally convergent for equation (2.7) by showing that the iterates form non-decreasing sequences which are bounded from above. The proof for equation (2.8) is qualitatively identical, except that a non-increasing sequence bounded from below is obtained.

Writing equation (2.36) for iteration k gives

$$\left[I + (1-\theta)\hat{M}^{k-1}\right]\bar{U}^k = \left[I - \theta\hat{M}^n\right]U^n,$$

which can also be expressed as

$$\left[I + (1-\theta)\hat{M}^{k}\right]\bar{U}^k + (1-\theta)\left[\hat{M}^{k-1} - \hat{M}^k\right]\bar{U}^k = \left[I - \theta\hat{M}^n\right]U^n. \tag{2.42}$$

Subtracting equation (2.42) from equation (2.36) gives

$$\left[I + (1-\theta)\hat{M}^{k}\right](\bar{U}^{k+1} - \bar{U}^k) = (1-\theta)\left[\hat{M}^{k-1} - \hat{M}^k\right]\bar{U}^k. \tag{2.43}$$

We wish to show that the iterates form a bounded non-decreasing sequence. Expanding the right hand side of equation (2.43) using definition (2.34) for node i leads to

$$\begin{aligned}(1-\theta)\left(\left[\hat{M}^{k-1} - \hat{M}^k\right]\bar{U}^k\right)_i = \\ (1-\theta)\left[(-\alpha_i^k + \alpha_i^{k-1} - \beta_i^k + \beta_i^{k-1})U_i^k + (\alpha_i^k - \alpha_i^{k-1})U_{i-1}^k + (\beta_i^k - \beta_i^{k-1})U_{i+1}^k\right].\end{aligned} \tag{2.44}$$

By the results of Lemma (2.1), the right-hand side of equation (2.44) must be non-negative. Then, by expression (2.43), and the fact that $\left[I + (1-\theta)\hat{M}^k\right]$ is an M-matrix, it

follows that

$$\bar{U}^{k+1} - \bar{U}^k \geq 0 \; ; \quad k \geq 1, \tag{2.45}$$

and hence the iterates form a non-decreasing sequence for all iterations *after* the first iteration ($k \geq 1$) at each timestep.

Now that the iterates have been shown to be non-decreasing (equation (2.7)) or non-increasing (equation (2.8)), we need to show that they are bounded. To do this, let $b = [I - \theta\hat{M}^n]U^n$. For a fixed mesh, $\|b\|_\infty$ is bounded. By equations (2.34) and (2.36), we have

$$\begin{aligned} \left[I + (1-\theta)\hat{M}^k\right]\bar{U}^{k+1} &= b \\ \left[1 + (1-\theta)(\alpha_i^k + \beta_i^k + r\Delta\tau)\right]\bar{U}_i^{k+1} &= (1-\theta)\alpha_i^n\bar{U}_{i-1}^{k+1} + (1-\theta)\beta_i^k\bar{U}_{i+1}^{k+1} + b_i. \end{aligned} \tag{2.46}$$

Now let $U_{max} = \max_i(\bar{U}_i^{k+1}), U_{min} = \min_i(\bar{U}_i^{k+1}), b_{max} = \max_i(b_i)$ and $b_{min} = \min_i(b_i)$. Note that $(\hat{M}^k)_{imax} \equiv 0$ and that $b_{imax} = U_{imax}$. Then since all coefficients of the U terms are positive (recall the M-matrix properties), we have

$$\begin{aligned} \left[1 + (1-\theta)(\alpha_i^k + \beta_i^k + r\Delta\tau)\right]\bar{U}_i^{k+1} &\leq (1-\theta)\alpha_i^n U_{max} + (1-\theta)\beta_i^k U_{max} + b_{max} \quad ; \quad i < imax \\ U_{max} &\leq \max\left(\frac{b_{max}}{1 + (1-\theta)r\Delta\tau}, b_{max}\right). \end{aligned} \tag{2.47}$$

Similarly,

$$U_{min} \geq \min\left(\frac{b_{min}}{1 + (1-\theta)r\Delta\tau}, b_{min}\right). \tag{2.48}$$

Thus $\|U_i^{k+1}\|_\infty \leq \|b\|_\infty$, independent of k. Consequently, since the iterates are either non-increasing or non-decreasing, and $\|U_i^{k+1}\|_\infty$ is bounded independent of k, the iteration (2.36) converges. Note that if $\bar{U}^{k+1} = \bar{U}^k$, then the residual of the nonlinear equations is identically zero, and $\bar{U}^{k+1}$ is an exact solution of equation (2.35).

As for uniqueness, again assume that equation (2.7) is being solved. Further assume that two solutions to equation (2.36), U_1 and U_2, exist such that

$$\left[I + (1-\theta)\hat{M}_1\right] U_1 = \left[I - \theta\hat{M}^n\right] U^n \tag{2.49}$$

$$\left[I + (1-\theta)\hat{M}_2\right] U_2 = \left[I - \theta\hat{M}^n\right] U^n, \tag{2.50}$$

where $\hat{M}_1 = \hat{M}(U_1), \hat{M}_2 = \hat{M}(U_2)$. Equation (2.49) can then be rewritten as

$$\left[I + (1-\theta)\hat{M}_2\right] U_1 + (1-\theta)\left[\hat{M}_1 - \hat{M}_2\right] U_1 = \left[I - \theta\hat{M}^n\right] U^n. \tag{2.51}$$

Subtracting equation (2.50) from equation (2.51) gives

$$\left[I + (1-\theta)\hat{M}_2\right] (U_1 - U_2) = (1-\theta)\left[\hat{M}_2 - \hat{M}_1\right] U_1. \tag{2.52}$$

In component form, an expression equivalent to expression (2.37) is recovered. From Lemma (2.1), we know that the right hand side of equation (2.52) is always non-negative. It follows that $U_1 \geq U_2$. Interchanging subscripts gives $U_2 \geq U_1$, and hence $U_1 = U_2$. Similar arguments can be used for equation (2.8)

□

2.7 Convergence to the Viscosity Solution

When numerically solving a nonlinear PDE, more than one solution is possible. For the problems studied in this thesis, convergence to the financially relevant solution is sought. The financially relevant solution can be defined in terms of a no-arbitrage analysis. For the regular Black-Scholes equation, a hedging portfolio can be constructed that perfectly replicates the option payoff. For the nonlinear problems to be studied in later chapters,

such a portfolio is impossible to obtain. However, a dominating strategy which guarantees a non-negative balance in the hedging account can be obtained [6]. Thus, convergence to the solution which determines this dominating or no arbitrage strategy is desired.

To obtain this convergence, note that the equations and arguments that lead to the no-arbitrage equations can also be interpreted in a stochastic control context. For example, in equations (2.7) and (2.8), q is the control variable, U is the value function, and the PDEs are Hamilton-Jacobi-Bellman (HJB) equations that result from applying Bellman's principal of dynamic programming. In this stochastic control context, the correct value function is recovered by finding the viscosity solution of the HJB equation [22]. Hence, the viscosity solution is equivalent to the no-arbitrage solution in a financial context.

The concept of a viscosity solution is closely related to the entropy condition (E-condition) of weak solutions to conservation law problems [35]. First introduced in [16], a full description of viscosity solutions is given in [15]. A brief overview of the relevant theory required by this thesis is given below.

To understand the notion of a viscosity solution, consider the general form of a second order parabolic PDE:

$$\frac{\partial V}{\partial t} + F(x, V, V_x, V_{xx}) = 0. \tag{2.53}$$

We assume that F satisfies the ellipticity condition

$$F(x, V, V_x, V_{xx} + \epsilon) \leq F(x, V, V_x, V_{xx}) \text{ if } \epsilon \geq 0. \tag{2.54}$$

Note the sign convention for F as given in equation (2.53), which is standard in the literature. This ellipticity property is crucial for the definition of viscosity solutions.

To motivate the definition, consider functions $v(t, x) \in C^2$ and $V(t, x) \in C^2$. Let (t_0, x_0) be a local maximum of $V - v$. From basic calculus, we know that $\partial V/\partial t = \partial v/\partial t, V_x = v_x$,

and $v_{xx} \geq V_{xx}$ near (t_0, x_0). Using these relations and the ellipticity property (2.54) gives

$$\frac{\partial v}{\partial t}(t_0, x_0) + F(x_0, V(t_0, x_0), v_x(t_0, x_0), v_{xx}(t_0, x_0)) \leq 0. \tag{2.55}$$

If (2.55) holds for all $v \in C^2$ for which (t_0, x_0) is a local maximum of $V - v$, then V is said to be a *viscosity subsolution* of equation (2.53). In a sense, the functions v have been used to provide an upper bound to possible solutions. Similarly, $V(t, x)$ is a *viscosity supersolution* of equation (2.53) if $\forall v \in C^2$, if (t_0, x_0) is a local minimum point of $V - v$, then

$$\frac{\partial v}{\partial t}(t_0, x_0) + F(x_0, V(t_0, x_0), v_x(t_0, x_0), v_{xx}(t_0, x_0)) \geq 0. \tag{2.56}$$

A solution that is both a viscosity subsolution and a viscosity supersolution is a viscosity solution of (2.53). In essence, the definition sandwiches the desired viscosity solution between smooth solutions that are "above the PDE" or "below the PDE", in the sense of definitions (2.55)-(2.56). Note that a classical solution to equation (2.53) is also a viscosity solution, as verified by letting $v = V$.

However, definitions (2.55)-(2.56) can still be used even when V is not smooth, since existence of the first and second derivatives of V are not required. In this way, non-smooth solutions to (2.53) can be defined. The existence and uniqueness of viscosity solutions has also been established [22], which contributes to their practical importance.

For most real world problems, analytic viscosity solutions are impossible to find. Numerical schemes for generating viscosity solutions are therefore required. This topic was discussed in [8]. In particular, conditions for numerical schemes which guarantee convergence to the viscosity solution were developed. Since this result plays a pivotal role for many results later in this thesis, it is briefly summarized below. For a full description of the terminology and further technical details, the reader is referred to [8] and the references therein.

Let a discrete scheme that approximates equation (2.53) be defined as

$$S(n, \Delta t, h, x, V^n(x), V^n) = 0 \text{ on } \bar{\Omega}, \tag{2.57}$$

where n is the time level, h is a grid spacing parameter, $V^n(x)$ is the solution at point x and time $t = n\Delta t$, V^n denotes all points on which the solution $V^n(x)$ depends, and $\bar{\Omega}$ is the discrete domain (including the boundary points). Note that the function S is defined point-wise for a discrete grid.

Assume that the discrete scheme satisfies the following conditions.

Stability For any $n > 0$, the scheme has a solution V^n. Moreover, V^n is uniformly bounded, which means there exists a constant C such that

$$-C \leq V^n \leq C \text{ on } \bar{\Omega}$$

for any $n > 0$.

Consistency For any smooth function ϕ, one has

$$\lim_{\substack{h \to 0 \\ \Delta t \to 0 \\ n\Delta t \to t}} S(n, \Delta t, h, x, \phi(x), \phi) \to \phi_t + F(x, \phi(x), D\phi(x), D^2\phi(x))$$

Note that this is a simpler consistency condition than what appeared in [8], which used a more complex statement to handle unusual boundary conditions. The more complex definition is not required in this thesis.

Monotonicity A simple definition is

$$S(n, \Delta t, h, x, w, u) \leq S(n, \Delta t, h, x, w, v) \text{ if } u \geq v$$

for any $n > 0$, $x \in \bar{\Omega}$. Since showing monotonicity is the most difficult property to verify for numerical schemes, it is discussed in detail in chapter 3 for one factor uncertain volatility problems. In particular, a more precise discrete definition of monotonicity for the actual discretization scheme studied in this thesis is provided.

Note that the concept of ellipticity for equation (2.53) is replaced by monotonicity for discrete equations.

Strong Comparison Result If u is an upper semicontinuous subsolution of equation (2.53) and if v is a lower semicontinuous supersolution of equation (2.53), then

$$u \leq v.$$

The strong comparison property has been proven for first order equations for "all kinds of classical equations and boundary conditions" [8]. It has also been proven for second order equations with Neumann boundary conditions and classical Dirichlet boundary conditions (only fully nonlinear degenerate equations are not well understood). As such, the strong comparison result automatically holds for all problems studied in this thesis, and thus will not be discussed in later chapters.

The main result of [8] can then be written as

Theorem 2.2 (Result of [8], page 12). *Under the above assumptions, the solution V^n of the discrete scheme converges uniformly on each compact subset of $\bar{\Omega}$ to the unique viscosity solution of the equation.*

Again, since all problems to be studied in later chapters automatically satisfy the strong comparison result, we will be concerned with showing consistency, stability and monotonicity of the numerical schemes.

Chapter 3

One Factor Uncertain Volatility

The first study of a particular nonlinear pricing PDE in this thesis is for the one factor uncertain volatility model. Due to the mild form of the nonlinearity, a detailed analysis of the discrete equations can be performed. The main theoretical result concerns a proof of the conditions for guaranteed convergence to the viscosity solution. The numerical examples also establish baseline convergence rates which can help guide the more difficult two factor uncertain volatility (chapter 4) and passport option (chapter 5) experiments.

3.1 Background

For options based on a single stock value S, the stochastic process implied by equation (2.1) is often simplified to

$$dS = \mu S dt + \sigma(S,t) S dW \tag{3.1}$$

where μ is the expected drift rate and $\sigma(S,t)$ is the stock volatility. Several models for volatility have been proposed in the option pricing literature. The simplest model assumes constant volatility. This was the approach taken by [10] and [39] in the original option pricing research, and is still the industry standard. However, it is generally agreed that constant

volatility cannot explain observed market prices for options. More complicated models assume volatility surfaces across underlying asset prices and time (see [4, 14] and references therein). These surfaces are often constructed by the implied volatilities under the Black-Scholes model for a variety of currently traded contracts. This means that $\sigma^{imp}(S,t)$ and/or $\sigma(S,t)$ are determined by matching the pricing PDE to observed market prices. A third modeling approach uses stochastic volatility, for which the volatility is assumed to follow a random process [29]. A downside of stochastic volatility for numerical pricing methods is an increase in the number of state variables (factors) that need to be considered. Next, while not modeling volatility directly, jump diffusion attempts to model the random behaviour of the underlying asset. Recent research tends to support a jump diffusion framework (see [19] and the references therein), especially for short-lived options. However, care must be taken when solving the resulting equations to ensure reasonable performance and accuracy [19].

Another approach, and the one that will be studied in this chapter, is uncertain volatility. The uncertain volatility model was independently developed in [6] and [36]. Following standard arguments, the PDE for the fair price of a contingent claim on one asset in the Black-Scholes model with uncertain volatility is given by

$$U_\tau = \frac{\sigma(\Gamma)^2}{2} S^2 U_{SS} + rSU_S - rU, \tag{3.2}$$

where S represents the underlying asset price, T is the maturity time of the option, $\tau = T - t$ is time in the backwards direction, $\Gamma = U_{SS}$, $\sigma(\Gamma)$ is the uncertain volatility (more details below), and r is the risk-free interest rate.

The volatility is assumed to lie within the range

$$\sigma_{\min} \leq \sigma(\Gamma) \leq \sigma_{\max}.$$

In the following, it is assumed that $\sigma_{\min} > 0$, $r > 0$. With a range of possible volatility values, equation (3.2) is nonlinear and does not possess a unique solution. Nevertheless, best-case and worst-case values exist, and are unique [56]. These values are found by either maximizing or minimizing the diffusion term by selecting σ according to the value of $\Gamma = \partial^2 U/\partial S^2$. Hence, $\sigma(\Gamma)$ is written in equation (3.2) to denote the explicit dependence of volatility on the value of gamma. Specifically, if the worst-case for an investor with a long position in the option is considered, then

$$\sigma(\Gamma)^2 = \begin{cases} \sigma_{max}^2 & \text{if } \Gamma \leq 0 \\ \sigma_{min}^2 & \text{if } \Gamma > 0. \end{cases} \tag{3.3}$$

On the other hand, the best-case for an investor with a long position is determined by

$$\sigma(\Gamma)^2 = \begin{cases} \sigma_{max}^2 & \text{if } \Gamma > 0 \\ \sigma_{min}^2 & \text{if } \Gamma \leq 0. \end{cases} \tag{3.4}$$

Prices for investors with short positions are given by the negative of the solutions when applying equations (3.3) and (3.4). By assuming the worst-case, the seller of an option can hedge his/her position and obtain a non-negative balance in the hedging portfolio, regardless of the actual volatility movement (provided that the volatility remains within the specified range) [6]. Note that the Leland model of transaction costs [34] can be formulated as a nonlinear PDE which is mathematically identical to equation (3.2) with nonlinearities of the form (3.3)-(3.4) [59].

For simple options with convex payoffs, the solution of equation (3.2) reduces to that of a constant volatility problem with one of the extreme volatility values. Most authors of previous studies (e.g. [36, 6, 21, 37, 24]) therefore choose to study more exotic options with non-convex payoff functions. Barrier options seem to be the most popular. The nonlinearity

of the problem also makes portfolio evaluation more difficult [5]. Most previous studies have focused on theoretical properties of the model or on model extensions. The results of this thesis will be primarily concerned with the numerical convergence of the pricing equation, and in developing robust solution techniques for any payoff structure.

3.2 Discretization

For one factor uncertain volatility problems, the discrete equations have the form of equation (2.11). Nevertheless, the exact form of the α and β coefficients are provided in order to show that the conditions for Theorem 2.1 can be satisfied, and for proving convergence to the viscosity solution.

Discretizing the first derivative term of equation (3.2) with central differences leads to

$$\begin{aligned}
\alpha^n_{i,central} &= \left[\frac{\sigma(\Gamma^n_i)^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})} - \frac{rS_i}{S_{i+1} - S_{i-1}}\right]\Delta\tau \\
\beta^n_{i,central} &= \left[\frac{\sigma(\Gamma^n_i)^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})} + \frac{rS_i}{S_{i+1} - S_{i-1}}\right]\Delta\tau.
\end{aligned} \tag{3.5}$$

If $\alpha_{i,central}$ is negative, oscillations may appear in the solution ($\beta_{i,central}$ is always positive). The oscillations can be avoided by using forward differences at the problem nodes, leading to

$$\begin{aligned}
\alpha^n_{i,forward} &= \frac{\sigma(\Gamma^n_i)^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})}\Delta\tau \\
\beta^n_{i,forward} &= \left[\frac{\sigma(\Gamma^n_i)^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})} + \frac{rS_i}{S_{i+1} - S_i}\right]\Delta\tau.
\end{aligned} \tag{3.6}$$

The choice to use a central or forward discretization at a given node for equation (2.11) is made by Algorithm 3.1. Note that the use of $\sigma_{\min}$ in the test condition guarantees that α_i and β_i are always positive, regardless of the choice of $\sigma(\Gamma)$. Hence, the discretization

if $\left[\frac{\sigma_{min}^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})} - \frac{rS_i}{S_{i+1} - S_{i-1}}\right] \geq 0$ **then**
  $\alpha_i = \alpha_{i,central}$
  $\beta_i = \beta_{i,central}$
else
  $\alpha_i = \alpha_{i,forward}$
  $\beta_i = \beta_{i,forward}$
end if

ALGORITHM 3.1: *Deciding between a central or forward discretization for one factor uncertain volatility problems.*

scheme is fixed at each node, and the conditions in Theorem 2.1 are satisfied.

At $S = 0$, we have the boundary condition

$$U_\tau = -rU, \tag{3.7}$$

while as $S \to \infty$, we have the Dirichlet condition

$$U \simeq A(\tau)S + B(\tau), \tag{3.8}$$

where A and B can be determined by financial reasoning. In practice, a finite computational domain is used so that condition (3.8) is applied at a finite value S_{max}.

The set of algebraic equations (2.11) is non-smooth due to the form of equations (3.3)-(3.4). The non-smoothness can be made clear by re-writing the discrete equations at each node as

$$\begin{aligned} g_i = &-U_i^{n+1} + U_i^n + (1-\theta)\left[\sum_{j\in\eta_i} \Delta\tau\gamma_{ij}(U_j^{n+1} - U_i^{n+1}) - r\Delta\tau U_i^{n+1}\right] \\ &+ \theta\left[\sum_{j\in\eta_i} \Delta\tau\gamma_{ij}(U_j^{n} - U_i^{n}) - r\Delta\tau U_i^{n}\right] + (1-\theta)\Delta\tau\frac{\sigma(\Gamma_i^{n+1})^2 S_i^2}{2}\Gamma_i^{n+1} \\ &+ \theta\Delta\tau\frac{\sigma(\Gamma_i^{n})^2 S_i^2}{2}\Gamma_i^{n} = 0, \end{aligned} \tag{3.9}$$

where

$$\gamma_{ij} = \begin{cases} rS_i/(S_{i+1} - S_{i-1}) & \text{if } j = i+1 \text{ and central differences} \\ rS_i/(S_{i+1} - S_i) & \text{if } j = i+1 \text{ and forward differences} \\ -rS_i/(S_{i+1} - S_{i-1}) & \text{if } j = i-1 \text{ and central differences} \\ 0 & \text{if } j = i-1 \text{ and forward differences.} \end{cases} \tag{3.10}$$

For future reference, note that if α_i and β_i are defined as in Algorithm 3.1, then

$$\gamma_{ij} + \frac{\sigma_{min}^2 S_i^2}{(S_{i+1} - S_{i-1})|S_j - S_i|} \geq 0. \tag{3.11}$$

Consider now a long investment. The nonlinear component of equation (3.9) can be written as

$$\sigma(\Gamma)^2\Gamma = \begin{cases} \max(\sigma_{min}^2\Gamma, 0) + \min(\sigma_{max}^2\Gamma, 0) \ ; & \text{worst-case long} \\ \max(\sigma_{max}^2\Gamma, 0) + \min(\sigma_{min}^2\Gamma, 0) \ ; & \text{best-case long.} \end{cases} \tag{3.12}$$

In order to apply a Newton iteration to the non-smooth equations (3.9), the element of the generalized Jacobian that will be used in the Newton iteration must be specified [44, 41, 48]. The derivatives for a best-case long position will be defined as

$$\frac{\partial \sigma(\Gamma)^2\Gamma}{\partial \Gamma} = \begin{cases} \sigma_{max}^2 & \text{if } \Gamma > 0 \\ \sigma_{min}^2 & \text{if } \Gamma \leq 0, \end{cases} \tag{3.13}$$

and for a worst-case long position as

$$\frac{\partial \sigma(\Gamma)^2\Gamma}{\partial \Gamma} = \begin{cases} \sigma_{min}^2 & \text{if } \Gamma > 0 \\ \sigma_{max}^2 & \text{if } \Gamma \leq 0. \end{cases} \tag{3.14}$$

In the discussion of the Newton iteration in the next section, the concept of a semismooth

function occurs [45]. Roughly speaking, a function is semismooth if its directional derivatives are continuous. Mathematically, let h be the direction in which the classical derivative value is computed:

$$F'(x;h) = \lim_{t\downarrow 0} \frac{F(x+th) - F(x)}{t}. \tag{3.15}$$

A semismooth function then has the property that the Hadamard directional derivative

$$\lim_{t\downarrow 0, h'\to h} \frac{F(x+th') - F(x)}{t} \tag{3.16}$$

exists. A more rigorous definition of semismooth functions is given in [45] and the references therein.

For further ease of analysis, the discrete equations (2.11) can be written in matrix form. Following the notation and approach of section 2.6, the discrete equations become

$$\left[I + (1-\theta)\hat{M}^{n+1}\right] U^{n+1} = \left[I - \theta\hat{M}^{n}\right] U^{n}. \tag{3.17}$$

Note that the boundary conditions (3.7) and (3.8) are of Dirichlet type, as assumed in section 2.6.

In the analysis of the following sections, we alternate between the discrete equation representation (3.9) and the matrix representation (3.17) as appropriate. Generally speaking, the matrix form is useful when discussing properties of the algorithm (e.g. convergence of the frozen coefficient iteration), while the discrete equation form is suitable for determining properties of the equation (e.g. showing monotonicity).

3.3 Theoretical Results

3.3.1 Convergence of Frozen Coefficient Iterative Scheme

Since the discrete equations for uncertain volatility problems are nonlinear, an iterative scheme must be used at each timestep. Convergence of the iterative scheme is therefore important. To this end, we have the following theorem:

Theorem 3.1 (Convergence of the nonlinear iterations). *When applied to the discrete one factor uncertain volatility equations (equation 3.17 using definitions* (3.5) *and* (3.6)*), the frozen coefficient iteration scheme described in section 2.6 converges to the unique solution given any initial iterate $\bar{U}^0$. Moreover, the iterates converge monotonically. For $\bar{U}^k$ sufficiently close to the solution, convergence is quadratic.*

Proof. The bulk of the proof follows directly from Theorem 2.1. In particular, Algorithm 3.1 guarantees that the discretization scheme does not switch between iteration k and $k+1$ (for a given node i, central differencing or forward differencing is always used, independent of k). The discretization scheme of Algorithm 3.1 also guarantees that $\left[I + (1-\theta)\hat{M}^k\right]$ in equation (3.17) is an M-matrix.

To complete the proof, note that the one factor uncertain volatility problems are of the bang-bang type (σ always assumes one of the extreme values [6]). Hence, the frozen coefficient method is equivalent to a non-smooth Newton iteration using definitions (3.13) and (3.14). Further, since Γ_i^n as defined by equation (2.14) is a simple linear function of U_i^n and U_j^n, the non-smooth equations (3.9) are strongly semismooth [45]. This means that convergence will be quadratic in a sufficiently small neighbourhood of the solution [44].

□

Thus, it has been shown that frozen coefficient method will converge to the unique solution of equation (3.2) at each timestep. However, this says nothing about convergence

to the viscosity solution of the PDE as the timestep and mesh size are reduced. This topic will be discussed in the next section.

3.3.2 Convergence to the Viscosity Solution

In the previous section, global convergence of the iterations for the nonlinear algebraic equations at each timestep was proven. However, since the PDE is nonlinear, questions remain about convergence to the financially correct solution as $\Delta S, \Delta\tau \to 0$. In a financial context, convergence to the viscosity solution [15] is desired. The existence and uniqueness of the viscosity solution for one factor uncertain volatility problems was proven in [56], with related comments/viewpoints in [9] and [38]. From Theorem 2.2, we know that a stable, consistent, and monotone discretization will converge to the viscosity solution. To this end, it is helpful to review the concept of a monotone discretization.

The set of discrete equations (3.9) can be written as

$$g_i(U_i^{n+1}, U_j^{n+1}, U_i^n, U_j^n) = 0 \quad \forall i \; ; \quad j \in \eta_i. \tag{3.18}$$

With a monotone discretization, a positive perturbation to any of $\{U_j^{n+1}, U_i^n, U_j^n\}$ produces a positive perturbation of U_i^{n+1}. If g_i is differentiable, then this is equivalent to stating that

$$\begin{aligned}
\frac{\partial U_i^{n+1}}{\partial U_i^n} &= -\frac{\partial g_i/\partial U_i^n}{\partial g_i/\partial U_i^{n+1}} \geq 0 \\
\frac{\partial U_i^{n+1}}{\partial U_j^{n+1}} &= -\frac{\partial g_i/\partial U_j^{n+1}}{\partial g_i/\partial U_i^{n+1}} \geq 0 \\
\frac{\partial U_i^{n+1}}{\partial U_j^n} &= -\frac{\partial g_i/\partial U_j^n}{\partial g_i/\partial U_i^{n+1}} \geq 0.
\end{aligned} \tag{3.19}$$

In the case of nondifferentiable g_i, which is the case for the uncertain volatility discretization, the following definition of monotonicity can be used:

Definition 3.1 (Monotone Discretizations). *A discretization of the form* (3.18) *is monotone if either*

$$\begin{aligned} g_i(U_i^{n+1}, U_j^{n+1} + \rho_j^{n+1}, U_i^n + \rho_i^n, U_j^n + \rho_j^n) \geq g_i(U_i^{n+1}, U_j^{n+1}, U_i^n, U_j^n) \quad \forall i \; ; \quad j \in \eta_i \\ \forall \rho_i^n \geq 0, \; \forall \rho_j^{n+1} \geq 0, \; \forall \rho_j^n \geq 0 \\ g_i(U_i^{n+1} + \rho_i^{n+1}, U_j^{n+1}, U_i^n, U_j^n) \leq g_i(U_i^{n+1}, U_j^{n+1}, U_i^n, U_j^n) \quad \forall i \; ; \quad j \in \eta_i \\ \forall \rho_i^{n+1} \geq 0 \end{aligned} \tag{3.20}$$

or

$$\begin{aligned} g_i(U_i^{n+1}, U_j^{n+1} + \rho_j^{n+1}, U_i^n + \rho_i^n, U_j^n + \rho_j^n) \leq g_i(U_i^{n+1}, U_j^{n+1}, U_i^n, U_j^n) \quad \forall i \; ; \quad j \in \eta_i \\ \forall \rho_i^n \geq 0, \; \forall \rho_j^{n+1} \geq 0, \; \forall \rho_j^n \geq 0 \\ g_i(U_i^{n+1} + \rho_i^{n+1}, U_j^{n+1}, U_i^n, U_j^n) \geq g_i(U_i^{n+1}, U_j^{n+1}, U_i^n, U_j^n) \quad \forall i \; ; \quad j \in \eta_i \\ \forall \rho_i^{n+1} \geq 0 \end{aligned} \tag{3.21}$$

This somewhat longwinded definition, with either condition (3.20) or (3.21), avoids having to define $g_i(\ldots)$ with a standard sign convention.

The difficulty in verifying these relations for the discrete equations (3.9) comes from the nonlinear term. However, note that

$$\begin{aligned} \Gamma_i^n(U_i^n, U_j^n + \rho_j^n) &= \Gamma_i^n(U_i^n, U_j^n) + \rho_j^n \frac{2}{(S_{i+1} - S_{i-1})|S_j - S_i|} \; ; \quad \forall \rho_j^n \geq 0 \\ \Gamma_i^n(U_i^n + \rho_i^n, U_j^n) &= \Gamma_i^n(U_i^n, U_j^n) - \rho_i^n \sum_{j \in \eta_i} \frac{2}{(S_{i+1} - S_{i-1})|S_j - S_i|} \; ; \quad \forall \rho_i^n \geq 0. \end{aligned} \tag{3.22}$$

Further, regardless of whether $\sigma(\Gamma)$ is defined for best or worst-case, it follows from equations

(3.3)-(3.4) that

$$\begin{aligned}
&\sigma(\Gamma)^2\Gamma + \sigma_{max}^2\Delta\Gamma \geq \sigma(\Gamma+\Delta\Gamma)^2(\Gamma+\Delta\Gamma) \geq \sigma(\Gamma)^2\Gamma + \sigma_{min}^2\Delta\Gamma \ ; \quad \forall \Delta\Gamma \geq 0 \\
&\sigma(\Gamma)^2\Gamma - \sigma_{max}^2\Delta\Gamma \leq \sigma(\Gamma-\Delta\Gamma)^2(\Gamma-\Delta\Gamma) \leq \sigma(\Gamma)^2\Gamma - \sigma_{min}^2\Delta\Gamma \ ; \quad \forall \Delta\Gamma \geq 0.
\end{aligned} \tag{3.23}$$

We are now in a position to test equations (3.9) for monotonicity, and hence determine the conditions under which convergence to the viscosity solution is guaranteed. This will be done separately for the fully implicit and Crank-Nicolson cases.

Fully Implicit Discretization

For a fully implicit discretization, set $\theta = 0$ in equation (3.9) to obtain (for $i < imax$)

$$g_i = -U_i^{n+1} + U_i^n + \left[\sum_{j\in\eta_i} \Delta\tau\gamma_{ij}(U_j^{n+1} - U_i^{n+1}) - r\Delta\tau U_i^{n+1}\right] + \Delta\tau\frac{\sigma(\Gamma_i^{n+1})^2 S_i^2}{2}\Gamma_i^{n+1} = 0, \tag{3.24}$$

and for $i = imax$

$$g_{imax} = -U_{imax}^{n+1} + U_{imax}^n = 0 \ .$$

In order to show convergence to the viscosity solution of this discretization, the following Lemmas will be used:

Lemma 3.1 (Monotonicity of the fully implicit discretization). *The fully implicit discretization* (3.24) *is monotone, independent of any choice of* $\Delta\tau$ *and grid spacing.*

Proof. Note that all U_j^n terms have disappeared in equation (3.24). Consider perturbing

U_j^{n+1} by an amount $\epsilon > 0$. In this case, using relations (3.22), (3.23), and (3.24) gives

$$\begin{aligned} g_i(U_i^{n+1}, U_j^{n+1} + \epsilon, U_i^n) &\geq g_i(U_i^{n+1}, U_j^{n+1}, U_i^n) + \Delta\tau\epsilon \left(\frac{\sigma_{min}^2 S_i^2}{(S_{i+1} - S_{i-1})|S_j - S_i|} + \gamma_{ij} \right) \\ &\geq g_i(U_i^{n+1}, U_j^{n+1}, U_i^n), \end{aligned} \tag{3.25}$$

where the last line follows from relation (3.11). Continuing in the same manner, perturb U_i^{n+1} by $\epsilon > 0$ to get:

$$\begin{aligned} g_i(U_i^{n+1} + \epsilon, U_j^{n+1}, U_i^n) &\leq g_i(U_i^{n+1}, U_j^{n+1}, U_i^n) - \epsilon \\ &\quad - \Delta\tau\epsilon \sum_{j \in \eta_i} \left(\frac{\sigma_{min}^2 S_i^2}{(S_{i+1} - S_{i-1})|S_j - S_i|} + \gamma_{ij} \right) - \Delta\tau\epsilon r \\ &\leq g_i(U_i^{n+1}, U_j^{n+1}, U_i^n). \end{aligned} \tag{3.26}$$

Again, the factor in the summation is guaranteed to be positive by relation (3.11). It is obvious from equation (3.24) that

$$g_i(U_i^{n+1}, U_j^{n+1}, U_i^n + \epsilon) \geq g_i(U_i^{n+1}, U_j^{n+1}, U_i^n). \tag{3.27}$$

The monotonicity of equation (3.24) now follows directly from definition (3.20). □

Lemma 3.2 (Stability of the fully implicit discretization). *The fully implicit discretization* (3.24) *is unconditionally stable, in the sense that*

$$\|U^{n+1}\|_\infty \leq \|U^n\|_\infty$$

independent of the timestep and mesh spacing.

Proof. Define

$$\begin{aligned} U^n_{max} &= \max_i U^n_i \\ U^n_{min} &= \min_i U^n_i. \end{aligned} \tag{3.28}$$

From Lemma 3.1 we have that a fully implicit discretization is unconditionally monotone. It follows that

$$\begin{aligned} U^{n+1}_{max} &\leq \max_i (U^*_{max})_i \\ U^{n+1}_{min} &\geq \min_i (U^*_{min})_i, \end{aligned} \tag{3.29}$$

where $(U^*_{max})_i, (U^*_{min})_i$ are given from the solutions to

$$\begin{aligned} g_i((U^*_{max})_i, (U^*_{max})_i, U^n_{max}) &= 0 \\ g_i((U^*_{min})_i, (U^*_{min})_i, U^n_{min}) &= 0. \end{aligned} \tag{3.30}$$

Hence, equation (3.24) and equation (3.30) lead to

$$\max\left(U^n_{max}, \frac{U^n_{max}}{1+r\Delta\tau}\right) \geq U^{n+1}_i \geq \min\left(U^n_{min}, \frac{U^n_{min}}{1+r\Delta\tau}\right); \quad \forall i \ . \tag{3.31}$$

Consequently, we have that $\|U^{n+1}\|_\infty \leq \|U^n\|_\infty$. □

Let $\Delta S = \max_i(S_{i+1} - S_i)$. Our main result concerning convergence of the fully implicit discretization is the following:

Theorem 3.2 (Convergence of the fully implicit discretization). *The fully implicit discretization* (3.24) *converges unconditionally to the viscosity solution of the nonlinear PDE* (3.2)*, as* $\Delta\tau$, $\Delta S \to 0$.

Proof. Theorem 2.2 shows that a consistent, stable and monotone discretization converges to the viscosity solution. Since equation (3.24) is a consistent discretization, Theorem 3.2 follows directly from the results of Theorem 2.2, and Lemmas 3.1-3.2. □

Crank-Nicolson Discretization

For a Crank-Nicolson discretization, set $\theta = 1/2$ in equation (3.9) to obtain (for $i < imax$)

$$\begin{aligned} g_i = -U_i^{n+1} + U_i^n &+ \frac{1}{2}\left[\sum_{j\in\eta_i} \Delta\tau\gamma_{ij}(U_j^{n+1} - U_i^{n+1}) - r\Delta\tau U_i^{n+1}\right] \\ &+ \frac{1}{2}\left[\sum_{j\in\eta_i} \Delta\tau\gamma_{ij}(U_j^{n} - U_i^{n}) - r\Delta\tau U_i^{n}\right] + \frac{1}{2}\Delta\tau\frac{\sigma(\Gamma_i^{n+1})^2 S_i^2}{2}\Gamma_i^{n+1} \\ &+ \frac{1}{2}\Delta\tau\frac{\sigma(\Gamma_i^{n})^2 S_i^2}{2}\Gamma_i^{n} = 0, \end{aligned} \tag{3.32}$$

and for $i = imax$

$$g_i = -U_i^{n+1} + U_i^n = 0\ .$$

As with the fully implicit discretization, we will first determine the conditions for monotonicity.

Lemma 3.3 (Monotonicity of Crank-Nicolson discretization). *The Crank-Nicolson discretization* (3.32) *is monotone if the timestep is selected to satisfy*

$$\Delta\tau < 2\min_i \left(r + \sum_{j\in\eta_i} \left[\frac{\sigma_{max}^2 S_i^2}{(S_{i+1} - S_{i-1})|S_j - S_i|} + \gamma_{ij} \right] \right)^{-1} . \tag{3.33}$$

Proof. Following the fully implicit analysis above, we immediately obtain in the Crank-

Nicolson case (for $\epsilon > 0$)

$$g_i(U_i^{n+1}, U_j^{n+1} + \epsilon, U_i^n, U_j^n) \geq g_i(U_i^{n+1}, U_j^{n+1}, U_i^n, U_j^n)$$
$$g_i(U_i^{n+1}, U_j^{n+1}, U_i^n, U_j^n + \epsilon) \geq g_i(U_i^{n+1}, U_j^{n+1}, U_i^n, U_j^n)$$
$$g_i(U_i^{n+1} + \epsilon, U_j^{n+1}, U_i^n, U_j^n) \leq g_i(U_i^{n+1}, U_j^{n+1}, U_i^n, U_j^n).$$

Using relations (3.22), (3.23), and (3.32) gives ($\epsilon > 0$)

$$g_i(U_i^{n+1}, U_j^{n+1}, U_i^n + \epsilon, U_j^n) \geq g_i(U_i^{n+1}, U_j^{n+1}, U_i^n, U_j^n) + \epsilon\left(1 - \frac{r\Delta\tau}{2}\right) - \frac{\Delta\tau\epsilon}{2}\sum_{j\in\eta_i}\left(\frac{\sigma_{max}^2 S_i^2}{(S_{i+1} - S_{i-1})|S_j - S_i|} + \gamma_{ij}\right). \tag{3.34}$$

Since $\sigma_{max} \geq \sigma_{min}$, it follows from equation (3.11) that

$$\gamma_{ij} + \frac{\sigma_{max}^2 S_i^2}{(S_{i+1} - S_{i-1})|S_j - S_i|} \geq 0. \tag{3.35}$$

Consequently, for the perturbation ($\epsilon > 0$) to produce a positive change, we require

$$0 < \epsilon\left[1 - \frac{r\Delta\tau}{2} - \frac{\Delta\tau}{2}\sum_{j\in\eta_i}\left(\frac{\sigma_{max}^2 S_i^2}{(S_{i+1} - S_{i-1})|S_j - S_i|} + \gamma_{ij}\right)\right] \quad \forall i. \tag{3.36}$$

To ensure that the timestep condition is satisfied for all i, the worst-case is

$$\Delta\tau < 2\min_i\left(r + \sum_{j\in\eta_i}\left[\frac{\sigma_{max}^2 S_i^2}{(S_{i+1} - S_{i-1})|S_j - S_i|} + \gamma_{ij}\right]\right)^{-1}, \tag{3.37}$$

which completes the proof. □

Note that condition (3.33) implies that a Crank-Nicolson scheme will be monotone only

if the timestep size is less than twice the maximum stable explicit timestep size.

Lemma 3.4 (Stability of the Crank-Nicolson discretization). *If condition* (3.33) *is satisfied, then the Crank-Nicolson discretization is stable, in the sense that*

$$\|U^{n+1}\|_\infty \leq \|U^n\|_\infty$$

independent of the timestep and mesh spacing.

Proof. If condition (3.33) is satisfied, then by Lemma 3.3, the Crank-Nicolson discretization is monotone, and stability of the discrete equations follows by bounding the U_i values as was done in the fully implicit case (see Lemma 3.2). □

Combining these results allows us to state the following theorem:

Theorem 3.3 (Convergence of the Crank-Nicolson discretization). *If condition* (3.33) *is satisfied, then the Crank-Nicolson discretization* (3.32) *converges to the viscosity solution of the nonlinear PDE* (3.2) *as* $\Delta\tau, \Delta S \to 0$.

Proof. Again, this follows directly from Lemmas 3.3 and 3.4 and Theorem 2.2 since the condition of this theorem ensures a monotone, stable, and consistent discretization. □

If condition (3.33) is not satisfied, then for $\theta = 1/2$ (Crank-Nicolson) in equation (2.35), we have

$$\left[I + \frac{\hat{M}^{n+1}}{2}\right] U^{n+1} = \left[I - \frac{\hat{M}^n}{2}\right] U^n, \tag{3.38}$$

or

$$\begin{aligned} U^{n+1} &= \left[I + \frac{\hat{M}^{n+1}}{2}\right]^{-1} \left[I - \frac{\hat{M}^{n}}{2}\right] U^n \\ &= \left[I + \frac{\hat{M}^{n+1}}{2}\right]^{-1} \left[I - \frac{\hat{M}^{n}}{2}\right] \left[I + \frac{\hat{M}^{n}}{2}\right]^{-1} \left[I - \frac{\hat{M}^{n-1}}{2}\right] \left[I + \frac{\hat{M}^{n-1}}{2}\right]^{-1} \\ &\times \ldots \left[I - \frac{\hat{M}^{0}}{2}\right] U^0. \end{aligned} \tag{3.39}$$

Assuming that $r > 0$, all rows of $\hat{M}^n$ are diagonally dominant, with positive diagonal entries, except for the last row which is identically zero. It follows that all eigenvalues of $\hat{M}^n$ have non-negative real parts, and hence the eigenvalues of each term

$$\left[I - \frac{\hat{M}^{k}}{2}\right] \left[I + \frac{\hat{M}^{k}}{2}\right]^{-1}$$

are less than or equal to one in magnitude, with only one eigenvalue having magnitude one. However, stability does not necessarily follow, since the $\hat{M}^k$s are not normal matrices. In fact, as will be seen, Crank-Nicolson timestepping appears to be unstable with discontinuous payoffs when condition (3.33) is violated.

3.4 Numerical Results

3.4.1 Butterfly Spread

To illustrate the results of the theoretical analysis, tests are made of the uncertain volatility model on a "butterfly spread" payoff. This payoff is a combination of options with three different strike prices. It can be formed using either call options or put options. The test

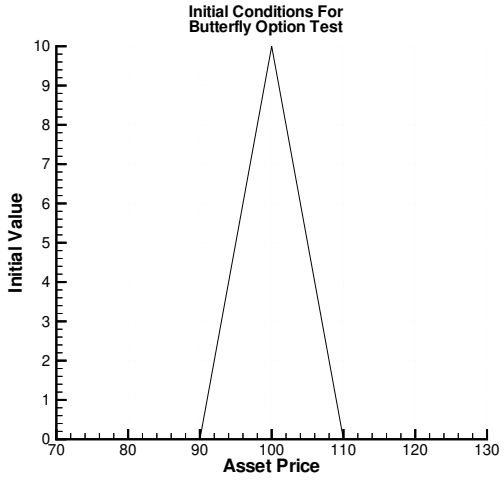


FIGURE 3.1: *Sample payoff function for a butterfly spread. Parameters for equation* (3.40) *are taken from Table 3.1.*

problems in this section use call options, so the payoff can be written as

$$U(S, \tau = 0) = \max(S - K_1, 0) - 2\max(S - (K_1 + K_2)/2, 0) + \max(S - K_2, 0). \quad (3.40)$$

This corresponds to a long position in two calls at strikes K_1, K_2, and a short position in two calls at strike $(K_1 + K_2)/2$. Recall that $\tau = T - t$, so that the payoff is the value of the option at expiry $t = T$, or the initial condition of the PDE at $\tau = 0$. Figure 3.1 provides a diagram of a sample payoff function. A complete specification of the test problem, including the PDE parameters, is given in Table 3.1. Note that unless the problem has a non-convex solution at some time during the solution process, the sign of gamma (U_{SS}) will not change. In these cases the nonlinearity disappears, and σ will always take on one of the extreme volatility values.

Solutions were computed on a sequence of uniformly refined grids, starting with 61 non-uniform points. At each grid refinement, the timestep was halved (initial timestep was 0.01). The convergence tolerance for frozen coefficient method described in Algorithm

Type	Butterfly spread (using call options)
Scenario	Worst-case long
Time to expiry	0.25 years
r	0.1
K_1	90
K_2	110
σ_{max}	0.25
σ_{min}	0.15

TABLE 3.1: *Model parameters for the butterfly spread test problem.*

	Fully Implicit			Crank-Nicolson		
Nodes	Value	Diff	Ratio	Value	Diff	Ratio
61	2.3501			1.7246		
121	2.3250	0.0251		1.5713	0.1533	
241	2.3116	0.0134	1.87	1.4622	0.1091	1.41
481	2.3047	0.0069	1.95	1.3806	0.0816	1.34
961	2.3012	0.0035	1.97	1.3264	0.0542	1.51

TABLE 3.2: *Convergence results for an at-the-money ($S = 100$) butterfly spread with uncertain volatility. Parameters are provided in Table 3.1. The timestep is halved at each grid refinement. "Diff" is the absolute value of the change in the solution as the grid is refined. "Ratio" is the ratio of successive differences. Timestepping data are given in Table 3.3.*

2.2 was 10^{-6}. Convergence results using fully implicit and Crank-Nicolson timestepping are given in Table 3.2. The timestep was selected so that condition (3.33) was violated. We can see that a fully implicit method converges at a linear rate, as expected. From Theorem 3.2, we know that this solution is the viscosity solution. However, the Crank-Nicolson method is either converging to a non-viscosity solution, or has a slowly growing instability. Timestepping and nonlinear iteration data are given in Table 3.3. Note that in the fully implicit case, the average number of nonlinear iterations per step is close to two. For a linear problem, the number of nonlinear iterations at each step would be exactly two (of course, iteration would be unnecessary in this case).

To understand the difficulty with Crank-Nicolson timestepping, consider plots of the solution values, deltas (U_S) and gammas (U_{SS}), as shown in Figure 3.2 for both fully

No. of Timesteps	$\Delta\tau$	Fully Implicit		Crank-Nicolson	
		No. of Iterations	Average No. of Iterations	No. of Iterations	Average No. of Iterations
25	0.01	58	2.32	87	3.48
50	0.005	116	2.32	204	4.08
100	0.0025	236	2.36	432	4.32
200	0.00125	461	2.31	886	4.43
400	0.000625	868	2.17	1858	4.65

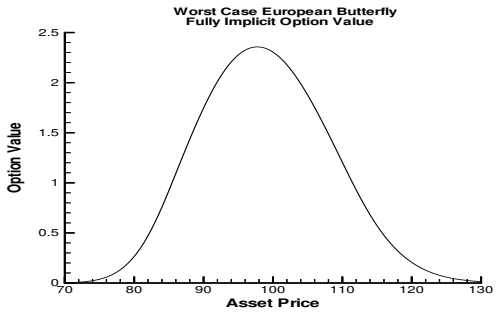
TABLE 3.3: *Timestepping information for an at-the-money* ($S = 100$) *butterfly spread with uncertain volatility. Parameters are provided in Table 3.1. "No. of Iterations" is the total number of nonlinear iterations used during the solution process. "Average No. of Iterations" is the number of iterations divided by the number of timesteps. The number of nodes is doubled each time the timestep is halved. The convergence tolerance was* 10^{-6} (*Algorithm 2.2*). *Convergence data are given in Table 3.2.*

implicit and Crank-Nicolson timestepping.[1] Delta and gamma values were determined by finite differencing the numerical solution. We see that all implicit plots are smooth, as would be expected. However, a small "kink" in the Crank-Nicolson solution at $S = 100$ leads to a discontinuity in the solution delta (U_S), and to major oscillations in the solution gamma (U_{SS}) values. Since the uncertain volatility model has a crucial dependence on the sign of gamma, problems are anticipated.

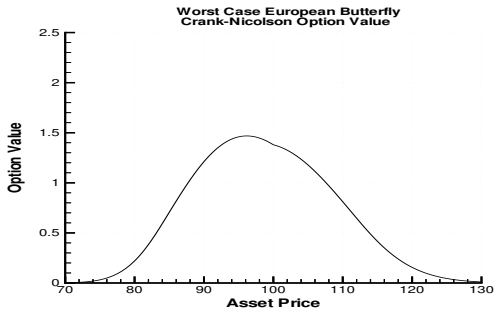
To further isolate the source of Crank-Nicolson timestepping difficulties, consider the solution after one timestep. Figure 3.3(a) shows that implicit timestepping leads to a smooth curve. On the other hand, Figure 3.3(b) reveals that Crank-Nicolson timestepping has introduced a cusp at the strike price of 100 (other problems at $S = 90$ and $S = 110$ are not shown). For linear problems, such oscillations would eventually be damped out, since Crank-Nicolson is a stable method.

However, for the present problem, the oscillations have caused the computed values of gamma at all nodes where the initial payoff has a discontinuity in delta to have the wrong sign. This is shown in Figure 3.4. Consequently, different values of σ will be used at

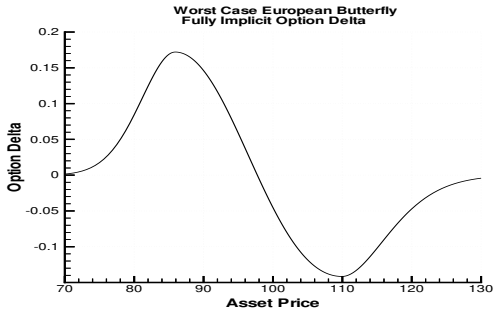
[1]Recall that risk hedging strategies typically involve delta and gamma, so it is important to accurately compute not only the option value but also its first and second derivatives with respect to the price of the underlying asset.



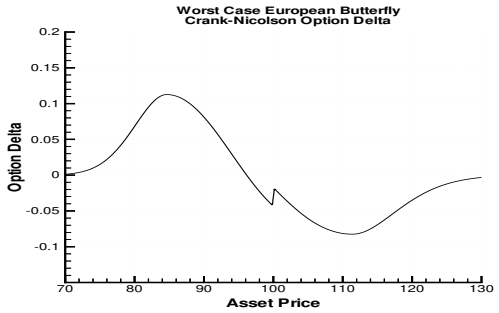
(A) Value, fully implicit timestepping.



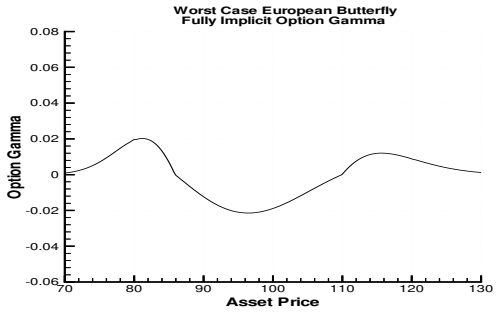
(B) Value, Crank-Nicolson timestepping.



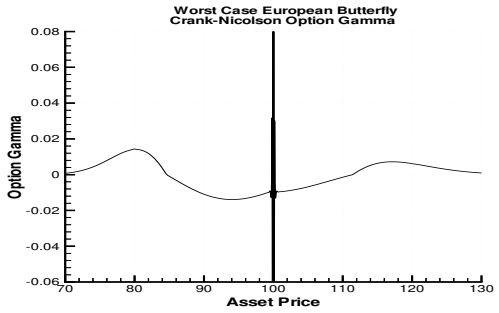
(C) Delta, fully implicit timestepping.



(D) Delta, Crank-Nicolson timestepping.

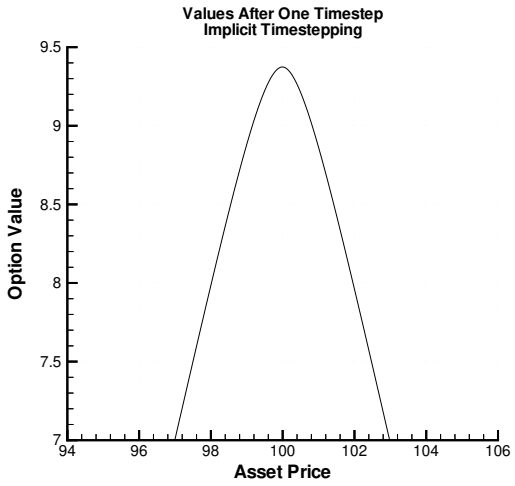


(E) Gamma, fully implicit timestepping.

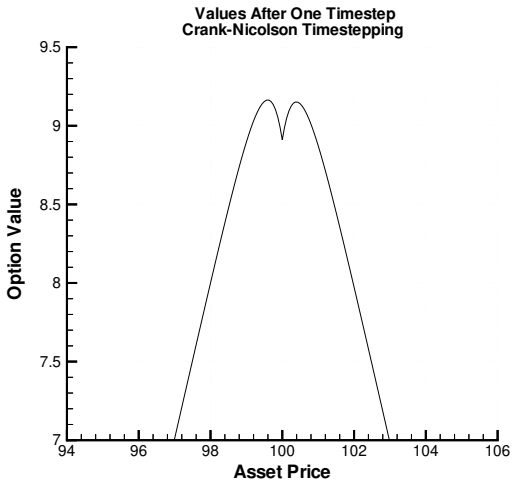


(F) Gamma, Crank-Nicolson timestepping.

FIGURE 3.2: *Butterfly spread solution value* (U)*, delta* (U_S)*, and gamma* (U_{SS}) *for both fully implicit and Crank-Nicolson timestepping at time* $\tau = T$*. Parameters are provided in Table 3.1.*

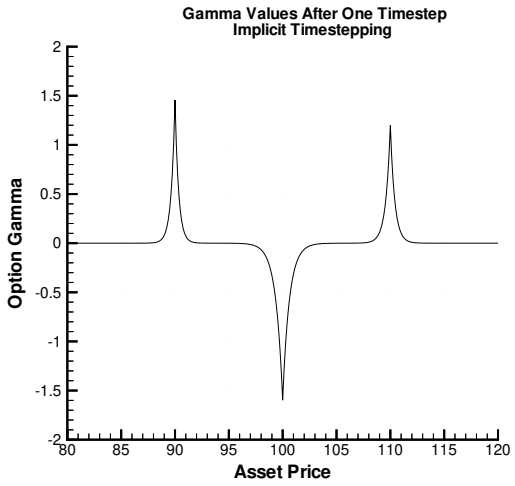


(A) Fully implicit timestepping.

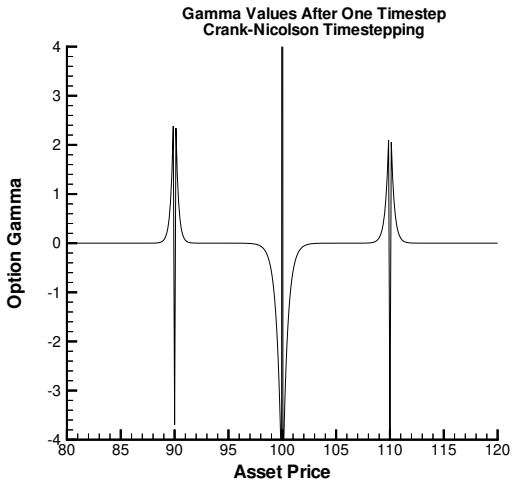


(B) Crank-Nicolson timestepping.

FIGURE 3.3: *Butterfly spread solution value (U) after the first timestep ($\tau = \Delta\tau$).*



(A) Fully implicit timestepping.



(B) Crank-Nicolson timestepping.

FIGURE 3.4: *Butterfly spread solution gamma (U_{SS}) after the first timestep ($\tau = \Delta\tau$). Note that the sign of gamma is different at asset values of 90, 100, and 110 where the delta is originally discontinuous.*

the second timestep at the nodes where oscillations occurred. In this case, since we have the convergence results from a monotone scheme, which is guaranteed to converge to the viscosity solution, we know that the Crank-Nicolson solution is incorrect.

The oscillations also have an effect on how strongly nonlinear the problem becomes. Table 3.3 shows that using implicit timestepping leads to just over 2 nonlinear iterations per timestep, indicating fairly mild nonlinearity. On the other hand, the average number of nonlinear iterations per timestep for Crank-Nicolson timestepping starts at 3.48 for the the coarsest grid, and increases as the grids are refined. Clearly, the non-monotone discretization amplifies the nonlinear properties of the problem.

Unfortunately, we would like to use Crank-Nicolson timestepping for the potential of second order convergence. Since the difficulty appears to be oscillations at the first timestep, it makes sense to start with fully implicit timestepping, and then to switch to Crank-Nicolson timestepping. Such a method was discussed in section 2.4, and will be called Rannacher timestepping below. Since only a finite number of fully implicit steps are taken, the overall convergence rate can be shown to be quadratic for linear problems [46]. Results using this method with 2 and 4 initial implicit steps are given in Table 3.4. Both approaches give (nearly) quadratic convergence, although taking 4 steps appears to converge at a slightly higher rate. More importantly, even though Rannacher timestepping is not unconditionally monotone (strictly speaking, and considering that the timesteps taken violated condition (3.33)), both methods appear to converge to the correct solution, with no evidence of instability. Assuming a linear rate of convergence, the extrapolated solution using fully implicit timestepping (Table 3.2) is 2.2977, in excellent agreement with the results in Table 3.4. Further, as shown in Table 3.5, Rannacher timestepping reduces the average number of nonlinear iterations per timestep to approximately the same levels as for implicit timestepping.

It is interesting to observe that difficulties normally arise in nonlinear financial PDEs

	Rannacher (2 steps)			Rannacher (4 steps)		
Nodes	Value	Diff	Ratio	Value	Diff	Ratio
61	2.2985785			2.3040657		
121	2.2980535	0.0005250		2.2996153	0.0044504	
241	2.2977860	0.0002675	1.96	2.2981945	0.0014208	3.13
481	2.2977116	0.0000744	3.60	2.2978172	0.0003773	3.77
961	2.2976910	0.0000206	3.61	2.2977178	0.0000994	3.80

TABLE 3.4: *Convergence results for an at-the-money* ($S = 100$) *butterfly spread with uncertain volatility and Rannacher timestepping. Parameters are provided in Table 3.1. The timestep is halved at each grid refinement. "Diff" is the absolute value of the change in the solution as the grid is refined. "Ratio" is the ratio of successive differences. Timestepping data are given in Table 3.5.*

		Rannacher (2 steps)		Rannacher (4 steps)	
No. of Timesteps	$\Delta\tau$	No. of Iterations	Average No. of Iterations	No. of Iterations	Average No. of Iterations
25	0.01	59	2.36	59	2.36
50	0.005	116	2.32	118	2.36
100	0.0025	235	2.35	235	2.35
200	0.00125	459	2.30	456	2.28
400	0.000625	847	2.12	847	2.12

TABLE 3.5: *Timestepping information for an at-the-money* ($S = 100$) *butterfly spread with uncertain volatility and Rannacher timestepping. Parameters are provided in Table 3.1. "No. of Iterations" is the total number of nonlinear iterations used during the solution process. "Average No. of Iterations" is the number of iterations divided by the number of timesteps. The number of nodes is doubled as the timestep sized is halved. The convergence tolerance was* 10^{-6} (*Algorithm 2.2*). *Convergence data are given in Table 3.4.*

when the PDE degenerates to a nonlinear hyperbolic problem. In our case, no degeneracy occurs. However, the payoff has a discontinuous first derivative, which seems to be enough to cause difficulty. The use of a few fully implicit steps at the start smooths the solution, curing the problem.

An obvious approach which avoids having to solve a set of nonlinear algebraic equations at each step is to evaluate the uncertain volatility explicitly (see the end of section 2.5). More precisely, equation (2.35) becomes

$$\left[I + (1-\theta)\hat{M}^n\right] U^{n+1} = \left[I - \theta\hat{M}^n\right] U^n. \tag{3.41}$$

This approach is equivalent to forcing one nonlinear iteration per timestep. Since matrix $\left[I + (1-\theta)\hat{M}^n\right]$ is a diagonally dominant M-matrix, algorithm (3.41) with $\theta = 0$ is a positive coefficient discretization [32]. Consequently, algorithm (3.41) with $\theta = 0$ is unconditionally stable, but not monotone. As a result, convergence to the viscosity solution is not guaranteed.

Results using this approach are given in Table 3.6. Algorithm (3.41) with $\theta = 0$ leads to linear convergence, and apparently converges to the viscosity solution. Algorithm (3.41) with $\theta = 1/2$ leads to values away from the correct solution, with perhaps some instability. Setting $\theta = 1/2$ in algorithm (3.41) and using Rannacher timestepping (with 4 initial steps using $\theta = 0$) leads to linear convergence. The potential for quadratic convergence is lost by only taking one nonlinear iteration per timestep. However, if high accuracy is not required, the simplicity and speed of using fully implicit timestepping with the linear approximation (explicit evaluation of the volatility) may be advantageous.

	Algorithm (3.41) ($\theta = 0$)			Algorithm (3.41) ($\theta = 1/2$)			Algorithm (3.41) ($\theta = 1/2$) and Rannacher (4 steps)		
Nodes	Value	Diff	Ratio	Value	Diff	Ratio	Value	Diff	Ratio
61	2.4234699			3.2100384			2.3766897		
121	2.3676732	0.05580		3.2593402	0.04930		2.3419918	0.03470	
241	2.3347374	0.03294	1.69	3.3127764	0.05344	0.92	2.3212272	0.02076	1.67
481	2.3169350	0.01780	1.85	3.3176336	0.00486	11.0	2.3099994	0.01122	1.85
961	2.3075759	0.00936	1.90	3.3141664	-0.00346	-1.40	2.3040654	0.00593	1.89

TABLE 3.6: *Convergence results for an at-the-money ($S = 100$) butterfly spread with uncertain volatility, solved by forcing one nonlinear iteration per timestep (as per equation (3.41)). Parameters are provided in Table 3.1. "Diff" is the change in the solution as the grid is refined. "Ratio" is the ratio of successive differences. The timestep is halved as the grid is refined.*

3.4.2 Digital Call Options

We have seen how a non-smooth payoff condition can cause problems for non-monotone schemes. This situation will be even more problematic for discontinuous payoffs. A digital call option has the payoff

$$U(S, \tau = 0) = \begin{cases} 1 & \text{if } S \geq K \\ 0 & \text{if } S < K. \end{cases} \tag{3.42}$$

In chapter 2, it was shown that smoothing discontinuous initial conditions can be beneficial. For the digital option tests in this chapter, the initial conditions have been projected onto the space of linear basis functions. Note that such a projection would have no effect on the butterfly payoff above, and strictly speaking, is not required for implicit timestepping.

The digital option problem is solved on the same set of grids as for the butterfly spread. At each refinement stage, the number of nodes is doubled and the timestep is halved (initial timestep of 0.01). The convergence tolerance for the frozen coefficient nonlinear iteration was 10^{-6} (Algorithm 2.2). The remaining parameters are given in Table 3.7.

Results for fully implicit, Crank-Nicolson, and Rannacher timestepping (4 implicit steps) are given in Table 3.8. Implicit timestepping leads to consistent linear convergence. The Crank-Nicolson values appear to be unstable, having gone negative and approximately doubling in magnitude at each refinement. This indicates that the timestep restriction (3.33)

Type	Digital call
Scenario	Worst-case long
Time to expiry	0.25 years
r	0.1
K	100
σ_{max}	0.25
σ_{min}	0.15

TABLE 3.7: *Model parameters for the digital call option test problem.*

	Fully Implicit			Crank-Nicolson			Rannacher (4 steps)		
Nodes	Value	Diff	Ratio	Value	Diff	Ratio	Value	Diff	Ratio
61	0.4431338			-0.25573486			0.4432821		
121	0.4425493	0.0005845		-1.2994648	1.044		0.4426167	0.0006654	
241	0.4422251	0.0003242	1.80	-3.4379189	2.138	0.488	0.4422552	0.0003615	1.84
481	0.4420542	0.0001709	1.90	-7.7664872	4.329	0.494	0.4420673	0.0001878	1.92
961	0.4419641	0.0000901	1.90	-16.644796	8.878	0.488	0.4419698	0.0000975	1.93

TABLE 3.8: *Convergence results for an at-the-money ($S = 100$) digital call option with uncertain volatility. "Diff" is the absolute value of the change in the solution as the grid is refined. "Ratio" is the ratio of successive differences. The timestep is halved at each grid refinement. Timestepping data are given in Table 3.9.*

is of practical importance. Using Rannacher timestepping with 4 implicit steps restores convergence, but only at a linear rate. Similar observations apply to the average number of nonlinear iterations per timestep, as shown in Table 3.9. Both fully implicit and Rannacher timestepping take an average of just over 2 nonlinear iterations, while Crank-Nicolson requires over 4 iterations on average.

Given the lack of quadratic convergence (even for Rannacher timestepping), one may

		Fully Implicit		Crank-Nicolson		Rannacher (4 steps)	
No. of Timesteps	$\Delta\tau$	No. of Iterations	Average No. of Iterations	No. of Iterations	Average No. of Iterations	No. of Iterations	Average No. of Iterations
25	0.01	55	2.20	105	4.20	55	2.20
50	0.005	110	2.20	212	4.24	109	2.18
100	0.0025	212	2.12	442	4.42	208	2.08
200	0.00125	407	2.04	877	4.39	406	2.03
400	0.000625	805	2.01	1797	4.49	803	2.01

TABLE 3.9: *Timestepping information for an at-the-money ($S = 100$) digital call option with uncertain volatility. "No. of Iterations" is the total number of nonlinear iterations used during the solution process. "Average No. of Iterations" is the number of iterations divided by the number of timesteps. The number of nodes is doubled as the timestep is halved. The convergence tolerance was 10^{-6} (Algorithm 2.2). Convergence data are given in Table 3.8.*

	Algorithm (3.41) ($\theta = 0$)			Algorithm (3.41) ($\theta = 1/2$)			Algorithm (3.41) ($\theta = 1/2$) and Rannacher (4 steps)		
Nodes	Value	Diff	Ratio	Value	Diff	Ratio	Value	Diff	Ratio
61	0.459603			0.933258			0.459550		
121	0.456200	0.00340		1.374314	0.441		0.456176	0.00337	
241	0.452571	0.00363	0.94	2.172795	0.798	0.552	0.452566	0.00361	0.93
481	0.449795	0.00278	1.31	3.746378	1.57	0.507	0.449796	0.00277	1.30
961	0.447641	0.00216	1.29	6.941466	3.20	0.493	0.447642	0.00215	1.29

TABLE 3.10: *Convergence results for an at-the-money* ($S = 100$) *digital call option with uncertain volatility, solved by forcing one nonlinear iteration per timestep* (*as per equation* (3.41)). *Parameters are provided in Table 3.7. "Diff" is the absolute value of the change in the solution as the grid is refined. "Ratio" is the ratio of successive differences. The timestep is halved at each grid refinement.*

think that the simplicity of linearizing the problem as per equation (3.41) is even more advantageous for digital options. However, as shown by the values in Table 3.10, this is not necessarily true. The results for algorithm (3.41) ($\theta = 0$) and algorithm (3.41) ($\theta = 1/2$) with Rannacher timestepping (first four steps use $\theta = 0$ in equation (3.41), and $\theta = 1/2$ thereafter) appear to be convergent, but only at a sub-linear rate. Values for algorithm (3.41) using $\theta = 1/2$ at all steps are again unstable. Unless very low accuracy is desired, solving the nonlinear equations with fully implicit timestepping may be the best choice for digital options with uncertain volatility.

Chapter 4

Two Factor Uncertain Volatility

For the second example of nonlinear pricing equations in this thesis, the one factor uncertain volatility model is extended to two factors. We find that the equations are more difficult to analyze theoretically, and the numerical scheme becomes much more complex. Since the underlying discrete equations no longer form an M-matrix (in general), the proof techniques of chapter 3 no longer apply. Further, the discrete equations can no longer be optimized simply by looking at the sign of gamma. At each timestep, for each node in the discrete grid, a small constrained optimization problem must be solved. Details of this optimization process are provided. Numerical examples show that the price spread implied by reasonable volatility ranges can be large. To reduce this spread, the application of static hedging is explored.

4.1 Background

Details of one factor uncertain volatility problems were provided in the previous chapter. Moving beyond one factor problems, the author in [53] examined uncertainty in multi-factor reverse convertible contracts. However, in all examples provided, the uncertainty always

reduced to a single parameter. Uncertain correlation for rainbow options was studied in [54], but again, only uncertainty in a single parameter was involved. To the author's knowledge, no previous study has examined uncertainty in more than one parameter for multi-factor problems.

Volatility parameters typically appear in the diffusion tensor of equation (2.3). When volatility is uncertain, the diffusion tensor is no longer unique. Hence, as with the one factor problem, equation (2.3) will not have a unique solution. Nevertheless, the maximal and minimal solution values of the contingent claim equation are unique. Mathematically, if $\sigma \in \Omega$ implies that the volatility values (perhaps one for each underlying factor) are constrained to lie within values specified by the region Ω, then we can uniquely solve

$$\text{Max price:} \qquad U_\tau = -\mathbf{V} \cdot \nabla U + \sup_{\sigma \in \Omega} ((\mathbf{D}\nabla) \cdot \nabla U) - rU \tag{4.1}$$

$$\text{Min price:} \qquad U_\tau = -\mathbf{V} \cdot \nabla U + \inf_{\sigma \in \Omega} ((\mathbf{D}\nabla) \cdot \nabla U) - rU. \tag{4.2}$$

Note that it is easy to see that equations (4.1) and (4.2) locally maximize or minimize the price. However, it is interesting to note that the local maximization/minimization also produces globally maximized/minimized solutions [6].

In the one factor case, the discrete equations could be optimized by simply examining the sign of gamma. With two or more factors, a small constrained optimization problem must be solved at each node and each timestep to determine the appropriate volatility values. The optimization problem will depend on the second derivative values. However, as will be discussed below, one must be careful in how the second derivative values are computed. The optimization must treat derivative values consistently with the discrete equations.

For concreteness, the effect of uncertain volatility on two factor European options will be studied. Modifications for American or more exotic options are minor. For standard European options, volatility terms only appear in the diffusion tensor. For vanilla options

on two assets, the diffusion tensor is symmetric positive semi-definite, and has the form

$$\mathbf{D} = \frac{1}{2}\begin{bmatrix} \sigma_1^2 S_1^2 & \rho\sigma_1\sigma_2 S_1 S_2 \\ \rho\sigma_1\sigma_2 S_1 S_2 & \sigma_2^2 S_2^2 \end{bmatrix}. \tag{4.3}$$

The velocity vector in this case is

$$\mathbf{V} = -\begin{bmatrix} rS_1 \\ rS_2 \end{bmatrix}. \tag{4.4}$$

With uncertain volatility, we have

$$\begin{aligned} \sigma_{1,\min} &\leq \sigma_1 \leq \sigma_{1,\max} \\ \sigma_{2,\min} &\leq \sigma_2 \leq \sigma_{2,\max}. \end{aligned}$$

In addition, uncertain correlation between the two underlying assets is allowed:

$$\rho_{\min} \leq \rho \leq \rho_{\max}. \tag{4.5}$$

4.2 Discretization

For two factor uncertain volatility problems, the discrete equations (2.16) are used. The computation of σ_1, σ_2 and ρ for use in the convection and diffusion terms will be described in section 4.3.1. The boundary conditions to be used include

$$\begin{cases} U_\tau = \frac{1}{2}\sigma_2^2 S_2^2 \frac{\partial^2 U}{\partial S_2^2} + rS_2\frac{\partial U}{\partial S_2} - rU & \text{as } S_1 \to 0 \\ U_\tau = \frac{1}{2}\sigma_1^2 S_1^2 \frac{\partial^2 U}{\partial S_1^2} + rS_1\frac{\partial U}{\partial S_1} - rU & \text{as } S_2 \to 0 \\ U(S_1, S_2, \tau) = \text{payoff}(S_1, S_2) & \text{as } S_1, S_2 \to \infty \end{cases} \tag{4.6}$$

4.3 Theoretical Results

4.3.1 Optimization of Discrete Equations

When optimizing diffusion in equation (2.16), we are interested in the term

$$\sum_{j\in\eta_i}\gamma_{ij}\left(U_j^{n+1}-U_i^{n+1}\right) \tag{4.7}$$

(and the equivalent term at time level n).

Expanding equation (4.7) is tedious and not very enlightening. Further, when constructing the discrete equations, it is common to loop over the elements of the mesh. We can then consider the contribution to equation i by triangle (element) E containing node i. Let the remaining two nodes of triangle E be labeled as p and q. Finally, let A_E be the area of triangle E. After some tedious algebra, the contribution to the diffusion term in equation i by triangle E is:

$$\begin{aligned}
\text{contribution from } E \;=\; & -\sigma_1^2\frac{A_E S_1^2}{2}\frac{\partial N_i}{\partial S_1}\left(\frac{\partial N_p}{\partial S_1}(U_p-U_i)+\frac{\partial N_q}{\partial S_1}(U_q-U_i)\right)\\
& -\sigma_2^2\frac{A_E S_2^2}{2}\frac{\partial N_i}{\partial S_2}\left(\frac{\partial N_p}{\partial S_2}(U_p-U_i)+\frac{\partial N_q}{\partial S_2}(U_q-U_i)\right)\\
& -\rho\sigma_1\sigma_2\frac{A_E S_1 S_2}{2}\left(\frac{\partial N_i}{\partial S_1}\frac{\partial N_p}{\partial S_2}(U_p-U_i)+\frac{\partial N_i}{\partial S_2}\frac{\partial N_p}{\partial S_1}(U_p-U_i)\right.\\
& \qquad \left.+\frac{\partial N_i}{\partial S_1}\frac{\partial N_q}{\partial S_2}(U_q-U_i)+\frac{\partial N_i}{\partial S_2}\frac{\partial N_q}{\partial S_1}(U_q-U_i)\right).
\end{aligned}$$

In matrix notation, this can be represented by

$$\begin{bmatrix}\sigma_1 & \sigma_2\end{bmatrix}\begin{bmatrix}a_E & \rho c_E\\ \rho c_E & b_E\end{bmatrix}\begin{bmatrix}\sigma_1\\ \sigma_2\end{bmatrix} \tag{4.8}$$

where

$$\begin{aligned}
a_E &= \frac{A_E S_1^2}{2}\frac{\partial N_i}{\partial S_1}\left(\frac{\partial N_p}{\partial S_1}(U_p - U_i) + \frac{\partial N_q}{\partial S_1}(U_q - U_i)\right) \\
b_E &= \frac{A_E S_2^2}{2}\frac{\partial N_i}{\partial S_2}\left(\frac{\partial N_p}{\partial S_2}(U_p - U_i) + \frac{\partial N_q}{\partial S_2}(U_q - U_i)\right) \\
c_E &= -\frac{A_E S_1 S_2}{4}\left(\frac{\partial N_i}{\partial S_1}\frac{\partial N_p}{\partial S_2}(U_p - U_i) + \frac{\partial N_i}{\partial S_2}\frac{\partial N_p}{\partial S_1}(U_p - U_i)\right. \\
&\qquad \left. + \frac{\partial N_i}{\partial S_1}\frac{\partial N_q}{\partial S_2}(U_q - U_i) + \frac{\partial N_i}{\partial S_2}\frac{\partial N_q}{\partial S_1}(U_q - U_i)\right). \qquad (4.9)
\end{aligned}$$

To maximize or minimize the diffusion, we need to consider contributions from all triangles attached to node i. Hence, the optimization problem for node i has the form

$$\begin{bmatrix} \sigma_1 & \sigma_2 \end{bmatrix} \begin{bmatrix} \Sigma a_E & \rho\Sigma c_E \\ \rho\Sigma c_E & \Sigma b_E \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix}, \qquad (4.10)$$

where the sums are over all triangles attached to node i.

Unfortunately, it is hard to extract meaning from (4.10). To help understanding, we can compare (4.10) with an expansion of the original equation. This is done in the next section.

4.3.2 Connection Between Analytic and Discrete Forms

Equation (4.10) in the previous section provides little insight into the optimization problem. To gain further understanding, consider equation (2.4) directly. To maximize or minimize the option value, it suffices to maximize or minimize the diffusion term. To do this, use a forward difference approximation for the time derivative, expand the diffusion operator and rearrange terms to obtain

$$U_i^{n+1} = U_i^n + \Delta\tau\left(\frac{\sigma_1^2 S_1^2}{2}\frac{\partial^2 U}{\partial S_1^2} + \frac{\sigma_2^2 S_2^2}{2}\frac{\partial^2 U}{\partial S_2^2} + \rho S_1 S_2 \sigma_1 \sigma_2 \frac{\partial^2 U}{\partial S_1 \partial S_2}\right)_i^n + \text{other terms.} \quad (4.11)$$

The expression to optimize becomes

$$\frac{\sigma_1^2 S_1^2}{2}\frac{\partial^2 U}{\partial S_1^2} + \frac{\sigma_2^2 S_2^2}{2}\frac{\partial^2 U}{\partial S_2^2} + \rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 U}{\partial S_1 \partial S_2}. \tag{4.12}$$

By setting the derivative with respect to ρ equal to zero, we see that ρ must attain one of its extreme values. Which value depends on the sign of $\frac{\partial^2 U}{\partial S_1 \partial S_2}$ (assuming that S_1, S_2, σ_1 and σ_2 are all non-negative). For example, if this term is positive, set ρ to $\rho_{\max}$ when maximizing the option price. With the value of ρ decided, we are left with a constrained optimization problem of the following quadratic form

$$\begin{bmatrix} \sigma_1 & \sigma_2 \end{bmatrix} \begin{bmatrix} a & \rho c \\ \rho c & b \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix}, \tag{4.13}$$

where $a = \frac{S_1^2}{2}\frac{\partial^2 U}{\partial S_1^2}$, $b = \frac{S_2^2}{2}\frac{\partial^2 U}{\partial S_2^2}$ and $c = \frac{S_1 S_2}{2}\frac{\partial^2 U}{\partial S_1 \partial S_2}$.

It appears that second derivatives need to be computed to optimize the volatility values. However, a naive implementation of this approach will lead to inconsistent results. In obtaining equation (4.13), we have shown the steps required to optimize the analytic pricing equation. In practice, we should be numerically optimizing the discrete equations as in equation (4.10). Thus, if we were to select volatility values by numerically approximating the derivatives values in equation (4.13), the numerical derivative approximations must be consistent with the discretization.

In our case, the derivative approximation scheme outlined in [65] is consistent with the discretized equations (2.16). Specifically, the terms in the matrix of (4.10) are these derivative estimates (up to a constant factor related to the areas of surrounding triangles, proof omitted). Hence, we could use (4.13) to select the volatility values if we approximate the derivative values using the method of [65]. In general, any other derivative approximation method is inconsistent with our discretization. In practice, using another method would

mean that the discrete equations are not being maximized (minimized), and iterative solution methods of the nonlinear equations might oscillate between states. For this reason, it makes sense to optimize the actual discrete equations (4.10), instead of using derivative estimation techniques which are inconsistent with the method used to discretize the PDE.

4.3.3 Optimization Details

In both the theoretical and numerical cases, a quadratic expression with linear constraints needs to be optimized. Formally, we have

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{M} \mathbf{x} = \begin{bmatrix} \sigma_1 & \sigma_2 \end{bmatrix} \begin{bmatrix} a & c \\ c & b \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix} \tag{4.14}$$

subject to

$$\begin{aligned} \sigma_{1,\min} &\leq \sigma_1 \leq \sigma_{1,\max} \\ \sigma_{2,\min} &\leq \sigma_2 \leq \sigma_{2,\max}. \end{aligned}$$

Note that optimizing for ρ is assumed to have already occurred (and is embedded in c). For now we assume a minimization problem. With an objective function $f(\mathbf{x})$ of n variables, and m constraints of the form $g_i(\mathbf{x}) \leq b_i$, the Kuhn-Tucker conditions [49] are

$$\frac{\partial f(\mathbf{x})}{\partial x_j} - \sum_{i=1}^{i=m} \lambda_i \frac{\partial g_i(\mathbf{x})}{\partial x_j} = 0 \qquad (j = 1, 2, \ldots, n) \tag{4.15}$$

$$\lambda_i \left[b_i - g_i(\mathbf{x}) \right] = 0 \qquad (i = 1, 2, \ldots, m) \tag{4.16}$$

$$\lambda_i \geq 0 \qquad (i = 1, 2, \ldots, m). \tag{4.17}$$

Let $\mathbf{e}_1 = (1, 0)'$ and $\mathbf{e}_2 = (0, 1)'$. Applying conditions (4.15)-(4.17) to problem (4.14)

results in

$$2\mathbf{M}\mathbf{x} - \mathbf{e}_1\lambda_1 + \mathbf{e}_1\lambda_2 - \mathbf{e}_2\lambda_3 + \mathbf{e}_2\lambda_4 = \mathbf{0} \tag{4.18}$$

where

$$\begin{aligned}
\lambda_1 \geq 0 \quad &\text{and} \quad \lambda_1 = 0 \text{ if } \sigma_1 > \sigma_{1,\min} \\
\lambda_2 \geq 0 \quad &\text{and} \quad \lambda_2 = 0 \text{ if } \sigma_1 < \sigma_{1,\max} \\
\lambda_3 \geq 0 \quad &\text{and} \quad \lambda_3 = 0 \text{ if } \sigma_2 > \sigma_{2,\min} \\
\lambda_4 \geq 0 \quad &\text{and} \quad \lambda_4 = 0 \text{ if } \sigma_2 < \sigma_{2,\max}.
\end{aligned}$$

If the optimal value occurs in the interior of the region defined by the extreme volatility values, then $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 0$. We then have

$$\mathbf{M}\mathbf{x} = \mathbf{0}.$$

If $\mathbf{M}$ is non-singular, then $\mathbf{x} = (0, 0)$ is a solution. This will almost surely be an infeasible solution for option pricing problems. Hence, the optimal solution will be on the boundary. If $\mathbf{M}$ is singular, then the columns are linearly dependent. We must then check if the optimal solution lies on the line defined by $a\sigma_1 + c\sigma_2 = 0$.

In any case, we must check the four corners of the region, as well as the four edges. For example, if we assume that $\sigma_1 = \sigma_{1,\max}$, then we compute

$$\begin{aligned}
&\frac{\partial}{\partial \sigma_2}(a\sigma_{1,\max}^2 + b\sigma_2^2 + 2c\sigma_{1,\max}\sigma_2) = 0 \\
\rightarrow \quad &\sigma_2 = \frac{-c\sigma_{1,\max}}{b}.
\end{aligned}$$

Of course, we need to check if the computed σ_2 value is feasible, and compare to all previously computed optimal values. The remaining three edges are handled similarly. We also need to check "degenerate cases", such as when $a = 0, b = 0$ or $c = 0$. In these cases we can optimize by simply checking the signs of the remaining non-zero a, b or c.

if $\mathbf{M} = 0$ or $a = 0$ or $b = 0$ or $c = 0$ **then**
  Handle special case
else
  Initialize optimal value ($+\infty$ for min and $-\infty$ for max)
  for $\sigma_1 \in \{\sigma_{1,max}, \sigma_{1,min}\}$ **do**
    Compute optimal σ_2
    if computed σ_2 is feasible **then**
      Compare to optimal value so far
    else
      Compute values for $\sigma_2 \in \{\sigma_{2,min}, \sigma_{2,max}\}$
      Compare to optimal value so far
    end if
  end for
  for $\sigma_2 \in \{\sigma_{2,max}, \sigma_{2,min}\}$ **do**
    Compute optimal σ_1
    if computed σ_1 is feasible **then**
      Compare to optimal value so far
    end if
  end for
end if

ALGORITHM 4.1: *Method for computing the optimal two factor volatility values.*

Algorithmically, the overall process can be done as in algorithm (4.1) Note that corners are checked in the first for loop of the algorithm, and need not be checked again. Also, comparing optimal values in the algorithm means to compare final values from evaluations of equation (4.14).

4.4 Numerical Results

Our first test is for a simple European call option on the max of two assets, with a payoff of $\max(\max(S_1, S_2) - K, 0)$, where K is the strike price. All model parameters are given in Table 4.1. By default, a long position is assumed. Also by default, the frozen coefficient iterative scheme with tolerance of 10^{-6} was used to solve the nonlinear equations at each timestep. As will be shown below, Newton's method and the non-smooth Newton method

Type	Call (best-case long)
Payoff	$\max(\max(S_1, S_2) - K, 0)$
Time to expiry	0.5
r	0.05
strike - K	40
$\sigma_{1,\min}$	0.3
$\sigma_{1,\max}$	0.5
$\sigma_{2,\min}$	0.3
$\sigma_{2,\max}$	0.5
$\rho_{\min}$	0.3
$\rho_{\max}$	0.5

TABLE 4.1: *Model parameters for the max of two asset call option (best case long).*

often exhibited convergence difficulties, and are therefore not used for regular testing.

Convergence results for at-the-money values ($S_1 = S_2 = 40$) are given in Table 4.2. From the table, we see that both implicit timestepping and Rannacher timestepping are converging to the same solution. By Rannacher timestepping, we mean Crank-Nicolson timestepping preceded by 2 fully implicit steps [46]. From the results of chapter 3, we expect this solution to be the correct viscosity solution, although we have no theoretical proof. Rannacher timestepping converges at a quadratic rate, while implicit timestepping appears to be dropping to a linear rate. Crank-Nicolson timestepping is either converging to a non-viscosity solution, or has a slightly growing instability.

Extrapolating the results from Table 4.2 gives a best-case value of 9.937. Similarly, we can obtain a worst-case value of 5.831. Clearly, the spread between the best-case and worst-case prices is relatively large. Even though the volatility and correlation values can vary by a maximum of 20 percent, the prices vary by over 40 percent.

We can also compute max and min prices for constant volatility and correlation values for the same model parameters. Specifically, the max price was computed with $\sigma_1 = \sigma_{1,\max}$, $\sigma_2 = \sigma_{2,\max}$ and $\rho = \rho_{\min}$, whereas the min price was computed with $\sigma_1 = \sigma_{1,\min}, \sigma_2 = \sigma_{2,\min}$ and $\rho = \rho_{\max}$. Note that since U_{xy} is non-positive (for constant coefficient problems), the

	Fully Implicit			Crank-Nicolson			Rannacher		
Nodes	Value	Diff	Ratio	Value	Diff	Ratio	Value	Diff	Ratio
41×41	9.8172			10.042			9.8571		
81×81	9.8946	0.0774		10.247	0.205		9.9163	0.0592	
161×161	9.9205	0.0259	2.98	10.369	0.123	1.67	9.9318	0.0155	3.82
321×321	9.9300	0.0095	2.73	10.445	0.076	1.61	9.9357	0.0039	3.94

TABLE 4.2: *Convergence results for an at-the-money* ($S_1 = S_2 = 40$) *call option with parameters as given in Table 4.1. The initial timestep of 0.02 is halved at each grid refinement. "Diff" is the absolute value of the change in the solution as the grid is refined. "Ratio" is the ratio of successive differences.*

correlation is set to the smallest value to obtain the largest option price. Conversely, it is set to the largest possible value to get the smallest price. The max price was found to be 9.937 while the min price was 5.831 – the same values as for the worst-case and best-case pricing.

Such a result is not immediately obvious, since even though U_{xx} and U_{yy} are both non-negative, U_{xy} is non-positive for a regular call option on the max of two assets. Hence, maximizing or minimizing (4.12) is not necessarily trivial, although in this case it turns out that the same volatility and correlation values should be chosen (in theory at least). Further, it is important to note that the numerical scheme for the uncertain case did not always set the volatility and correlation values to the same levels as for the max and min pricing values at all nodes for each timestep. Due to the discrete form of the equations, the optimization parameters given by (4.10) often had different signs than would be expected from (4.12). In other words, the numerical second derivative values sometimes had a different sign than what theory would predict. However, by optimizing the discrete equations, the problem converges to the correct solution. Had we approximated the derivative values by some other method (potentially more accurate), the discrete equations would not be properly optimized. In fact, doing so often caused the iterative method used to solve the nonlinear discrete equations to oscillate indefinitely between two (or more) states.

The above test was repeated using Newton iteration and a non-smooth Newton iteration

to solve the nonlinear discrete equations at each timestep. For these tests, a direct method was used to solve the intermediate linear systems (an iterative solver sometimes had difficulty converging). Further, if convergence had not occurred after 50 nonlinear iterations, the timestep was cut in half. If convergence still had not occurred after cutting the timestep 4 times, the process was terminated, and the method deemed not to have converged. As can be seen by the total iteration counts in table 4.3, the frozen coefficient method worked well (at least for fully implicit and Rannacher timestepping), while Newton's method and the non-smooth Newton method behaved poorly.

Examining the columns for the frozen coefficient method in Table 4.3, we see that the Crank-Nicolson method has problems. Both implicit and Rannacher timestepping require just over two iterations on average. This indicates a rather mild nonlinearity. Nevertheless, tests using the full (numerical Jacobian) Newton method or the non-smooth Newton method struggle to converge. In the case of the full Newton method, convergence only occurred for the first grid (strictly speaking, convergence also occurred for the second grid for Crank-Nicolson, but required an enormous number of iterations due to a large number of timestep reductions). The non-smooth Newton method fared a little better, but still failed to converge for the final grid. Also, the non-smooth method required significantly more iterations than the frozen coefficient method (excluding Crank-Nicolson), especially as the grid is refined. Hence, the frozen coefficient method is recommended for solving two factor uncertain volatility problems.

Our next test keeps the same parameters as in Table 4.1, except that the payoff has been changed to a butterfly. In particular, the payoff is

$$\begin{aligned} S_{max} &= \max(S_1, S_2) \\ U(S_1, S_2, \tau = 0) &= \max(S_{max} - K_1, 0) + \max(S_{max} - K_2, 0) \\ &\quad -2\max(S_{max} - (K_1 + K_2)/2, 0) \ , \end{aligned} \tag{4.19}$$

	Frozen Coefficients			Newton			Non-smooth Newton		
Timesteps	Implicit	C-N	Rann.	Implicit	C-N	Rann.	Implicit	C-N	Rann.
25	69	108	66	243	310	361	74	108	75
50	121	252	120	—	60440	—	163	255	156
100	214	547	209	—	—	—	429	560	419
200	411	1131	407	—	—	—	—	—	—

TABLE 4.3: *Total iteration counts for the max of two asset call option test problem. C-N stands for Crank-Nicolson, while Rann stands for Rannacher. Parameters are provided in Table 4.1. Note that timesteps is the initial number of constant timesteps required, but since the method can reduce the timestep to help with convergence, it is not necessarily the actual number of timesteps taken.*

	Fully Implicit			Crank-Nicolson			Rannacher		
Nodes	Value	Diff	Ratio	Value	Diff	Ratio	Value	Diff	Ratio
41×41	1.64661			2.354			1.63913		
81×81	1.65254	0.00592		2.612	0.258		1.64997	0.01083	
161×161	1.65544	0.00291	2.03	2.790	0.179	1.44	1.65507	0.00510	2.12
321×321	1.65709	0.00165	1.76	2.908	0.118	1.52	1.65744	0.00237	2.16

TABLE 4.4: *Convergence results for an at-the-money ($S_1 = S_2 = 40$) butterfly spread call option with parameters as given in Table 4.1 and payoff specified by equation* (4.19)*. The initial timestep of 0.02 is halved at each grid refinement. "Diff" is the absolute value of the change in the solution as the grid is refined. "Ratio" is the ratio of successive differences.*

where we have set $K_1 = 35$ and $K_2 = 45$. In some sense, this test is more challenging, since the signs of U_{xx} and U_{yy} change over the solution domain. Convergence results using the frozen coefficient method for a best-case pricing are given in Table 4.4.

From Table 4.4, we see that implicit timestepping is converging to the (presumably correct) answer at a linear rate. Rannacher timestepping is converging to the same solution at what appears to be a linear rate as well. Crank-Nicolson timestepping is either converging to a non-viscosity solution, or has a slightly growing instability. These results are consistent with those seen for one factor uncertain volatility in chapter 3, and will be shown to be consistent with passport options in chapter 5.

Iteration counts for the frozen coefficient method are shown in Table 4.5 for the butterfly test problem. The trends are the same as in Table 4.3, although implicit and Rannacher timestepping tend to require about 3 iterations on average, instead of only 2 (as in Table 4.3. This is a direct result of the butterfly being a "harder" payoff. By having sign changes

Timesteps	Implicit	Crank-Nicolson	Rannacher
25	90	140	88
50	171	288	164
100	320	580	316
200	608	1162	593

TABLE 4.5: *Total iteration counts for the butterfly payoff test problem. Parameters (except for the payoff) are provided in Table 4.1.*

Test	Value
Uncertain best-case	1.66
$\sigma_1 = 0.3, \sigma_2 = 0.3, \rho = 0.3$	1.102
$\sigma_1 = 0.3, \sigma_2 = 0.3, \rho = 0.5$	1.098
$\sigma_1 = 0.4, \sigma_2 = 0.4, \rho = 0.4$	0.864
$\sigma_1 = 0.5, \sigma_2 = 0.5, \rho = 0.3$	0.716
$\sigma_1 = 0.5, \sigma_2 = 0.5, \rho = 0.5$	0.708
Uncertain worst-case	0.336

TABLE 4.6: *Option values at* $(S_1, S_2) = (40, 40)$ *for various parameter choices with a butterfly payoff (long position).*

in delta and gamma, there is more variation in the optimal volatility and correlation values.

As indicated by equation (4.19), a butterfly payoff can usually be considered as the sum of 4 separate call options: long one call at strike K_1, short two calls at strike $(K_1 + K_2)/2$ and long one call at strike K_2. However, with uncertain parameters, the equations are nonlinear. Hence, the sum of these 4 call options is not the same as an option with the payoff given by equation (4.19). For comparison, Table 4.6 gives prices of butterfly options using fixed volatility and correlation values. We see that the uncertain values form an envelope around the fixed parameter prices.

4.5 Application to Static Hedging

Since the best-case long price corresponds to the worst-case short price, the best-case and worst-case prices correspond to the bid-ask spread if option buyers and sellers price options using worst-case scenarios (from their own points of view). In our first example where a max

of two asset call option was priced, we saw that worst-case and best-case pricing leads to a rather large spread in prices (9.937 versus 5.831). Arguably, the spread is so large as to make the uncertain volatility method somewhat useless for option pricing, unless tight bounds on volatility and correlation ranges can be obtained. To address this issue, the Lagrangian uncertain volatility model was introduced in [7]. In this model, optimal static hedges are computed for the given problem using predetermined options with known market prices. The residual payoff is then priced using the uncertain volatility model. Since the magnitude of the residual is generally much less than the magnitude of the original payoff, computed spreads are much smaller. In this way, the effective (or implied) range of volatility values is reduced. Note that for linear problems, static hedging cannot reduce the cost of the option (which is, of course, the initial cost of the hedging portfolio). However, for nonlinear problems, it may be possible to reduce the cost of the option by static hedging.

In our brief review of the Lagrangian uncertain model, we follow the notation of [59]. Assume that n options will be used to form the static hedge, and that market prices C_i exist for all n of these options. For simplicity, we will ignore bid-ask spreads in C_i, and assume that all options under consideration expire at the same time. Let $\Lambda(S)$ be the payoff of the option we are trying to price, and let $\Lambda_i(S)$ be the payoffs of the options used in the static hedge. If the static hedge is formed by holding λ_i of each underlying option, then the residual payoff is given by

$$U(S,T) = \Lambda(S) - \sum_i \lambda_i \Lambda_i(S). \tag{4.20}$$

In general, $U(S,T)$ will not be a convex function, and the full nonlinear uncertain volatility model must be used for pricing. For example, even if $\Lambda(S)$ is convex, the addition of different options for static hedging may make the net result non-convex.

Denote the best-case price of the hedged contract today as $U^+(S_0,0)$, and the worst-

case price as $U^-(S_0, 0)$. The optimal static hedge using best-case pricing is then found by solving

$$\min_{\lambda_i} \left(U^+(S_0, 0) + \sum_i \lambda_i C_i \right). \tag{4.21}$$

Moreover, the final value of equation (4.21) is the final option price. Similarly, the optimal static hedge for worst-case pricing is found by solving

$$\max_{\lambda_i} \left(U^-(S_0, 0) + \sum_i \lambda_i C_i \right). \tag{4.22}$$

In testing below, optimization routines from the NAG library have been used to solve equations (4.21) and (4.22). Prices for $U^+(S_0, 0)$ and $U^-(S_0, 0)$ were determined using the methods described in previous sections.

As a first test of the Lagrangian uncertain model, we recalculate prices for the max of two asset call option problem described in Table 4.1. The static hedge will be composed of two max of two asset call options expiring in 6 months ($T = 0.5$), with strikes of 35 and 45 respectively. The residual payoff is then defined by

$$\begin{aligned} U(S, T) &= \max(\max(S_1, S_2) - 40, 0) \\ &\quad - \left[\lambda_1 \max(\max(S_1, S_2) - 35, 0) + \lambda_2 \max(\max(S_1, S_2) - 45, 0) \right]. \end{aligned} \tag{4.23}$$

For a best-case pricing, the task is to find the λ_1 and λ_2 such that equation (4.21) is minimized using payoff (4.23). Similarly, we find the λ_1 and λ_2 that maximize equation (4.22) using payoff (4.23) for a worst-case pricing. The C_i values for the hedging options were taken to be 11.48 and 5.04 respectively. These are simply the usual Black-Scholes prices using parameters $\sigma_1 = 0.4, \sigma_2 = 0.4, \rho = 0.4$ and $r = 0.05$. One can think of 11.48 and 5.04 as market prices with implied volatility values as given. From a model point of view, it is important that the implied volatility values are within the bounds of the uncertain

Method	Best-case	Worst-case	Spread
No Hedging	9.937	5.831	4.11
Optimal Hedge 1 (two factor options)	8.08	7.48	0.60
Optimal Hedge 2 (one factor options)	8.64	7.00	1.64

TABLE 4.7: *Spread values for uncertain volatility pricing for a max of two asset call option without static hedging, and by statically hedging with max of two asset call options (hedge 1) and with one asset call options (hedge 2).*

parameters to be used for pricing. If they were not, the optimal static hedging model would admit arbitrage [7].

Solving equation (4.21) gives a best-case price of 8.08. The optimal static hedge position is to buy 0.53 of the 35-strike call and 0.49 of the 45-strike call. Solving equation (4.22) gives the worst-case price of 7.48, using positions of 0.62 in the 35-strike call and 0.45 in the 45-strike call. Hence, by statically hedging the original option, the spread of prices using uncertain volatilities has been reduced from 9.937-5.831 to 8.08-7.48. Clearly, the spread has been greatly reduced, and gives a much more realistic estimate of the risk involved in buying/selling the original contract. These spread values are summarized in Table 4.7.

The payoff of the original call option and the (negative of the) residual of the statically hedged option (best-case long) are shown in Figure 4.1 (the negative of the residual is shown for clarity). Note how the magnitude of the payoff has been significantly reduced with static hedging. Also note that the residual payoff is non-convex, meaning that the uncertain pricing is non-trivial.

While the reduction in the spread is hopeful, two factor options may not exist in the market to be used as hedging instruments. Hence, we repeat the experiment, except that this time we hedge with two 1-factor options: an option on asset 1 with strike 40, and an option on asset 2 with strike 40. The prices of these one factor options have been set to 4.95, which corresponds to an implied volatility value of 0.40. In this case, solving equation (4.21) gives a best-case price of 8.64 (positions of 0.936 in both hedges), while solving equation

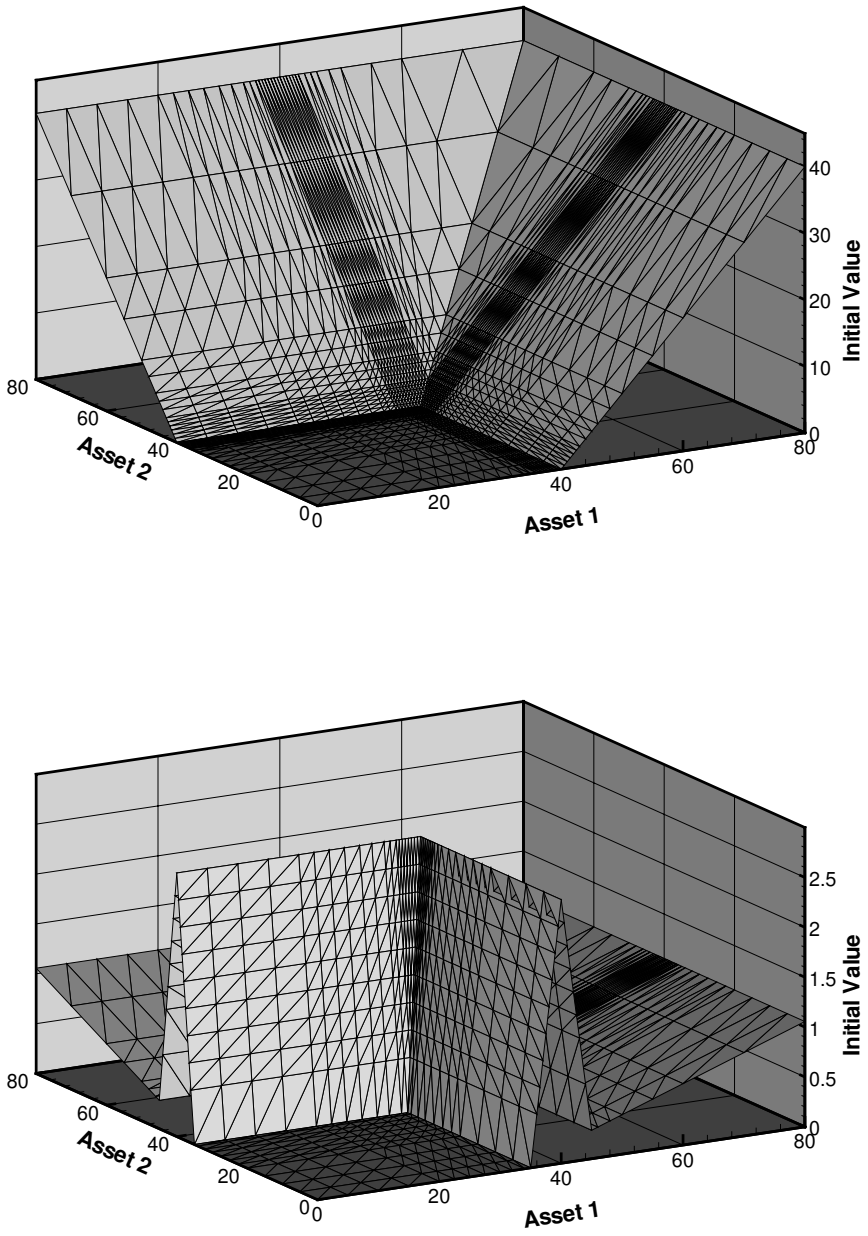


FIGURE 4.1: *Final payoff of unhedged option (top) and the residual payoff of the (best-case) optimally hedged option (bottom).*

(4.22) gives a worst-case price of 7.00 (position of 0.77 in both hedges). As expected, the spread is larger than when using two factor options to hedge. However, the spread is still much smaller than for the unhedged option.

Chapter 5

Passport Options

The final example in this thesis of a nonlinear option pricing equation involves passport options. The primary difference between passport options and the uncertain volatility models of previous chapters is the appearance of a nonlinear convection term. Since the discretization of the convection term can change (forward, backward, central, upstream), theoretical problems arise. In particular, a positive coefficient scheme cannot be guaranteed for one factor problems with non-convex payoffs. Nevertheless, numerical experiments demonstrate behaviour consistent with the uncertain volatility results. The chapter concludes with an application of passport option theory to the problem of trader compensation.

5.1 Background

While standard options are based on stock prices and/or interest rates, passport options are based on the performance of a trading account. Specifically, the holder of a passport option makes trades in an underlying asset for a specified period of time, taking both long and short positions as desired. At exercise, the investor may keep any net trading gain, while any loss is covered.

Passport options were first described in [31]. Since that article several studies have examined a variety of passport option properties. The use of partial differential equation (PDE) methods to price passport options is described in [3] and [52]. A detailed study of American passport options is given in [11]. The robustness of the optimal trading strategy with respect to the underlying model is analyzed in [26] by adding stochastic volatility. In [2] and [42], a variety of exotic features are discussed and priced, including passport options on multiple assets. A variation of passport options, which the authors call vacation calls and vacation puts, is tackled by probabilistic methods in [47]. Probabilistic methods are also used in [17] to relate the theory of passport options to general principles from martingale theory. The theory of passport options has also been applied to different areas. For example, commodity hedging strategies based on passport options are described in [27], while a connection to Asian options is made in [57]. Passport option theory was also used to examine trader bonus issues in [1]. The reader should note that little, if any, reference is made to viscosity solution issues in previous work that uses the numerical PDE approach.

The mechanics of a vanilla passport option are relatively straight forward. At inception, the buyer of the option pays the seller an up-front premium. The buyer also specifies an initial position - long or short q units - in the underlying asset. While fractional positions are allowed, the maximum position is usually limited to some fixed amount. During the life of the contract, the buyer can update his or her position in the underlying asset at any time. Note that the buyer does not actually trade the underlying asset. Rather, the desired position is simply communicated to the seller. At expiry, the net balance of the trading account w implied by the trading strategy $q(t)$ is determined. If the implied trading account is positive, that amount is paid back to the buyer. If the implied amount is negative, the buyer walks away with no liability. For example, assume the initial asset price is \$100. The buyer initially adopts a long position ($q = 1$), and holds this position until expiry. Ignoring the effects of dividends and interest rates, if the asset price is \$110 at expiry, then the seller

pays \$10 to the buyer. If the asset price had fallen to \$90, then no cash flow occurs.

Of course, the seller of the option can use the up-front premium to hedge the contract. The price of a passport option (the premium) is therefore determined by the trading strategy which gives the worst-case for the seller of the option. Since the pricing assumes a worst-case scenario, the seller's hedging account is expected to have a non-negative balance. In practice, the effect of dividends and interest rates must also be specified. For example, it is common for dividends to be paid to the buyer of the option, for the balance of the trading account to grow at the risk free rate, and for the position in the underlying asset to be financed at some cost of carry rate (such as the risk free rate minus the dividend rate). Other contract possibilities are described in [17].

As usual for option pricing problems, the underlying asset is assumed to follow a geometric Brownian motion process given by

$$dS = \mu S dt + \sigma S dW, \tag{5.1}$$

where S is the underlying asset price, μ is the expected drift rate, σ is the (constant) asset volatility, t is time, and dW is the increment of a Wiener process. Following the approach of [31], the PDE for the fair price of a passport option is then

$$\begin{aligned} -\frac{\partial U^q}{\partial t} = & -rU^q + (r-\gamma)S\frac{\partial U^q}{\partial S} - \left((\gamma - r + r_c)qS - r_t w\right)\frac{\partial U^q}{\partial w} \\ & + \frac{\sigma^2 S^2}{2}\left(\frac{\partial^2 U^q}{\partial S^2} + 2q\frac{\partial^2 U^q}{\partial S \partial w} + q^2\frac{\partial^2 U^q}{\partial w^2}\right), \end{aligned} \tag{5.2}$$

where

w is the accumulated wealth of the underlying trading account, and $-\infty \leq w \leq \infty$

r is the risk-free interest rate

γ is a dividend rate on the underlying asset S

r_c is a cost of carry rate

r_t is an interest rate for the trading account

t is time (forward direction)

$q(t)$ represents a trading strategy - investor holds q units at price $S(t)$ at time t.

The option value is written as U^q to denote the explicit dependence on the unspecified trading strategy $q(t)$. The usual payoff from a passport option at expiry time $t = T$ is simply

$$U^q(S, w, t = T) = \max(w, 0). \tag{5.3}$$

In other words, the investor keeps any accumulated wealth from trading, but is forgiven any loss. Of course, more exotic payoff functions are possible. The boundary conditions suggested by [52] include

$$\begin{aligned} U^q(0, w, t) &= \max(0, w) \\ \frac{\partial^2 U^q(\infty, w, t)}{\partial S^2} &= 0 \\ U^q(S, -\infty, t) &= 0 \\ U^q(S, \infty, t) &= w. \end{aligned}$$

For computational purposes, a finite domain is used. In general, setting U_{SS} to zero at the boundary will not lead to a positive coefficient scheme [62]. However, it must be remembered that a positive coefficient scheme cannot be obtained for general two factor problems with a non-zero cross-derivative term [68]. Hence, we have used the $U_{SS} = 0$ boundary condition for all of our two factor passport option tests. For further discussion of this boundary condition, see [62].

For all examples to be examined, the trading strategy q is limited to $|q(t)| \leq 1 \, \forall t$.

Different position limits can be handled by scaling [31]. When q is positive, the investor has a long position in S. When q is negative, the investor has gone short. By appealing to the principles of dynamic programming, a Hamilton-Jacobi-Bellman equation can be constructed from equation (5.2), giving

$$-\frac{\partial U}{\partial t} = -rU + (r-\gamma)S\frac{\partial U}{\partial S} + \max_{|q|\leq 1}\left\{-\left((\gamma - r + r_c)qS - r_t w\right)\frac{\partial U}{\partial w} + \frac{\sigma^2 S^2}{2}\left(\frac{\partial^2 U}{\partial S^2} + 2q\frac{\partial^2 U}{\partial S \partial w} + q^2\frac{\partial^2 U}{\partial w^2}\right)\right\}. \quad (5.4)$$

Solving this equation maximizes the value of the option by determining the optimal trading strategy q^* (much like solving an American option determines the optimal exercise boundary). Here, optimal implies the worst-case for a writer of the option. The values of q selected will not necessarily be optimal for the holder of the option.

The similarity reduction $U(t,S,w) = Su(t,w/S)$ reduces equation (5.4) to

$$-\frac{\partial u}{\partial t} = -\gamma u + \max_{|q|\leq 1}\left\{\left((r-\gamma-r_c)q - (r-\gamma-r_t)x\right)\frac{\partial u}{\partial x} + \frac{\sigma^2}{2}(x-q)^2\frac{\partial^2 u}{\partial x^2}\right\}, \quad (5.5)$$

where $x = w/S, -\infty \leq x \leq \infty$. Note that a similarity reduction must also be consistent with the PDE initial conditions (and boundary conditions). In the case of passport options, the payoff specified by equation (5.3) is consistent with $x = w/S$. The standard payoff for one factor problems then becomes

$$u(x, t = T) = \max(x, 0). \quad (5.6)$$

Boundary conditions for this payoff can then be

$$\begin{aligned} u(x \to -\infty, t) &= 0 \\ u(x \to \infty, t) &= x. \end{aligned}$$

For convex payoff functions, such as equation (5.6), it is always optimal to go long or short the maximal amount allowed by the position limits [3]. In this case equation (5.5) can be rewritten as

$$-\frac{\partial u}{\partial t} = -\gamma u + ((r-\gamma-r_c)q^* - (r-\gamma-r_t)x)\frac{\partial u}{\partial x} + \frac{\sigma^2}{2}(x-q^*)^2\frac{\partial^2 u}{\partial x^2}, \tag{5.7}$$

where

$$q^* = \text{ sgn}\left((r-\gamma-r_c)\frac{\partial u}{\partial x} - \sigma^2 x\frac{\partial^2 u}{\partial x^2}\right). \tag{5.8}$$

Equation (5.7) can also be written as

$$-\frac{\partial u}{\partial t} = -\gamma u + (-(r-\gamma-r_t)x)\frac{\partial u}{\partial x} + \frac{\sigma^2}{2}(x^2+1)\frac{\partial^2 u}{\partial x^2} + \left|(r-\gamma-r_c)\frac{\partial u}{\partial x} - \sigma^2 x\frac{\partial^2 u}{\partial x^2}\right|. \tag{5.9}$$

For general payoff functions, equation (5.7) remains the same, but the control specified by equation (5.8) becomes

$$q^* = \begin{cases} \psi & \text{if } \psi(x,t) \in [-1,1] \text{ and } \frac{\partial^2 u}{\partial x^2} < 0 \\ \text{sgn}\left(-\psi\frac{\partial^2 u}{\partial x^2}\right) & \text{otherwise} \end{cases} \tag{5.10}$$

where

$$\psi(x,t) = x - \frac{r-\gamma-r_c}{\sigma^2}\frac{\frac{\partial u}{\partial x}}{\frac{\partial^2 u}{\partial x^2}}. \tag{5.11}$$

Note that this control appeared incorrectly in [3], and was corrected in [52]. Equations (5.10)-(5.11) can be derived by differentiating the max expression in equation (5.5) with respect to q, setting the result equal to zero, and solving for q. Also note that equation (5.10) reduces to equation (5.8) when U_{xx} is positive (convex option/payoff values). For general payoff functions, if it is assumed that the solution is asymptotically linear, the boundary condition $u_{xx} = 0$ as $x \to \pm\infty$ can be imposed.

5.2 Discretization

Assuming the passport option payoff admits a similarity transformation, equation (5.7) needs to be discretized. The result is equation (2.11). If central differences are used for the first derivative term then

$$\begin{aligned}
\alpha^n_{i,central} &= \left[\frac{\sigma^2}{2}\frac{2(x_i - q_i^*)^2}{\Delta x_i^-(\Delta x_i^+ + \Delta x_i^-)} - \frac{r_1 q_i^* - r_2 x_i}{\Delta x_i^+ + \Delta x_i^-}\right]\Delta\tau \\
\beta^n_{i,central} &= \left[\frac{\sigma^2}{2}\frac{2(x_i - q_i^*)^2}{\Delta x_i^+(\Delta x_i^+ + \Delta x_i^-)} + \frac{r_1 q_i^* - r_2 x_i}{\Delta x_i^+ + \Delta x_i^-}\right]\Delta\tau \\
\psi_{i,central} &= x_i - \frac{r_1}{2\sigma^2}\frac{(U_{i+1} - U_{i-1})(\Delta x_i^+)(\Delta x_i^-)}{\Delta x_i^-(U_{i+1} - U_i) + \Delta x_i^+(U_{i-1} - U_i)}
\end{aligned} \tag{5.12}$$

where for notational convenience we have set

$$r_1 = r - \gamma - r_c$$

$$r_2 = r - \gamma - r_t.$$

Since central differencing is second order accurate, its use is preferred whenever possible. However, if $\alpha_{i,central}$ or $\beta_{i,central}$ is negative (note that r_1, r_2 and x_i can be negative), we can revert to first order accurate forward or backward differencing at the problem nodes. For forward differencing, this leads to

$$\begin{aligned}
\alpha^n_{i,forward} &= \left[\frac{\sigma^2}{2}\frac{2(x_i - q_i^*)^2}{\Delta x_i^-(\Delta x_i^+ + \Delta x_i^-)}\right]\Delta\tau \\
\beta^n_{i,forward} &= \left[\frac{\sigma^2}{2}\frac{2(x_i - q_i^*)^2}{\Delta x_i^+(\Delta x_i^+ + \Delta x_i^-)} + \frac{(r_1 q_i^* - r_2 x_i)}{\Delta x_i^+}\right]\Delta\tau \\
\psi_{i,forward} &= x_i - \frac{r_1}{2\sigma^2}\frac{(U_{i+1} - U_i)(\Delta x_i^+ + \Delta x_i^-)(\Delta x_i^-)}{\Delta x_i^-(U_{i+1} - U_i) + \Delta x_i^+(U_{i-1} - U_i)}.
\end{aligned} \tag{5.13}$$

$q^*_{i,central}$ = Equation (5.10) using $\psi_{i,central}$
$q^*_{i,forward}$ = Equation (5.10) using $\psi_{i,forward}$
$q^*_{i,backward}$ = Equation (5.10) using $\psi_{i,backward}$
if $\frac{\sigma^2}{2}\frac{2(x_i-q^*_{i,central})^2}{\Delta x_i^-(\Delta x_i^+ + \Delta x_i^-)} - \frac{r_1 q^*_{i,central} - r_2 x_i}{\Delta x_i^+ + \Delta x_i^-} \geq 0$ and $\frac{\sigma^2}{2}\frac{2(x_i-q^*_{i,central})^2}{\Delta x_i^+(\Delta x_i^+ + \Delta x_i^-)} + \frac{r_1 q^*_{i,central} - r_2 x_i}{\Delta x_i^+ + \Delta x_i^-} \geq 0$ **then**
  $\alpha_i = \alpha_{i,central}$
  $\beta_i = \beta_{i,central}$
else if $\frac{\sigma^2}{2}\frac{2(x_i-q^*_{i,forward})^2}{\Delta x_i^-(\Delta x_i^+ + \Delta x_i^-)} + \frac{r_1 q^*_{i,forward} - r_2 x_i}{\Delta x_i^+} \geq 0$ **then**
  $\alpha_i = \alpha_{i,forward}$
  $\beta_i = \beta_{i,forward}$
else
  $\alpha_i = \alpha_{i,backward}$
  $\beta_i = \beta_{i,backward}$
end if

ALGORITHM 5.1: *Deciding between a central, forward or backward discretization for one factor passport options.*

Similarly, backwards differencing at a problem node gives

$$\begin{aligned}
\alpha^n_{i,backward} &= \left[\frac{\sigma^2}{2}\frac{2(x_i - q_i^*)^2}{\Delta x_i^-(\Delta x_i^+ + \Delta x_i^-)} - \frac{(r_1 q_i^* - r_2 x_i)}{\Delta x_i^-}\right]\Delta\tau \\
\beta^n_{i,backward} &= \left[\frac{\sigma^2}{2}\frac{2(x_i - q_i^*)^2}{\Delta x_i^+(\Delta x_i^+ + \Delta x_i^-)}\right]\Delta\tau \\
\psi_{i,backward} &= x_i - \frac{r_1}{2\sigma^2}\frac{(U_i - U_{i-1})(\Delta x_i^+ + \Delta x_i^-)(\Delta x_i^+)}{\Delta x_i^-(U_{i+1} - U_i) + \Delta x_i^+(U_{i-1} - U_i)}.
\end{aligned} \tag{5.14}$$

Algorithmically, the type of differencing for equation (2.11) is decided by algorithm (5.1).

For many payoffs, or when the basic passport option is extended, a similarity reduction is not possible. In this case, equation (5.4) must be solved directly. For this task, the discretization given by equation (2.16) can be used. The diffusion term is specified by

$$\mathbf{D} = \frac{\sigma^2 S^2}{2}\begin{bmatrix} 1 & q \\ q & 1 \end{bmatrix}, \tag{5.15}$$

while the velocity vector becomes

$$\mathbf{V} = - \begin{bmatrix} (r-\gamma)S \\ (\gamma - r + r_c)qS + r_t w \end{bmatrix}. \tag{5.16}$$

As with the two factor uncertain volatility problems, care must be taken when maximizing the discrete equations with respect to the optimal holding strategy. Specifically, the expression to maximize is

$$\max_{|q|\leq 1} \left\{ - \left((\gamma - r + r_c)qS - r_t w \right) \frac{\partial U^q}{\partial w} + \frac{\sigma^2 S^2}{2} \left(\frac{\partial^2 U^q}{\partial S^2} + 2q \frac{\partial^2 U^q}{\partial S \partial w} + q^2 \frac{\partial^2 U^q}{\partial w^2} \right) \right\}. \tag{5.17}$$

Maximizing this expression by the standard approach of setting the derivative with respect to q equal to zero gives

$$q = \frac{r_1 S U_w - \sigma^2 S^2 U_{sw}}{\sigma^2 S^2 U_{ww}}. \tag{5.18}$$

For the derivative values (U_w, U_{sw}, U_{ww}), we must use the discrete forms as given by the finite volume discretization. In other words, we must maximize the discrete equations, not the analytic form presented in equation (5.18). It is tempting to call an external routine which provides accurate derivative estimates (such as the superconvergent patch recovery method described in [64], or higher order Taylor series expansion methods). However, in our experience, doing so often causes the iterative solution techniques described in section 2.5 to continuously alternate between states, since the computed values of q are inconsistent with the discretized algebraic equations. As shown in chapter 4, similar observations were made for two factor uncertain volatility problems.

5.3 Theoretical Results

Strictly speaking, algorithm (5.1) does not guarantee that α_i and β_i will be positive in the one factor discretization. The basic problem involves computing q_i such that the discrete equations are maximized. This means being consistent with discrete derivative values for both equation (2.11) and equation (5.10). Hence, when we switch from central to forward or backward differencing at a problem node, the value of q_i can change. The sign of the second (convective) term in either α_i or β_i can therefore be negative for all of central, forward, and backward differencing (if q_i did not change, then one of them must give positive coefficients).

For a specific example, consider the following theorem:

Theorem 5.1. *For general payoff structures, algorithm* (5.1) *with Dirichlet boundary conditions does not guarantee a positive coefficient discretization for one factor passport options.*

Proof. The proof will be by counterexample. Consider a butterfly type payoff with unequal slopes as drawn in Figure (5.1). To avoid notational clutter, we assume an equally spaced grid with spacing Δx. We further assume that the payoff slope is 1 to the left of the peak and -0.5 to the right of the peak. Let U_i be the option value at the peak, and U_{i-1} and U_{i+1} the values at its grid neighbours. We assume that Figure (5.1) shows the exact solution after an infinitesimal timestep. Hence, we will determine if it is possible to obtain a positive coefficient discretization given the solution in Figure (5.1).

With this setup, we find that $(U_{xx})_i = -3/(2\Delta x)$. Combined with $(U_x)_i$ values of $-0.5, 1$ and $1/4$ for forward, backward and central differencing respectively, applying (5.10) gives

$$\begin{aligned} q_{i,forward} &= x_i - \frac{r_1 \Delta x}{3\sigma^2} \\ q_{i,backward} &= x_i + \frac{2r_1 \Delta x}{3\sigma^2} \\ q_{i,central} &= x_i + \frac{r_1 \Delta x}{6\sigma^2}. \end{aligned}$$

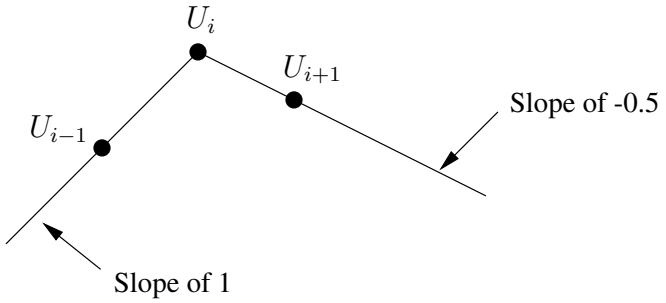


FIGURE 5.1: *Sample payoff that leads to non-positive coefficients when using central, forward and backward differencing.*

Using these values in equations (5.12)-(5.14) gives

$$\begin{aligned}
\beta_{i,forward} &= \frac{-5r_1^2}{18\sigma^2} + \frac{x_i}{\Delta x}(r_1 - r_2) \\
\alpha_{i,backward} &= \frac{-4r_1^2}{9\sigma^2} - \frac{x_i}{\Delta x}(r_1 - r_2) \\
\alpha_{i,central} &= \frac{-5r_1^2}{72\sigma^2} - \frac{x_i}{2\Delta x}(r_1 - r_2).
\end{aligned}$$

In each case, it is possible to obtain a negative coefficient. For example, if x_i is zero, or if r_1 equals r_2, then the terms above will be negative.

□

Problems occurred in the counterexample above since the q_i values for each discretization type were close to (but not equal to) x_i. This causes the first term (the diffusion part) in the coefficient equations to be small. In the case of convex payoffs, q is constrained by equation (5.8) to be either 1 or -1. Thus, either the q values are equal, or are sufficiently different to allow the first term to dominate. A precise treatment of this case is given by the next theorem:

Theorem 5.2. *For convex payoff structures, where q is limited to the values ± 1, there exists a grid $\{x_i\}$ such that algorithm* (5.1) *with Dirichlet boundary conditions guarantees*

a positive coefficient discretization for one factor passport options.

Proof. Consider the case where both forward differencing and backward differencing give negative coefficients. For backward differencing to give negative coefficients, equation (5.14) implies

$$\frac{r_1 q_{i,backward} - r_2 x_i}{\Delta x_i^-} > 0, \tag{5.19}$$

and

$$\frac{r_1 q_{i,backward} - r_2 x_i}{\Delta x_i^-} > \frac{\sigma^2}{2} \frac{2(x_i - q_{i,backward})^2}{\Delta x_i^- (\Delta x_i^+ + \Delta x_i^-)}. \tag{5.20}$$

Similarly, for forward differencing to give negative coefficients, equation (5.13) implies

$$\frac{r_1 q_{i,forward} - r_2 x_i}{\Delta x_i^+} < 0, \tag{5.21}$$

and

$$-\frac{r_1 q_{i,forward} - r_2 x_i}{\Delta x_i^+} > \frac{\sigma^2}{2} \frac{2(x_i - q_{i,forward})^2}{\Delta x_i^+ (\Delta x_i^+ + \Delta x_i^-)}. \tag{5.22}$$

Simplifying equations (5.20) and (5.22) gives the conditions for negative coefficients to be

$$\begin{aligned} x_{i+1} - x_{i-1} &> \frac{\sigma^2 (x_i - q_{i,backward})^2}{r_1 q_{i,backward} - r_2 x_i} \\ x_{i+1} - x_{i-1} &> \frac{\sigma^2 (x_i - q_{i,forward})^2}{-(r_1 q_{i,forward} - r_2 x_i)}. \end{aligned} \tag{5.23}$$

Conversely, we can say that a positive coefficient scheme can be achieved if

$$x_{i+1} - x_{i-1} < \max\left(\frac{\sigma^2 (x_i - q_{i,backward})^2}{r_1 q_{i,backward} - r_2 x_i}, \frac{\sigma^2 (x_i - q_{i,forward})^2}{-(r_1 q_{i,forward} - r_2 x_i)} \right) \tag{5.24}$$

for all i. We will now show that by possibly refining the grid, it is always possible to satisfy equation (5.24), and therefore to obtain a positive coefficient scheme. To start, note that if $q_{i,backward} = q_{i,forward}$ then one of equations (5.19) and (5.21) is violated, and

positive coefficients are ensured for node i. Hence, it suffices to examine the case when $q_{i,backward} = -(q_{i,forward}) = \pm 1$. Letting $q_i = \pm 1$, equation (5.24) can then be rewritten as

$$x_{i+1} - x_{i-1} < \max\left(\frac{\sigma^2(x_i - q_i)^2}{r_1 q_i - r_2 x_i}, \frac{\sigma^2(x_i + q_i)^2}{r_1 q_i + r_2 x_i)}\right). \tag{5.25}$$

Since the denominators in the max expression are bounded (for finite computational domains) and positive (see equations (5.19) – (5.22)), it is only possible for one of the terms in the max expression to be arbitrarily small. In the worst-case, we could solve

$$\min_x \left\{\frac{\sigma^2(x - q)^2}{r_1 q - r_2 x}, \frac{\sigma^2(x + q)^2}{r_1 q + r_2 x)}\right\} \tag{5.26}$$

for both $q = -1$ and $q = 1$ to obtain some x^* value which bounds the right hand side of equation (5.25) from below. This in turn bounds the right hand side of equation (5.24) from below. Since we are free to choose the grid points, we can make the left hand side of equation (5.24) as small as we want.

For example, given a grid $\{x_i\}$ of n points, one could (possibly) refine this grid to ensure positive coefficients as in algorithm (5.2) (note that x_1 is given in the original grid, and Dirichlet boundary conditions are assumed at $x = x_1$ and $x = x_{max}$). When new points are added at the midpoints, the grid spacing $x_{i+1} - x_{i-1}$ decreases. This grid spacing can never decrease below the value implied by equation (5.26). Hence, only a finite number of points can be added, and algorithm (5.2) must terminate.

Thus, by possibly refining the grid, a positive coefficient (and therefore provably stable for fully implicit timestepping) scheme can always be obtained for convex payoff functions (bang-bang problems). □

Remark 5.1. *Theorem 5.2 indicates that for convex payoffs, a positive coefficient scheme can always be constructed using central, forward and backward differencing. However, it does*

for $i = 2$ to $n-1$ **do**
  Test condition (5.24) for node x_i by setting $q_{backward} = 1, q_{forward} = -1$ and also $q_{backward} = -1, q_{forward} = 1$
  if the test fails **then**
    Insert points at $(x_i + x_{i-1})/2$ and $(x_i + x_{i+1})/2$
    $n \to n+2$
    $i \to i-1$ % test new point to left of x_i next
  end if
end for

ALGORITHM 5.2: *Procedure for refining a grid to obtain positive coefficients for one factor passport options with convex payoffs.*

not follow that a positive coefficient scheme is monotone. In fact, it appears that this method is not monotone, even in the convex case. We are therefore not assured of convergence to the viscosity solution (positive coefficients is a weaker requirement than monotonicity). We conjecture that it is not possible to develop a monotone method using central/forward/backward differences. We suspect that development of a monotone scheme for passport options will be exceedingly difficult.

5.4 Numerical Results

In this section we present our numerical test results. As with the uncertain volatility models in chapters 3 and 4, convergence to the viscosity solution is sought. The viscosity solution automatically results from a consistent, stable and monotone discretization (see Theorem 2.2). Unfortunately, proving that a discrete scheme has these characteristics appears to be quite difficult. For example, since a positive coefficient scheme cannot be guaranteed for non-convex payoffs, and since the discrete scheme can change from one iteration to the next, the approaches taken in chapters 2 and 3 do not apply. Nevertheless, by the numerical results of chapters 3 and 4, we expect implicit timestepping to be well-behaved. We also expect Rannacher timestepping to converge to the viscosity solution. In all tests below,

Rannacher timestepping implies Crank-Nicolson timestepping preceded by 4 fully implicit steps.

Our first test will be for a one factor problem with simplified parameter choices and a standard payoff: $u(x, t = T) = \max(x, 0)$. This establishes a baseline for all future tests. We then proceed to stress the methods by trying a capped call payoff. Due to the similarity reduction, the cap is relative to the underlying asset price. While this may not be entirely natural, our goal is to examine the impact on the numerical methods. A capped call payoff ensures that the sign of U_{xx} changes during the solution process, thereby enhancing the nonlinear aspects of the problem.

Subsequent tests focus on the full two dimensional problem of equation (5.4). The first of these attempts to reproduce the simple one factor result. Assuming the one factor computation was correct, we can comment on the effectiveness of the numerical techniques. We then test a capped call payoff for the two dimensional problem, where the cap is applied to the terminal wealth. This would be a more natural application of the cap, and is not equivalent to the one factor test. Unfortunately, we have no proof of convergence to the viscosity solution for any method, nor can we appeal to the one dimensional results.

When using equation (5.8) for q selection for one factor problems with a convex payoff, the grids used for testing always resulted in a positive coefficient scheme. However, when using equation (5.10) for q selection, Crank-Nicolson timestepping often failed to produce positive coefficients (see definition (2.1)), even if the payoff was convex. For non-convex payoffs, it was common for all timestepping methods to fail to produce positive coefficients, especially near $x = 0$. In these cases, we reverted to central differencing. Nevertheless, by examining the solution, stability was apparently maintained. For the two factor problems, we always used central differences in the finite volume discretization.

Type	Normal passport option
Time to expiry	1 year
r	0.0
Spot S	100
γ	0.0
r_t	0.0
r_c	0.0
σ	0.3

TABLE 5.1: *Model parameters for the first one factor passport test problem.*

5.4.1 One Factor Problems

The first round of tests focus on equation (5.7). Recall that this is the nonlinear PDE for the price of a passport option using a similarity reduction. In the first test, r and γ are set to zero. The remaining parameters are summarized in Table 5.1. Note that this example was examined by [3] and by [52]. With these choices, the selection of the optimal trading strategy in equation (5.8) becomes

$$q^* = \text{ sgn}\left(-\sigma^2 x \frac{\partial^2 u}{\partial x^2}\right). \tag{5.27}$$

Since the usual passport option payoff – $\max(x, 0)$ – is convex, the values for U_{xx} are always positive (see [3] for a proof). The choice of q then depends only on the sign of x, which makes the problem almost linear. Further, an analytic solution is available [3] which can be used to verify the numerical results. The analytic solution is assumed to be the viscosity solution, but a proof of this is unknown. In the numerical case, all methods are expected to be well-behaved, including Crank-Nicolson timestepping.

Numerical results for this problem using the frozen coefficient iterative method are shown in Table 5.2. The convergence tolerance was set to 10^{-6}. The initial timestep was 0.01. Since the problem is of the bang-bang type, results using the non-smooth Newton iteration method are identical to a frozen coefficient iteration. Further, a positive coefficient

	Fully Implicit			Crank-Nicolson			Rannacher (4 steps)		
Nodes	Value	Diff	Ratio	Value	Diff	Ratio	Value	Diff	Ratio
41	13.05083			13.06564			13.06505		
81	13.11237	0.06154		13.11977	0.05414		13.11963	0.05458	
161	13.12980	0.01743	3.53	13.13353	0.01376	3.94	13.13347	0.01384	3.94
321	13.13510	0.00530	3.29	13.13701	0.00348	3.95	13.13690	0.00347	3.99
641	13.13689	0.00179	2.96	13.13787	0.00086	4.07	13.13781	0.00087	4.00

TABLE 5.2: *Convergence results at $x = 0$ for a regular passport option, using equation* (5.8) *for q selection. Parameters are provided in Table 5.1. The initial timestep of 0.01 is halved at each grid refinement. "Diff" is the absolute value of the change in the solution as the grid is refined. "Ratio" is the ratio of successive differences. The analytic answer is 13.13810 [3].*

discretization was achieved in all cases. Nevertheless, the Newton (numerical Jacobian) method failed to converge after 2 grid refinements. Specifically, after two grid refinements, the Newton (numerical Jacobian) iteration failed to converge after 50 iterations during the first timestep.

As expected, all three timestepping methods converge to the same solution (the published analytic answer in [3]). Somewhat surprisingly, implicit timestepping appears to be converging at a higher than linear rate. However, the rate is seen to be decreasing, and is expected to be linear in the limit. Both Crank-Nicolson and Rannacher timestepping exhibit quadratic convergence.

Even though the payoff in this first example is convex, the use of equation (5.8) (or equivalently, equation (5.27)) to force q to always be ± 1 is based on an analytic treatment of the original PDE. When numerically solving the equations, the discrete approximation of derivative values, and the behaviour of the updates during the iterative process, may not enforce this condition. Hence, the test was repeated using the general equation (equation (5.10)) for the selection of q. The results are in Table 5.3. We see that the implicit and Rannacher results are the same as before. On the other hand, Crank-Nicolson timestepping appears to be converging to a different solution, or has a slowly growing instability. Similar behaviour was seen for uncertain volatility problems in chapter 3. Note also that Crank-Nicolson timestepping often failed to produce a positive coefficient scheme, unlike the other

	Fully Implicit			Crank-Nicolson			Rannacher (4 steps)		
Nodes	Value	Diff	Ratio	Value	Diff	Ratio	Value	Diff	Ratio
41	13.05083			13.06564			13.06505		
81	13.11237	0.06154		13.11977	0.05414		13.11963	0.05458	
161	13.12980	0.01743	3.53	13.14695	0.02718	1.99	13.13347	0.01384	3.94
321	13.13510	0.00530	3.29	13.13168	0.02121	1.28	13.13690	0.00347	3.99
641	13.13689	0.00179	2.96	13.13188	0.01976	1.07	13.13781	0.00087	4.00

TABLE 5.3: *Convergence results at $x = 0$ for a regular passport option, using equation (5.10) for q selection. Parameters are provided in Table 5.1. The initial timestep of 0.01 is halved at each grid refinement. "Diff" is the absolute value of the change in the solution as the grid is refined. "Ratio" is the ratio of successive differences. The analytic answer is 13.13810 [3].*

Type	normal passport option
Time to expiry	2 years
r	0.05
Spot S	100
γ	0.045
r_t	0
r_c	0
σ	0.3

TABLE 5.4: *Model parameters for the second passport test problem.*

timestepping methods.

In our next example, we set r to 5%, γ to 4.5% and consider a 2 year time horizon. All model parameters are summarized in Table 5.4. This problem was also solved by [3] and [52]. Our results (using equation (5.8) which forces $q = \pm 1$) are in Table 5.5. As before, the non-smooth Newton and frozen coefficient methods gave identical results. The regular Newton (numerical Jacobian) iteration results are not given since convergence only occurred for the initial grid. Also as before, if the general constraint for q is used (equation (5.10), Crank-Nicolson timestepping either converges to a different solution, or has a slowly growing instability.

Our final example for the one factor case uses the parameters in Table 5.4, except the payoff has been changed to a capped call. Specifically, the payoff is $\min(\max(x, 0), 0.2)$. Since the sign of U_{xx} changes during the solution process, we expect this to be a more

	Fully Implicit			Crank-Nicolson			Rannacher (4 steps)		
Nodes	Value	Diff	Ratio	Value	Diff	Ratio	Value	Diff	Ratio
41	17.32369			17.35750			17.35297		
81	17.40379	0.08009		17.42128	0.06379		17.41935	0.06638	
161	17.42837	0.02458	3.26	17.43756	0.01628	3.92	17.43648	0.01713	3.87
321	17.43669	0.00833	2.95	17.44146	0.00390	4.18	17.44087	0.00439	3.90
641	17.43988	0.00319	2.61	17.44231	0.00086	4.55	17.44201	0.00114	3.85

TABLE 5.5: *Convergence results at $x = 0$ for a regular passport option using equation* (5.8) *for q selection. Parameters are provided in Table 5.4. The initial timestep of 0.01 is halved at each grid refinement. "Diff" is the absolute value of the change in the solution as the grid is refined. "Ratio" is the ratio of successive differences.*

challenging test. Further, equation (5.10) must be used to determine the q values. As a result, the different iterative methods will, in general, behave differently. Also, it was impossible to achieve positive coefficients in many cases, especially near $x = 0$. Results for the frozen coefficient and non-smooth Newton methods are given in Table 5.6. Again, the Newton (numerical differentiation) method failed to converge for most tests.

For all methods, implicit and Rannacher timestepping converges to the same solution. All Crank-Nicolson results appear to be converging to a different answer, or perhaps diverging. Strangely, the non-smooth Newton method failed to converge for the last grid refinement. Looking at details of the iteration, the scheme ended up oscillating between several states.

To further differentiate the frozen coefficient and non-smooth Newton iterative methods, we report the number of iterations required to arrive at the solution. Total iteration counts for each run are shown in Table 5.7. We see that for coarse grids, both methods require about the same number of iterations. Further, both methods require an average of just over two iterations per timestep, indicating a fairly mild nonlinearity (a linear problem would require exactly two iterations, the second being a zero update). However, as the grid is refined, the average number of iterations per timestep increases for the non-smooth Newton method. The similarity of the solutions indicates that both methods ultimately converge to the same solution. Alternatively, one could argue that the frozen coefficient method is

Frozen Coefficient Method

	Fully Implicit			Crank-Nicolson			Rannacher (4 steps)		
Nodes	Value	Diff	Ratio	Value	Diff	Ratio	Value	Diff	Ratio
41	12.60966			12.63236			12.63114		
81	12.64414	0.03447		12.66723	0.03487		12.65503	0.02390	
161	12.65561	0.01148	3.00	12.68874	0.02151	1.62	12.66112	0.00608	3.93
321	12.65989	0.00428	2.68	12.70668	0.01794	1.20	12.66267	0.00155	3.92
641	12.66167	0.00178	2.40	12.73477	0.02809	0.64	12.66307	0.00040	3.86

Non-smooth Newton Method

	Fully Implicit			Crank-Nicolson			Rannacher (4 steps)		
Nodes	Value	Diff	Ratio	Value	Diff	Ratio	Value	Diff	Ratio
41	12.60966			12.63237			12.63114		
81	12.64414	0.03447		12.66723	0.03487		12.65504	0.02390	
161	12.65561	0.01148	3.00	12.68874	0.02151	1.62	12.66112	0.00608	3.93
321	12.65989	0.00428	2.68	12.70668	0.01794	1.20	12.66267	0.00155	3.92
641	12.66167	0.00178	2.40	–	–	–	12.66307	0.00040	3.86

TABLE 5.6: *Convergence results at $x = 0$ for passport option with a capped call payoff. Parameters are provided in Table 5.4. The initial timestep of 0.01 is halved at each grid refinement. "Diff" is the absolute value of the change in the solution as the grid is refined. "Ratio" is the ratio of successive differences.*

	Frozen Coefficients			Non-smooth Newton		
Timesteps	Implicit	Crank-Nicolson	Rannacher	Implicit	Crank-Nicolson	Rannacher
100	207	207	207	206	205	206
200	412	415	412	415	416	416
400	828	836	827	963	902	917
800	1655	1654	1654	2605	2509	2500
1600	3302	3315	3293	8050	—	7328

TABLE 5.7: *Total iteration counts for the capped call example. Parameters are provided in Table 5.4.*

converging more rapidly than expected. Based on these results, in all subsequent numerical examples, the frozen coefficient iteration will be used.

5.4.2 Two Factor Problems

Our first two factor test is the same as the simple one factor case considered before (Table 5.1 data). This should, of course, converge to the known analytic solution. Results are in Table 5.8. The initial (triangular) grid had 41 non-uniform nodes in the w direction, and

	Fully Implicit			Crank-Nicolson			Rannacher (4 steps)		
Nodes	Value	Diff	Ratio	Value	Diff	Ratio	Value	Diff	Ratio
41×41	13.06466			13.1305			13.07886		
81×81	13.11571	0.05104		13.2340	0.1035		13.12295	0.04409	
161×161	13.13061	0.01491	3.42	13.3159	0.0819	1.26	13.13427	0.01132	3.90
321×321	13.13530	0.00468	3.18	13.3441	0.0282	2.91	13.13714	0.00286	3.95

TABLE 5.8: *Convergence results at $w = 0, S = 100$ for a passport option with a normal payoff, using the full two factor equations. Parameters are provided in Table 5.1. The initial timestep of 0.01 is halved at each grid refinement. "Diff" is the absolute value of the change in the solution as the grid is refined. "Ratio" is the ratio of successive differences.*

	Fully Implicit			Crank-Nicolson			Rannacher (4 steps)		
Nodes	Value	Diff	Ratio	Value	Diff	Ratio	Value	Diff	Ratio
41×41	17.35096			17.4782			17.36631		
81×81	17.41434	0.06338		17.6048	0.1266		17.42243	0.05612	
161×161	17.43262	0.01828	3.47	17.7158	0.1110	1.14	17.43680	0.01437	3.91
321×321	17.43834	0.00573	3.19	17.8665	0.1507	0.74	17.44048	0.00368	3.90

TABLE 5.9: *Convergence results at $w = 0, S = 100$ for a passport option with a normal payoff, using the full two factor equations. Parameters are provided in Table 5.4. The initial timestep of 0.01 is halved at each grid refinement. "Diff" is the absolute value of the change in the solution as the grid is refined. "Ratio" is the ratio of successive differences.*

41 non-uniform nodes in the asset direction. Subsequent grids were created by uniform refinement. As in the one factor case, implicit timestepping is converging to the analytic answer. Although the apparent convergence rate is high, we again expect convergence in the limit to be linear. Perhaps surprisingly, pure Crank-Nicolson timestepping is either converging to a non-viscosity solution, or is unstable. Rannacher timestepping restores correct convergence at a quadratic rate.

For our next test we repeat the test summarized by the parameters in Table 5.4. The results are in Table 5.9. Again, implicit timestepping and Rannacher timestepping converge to the correct answer. Crank-Nicolson timestepping appears to be unstable, with the solution growing by approximately the same amount at each refinement. Clearly, Crank-Nicolson timestepping has problems in the case of the full two factor equations.

In anticipation of the application of passport options to trader compensation, we also test a two factor capped call payoff: $\min(\max(w, 0), 20)$. Other parameters are shown

	Fully Implicit			Crank-Nicolson			Rannacher (4 steps)		
Nodes	Value	Diff	Ratio	Value	Diff	Ratio	Value	Diff	Ratio
41×41	13.04017			12.8659			13.05641		
81×81	13.08173	0.04156		12.9644	0.0985		13.09009	0.03368	
161×161	13.09646	0.01473	2.82	13.0973	0.1328	0.742	13.10070	0.01061	3.17
321×321	13.10246	0.00600	2.46	13.3101	0.2128	0.624	13.10459	0.00389	2.72

TABLE 5.10: *Convergence results at $x = 0, S = 100$ for passport option with a capped call payoff, using the full two factor equations. Parameters are provided in Table 5.4. The initial timestep of 0.01 is halved at each grid refinement. "Diff" is the absolute value of the change in the solution as the grid is refined. "Ratio" is the ratio of successive differences.*

in Table 5.4. For this problem we have no analytic answer with which to compare, nor can we make a similarity reduction. Our results are in Table 5.10. Both implicit and Rannacher timestepping appear to be converging to the same answer at a linear rate. Crank-Nicolson timestepping appears to be unstable, with the solution differences growing at each refinement. It is interesting to note that implicit timestepping appears to be almost as accurate as Rannacher timestepping. Given that both appear to converge at the same rate, we would recommend implicit timestepping for nonlinear problems with non-convex payoff functions. Again, similar results were reported in chapter 3 for the uncertain volatility model.

Given that we can only achieve linear convergence for two factor problems with non-convex payoff functions, it would appear to be appropriate to evaluate the nonlinear coefficients explicitly. In other words, all coefficients are evaluated using solution data at the previous timestep. This avoids having to solve a nonlinear set of equations at each timestep, and is in fact equivalent to forcing only one iteration in algorithm 2.1. Unfortunately, it also means that the most we can hope for is linear convergence of the solution. See chapter 2 for more information about this method. Results using this approach are shown in Table 5.11. Perhaps surprisingly, these results are almost identical to the results in Table 5.10. This implies that the problem is almost linear, and little advantage is gained by doing a full nonlinear iteration at each step. This contrasts the observations in chapter 3, for which an

	Fully Implicit			Crank-Nicolson			Rannacher (4 steps)		
Nodes	Value	Diff	Ratio	Value	Diff	Ratio	Value	Diff	Ratio
41×41	13.03975			12.6981			13.05603		
81×81	13.08158	0.04183		12.7222	0.0240		13.08995	0.03392	
161×161	13.09641	0.01483	2.82	12.7455	0.0233	1.03	13.10065	0.01070	3.17
321×321	13.10244	0.00603	2.46	12.7853	0.0398	0.586	13.10458	0.00393	2.72

TABLE 5.11: *Convergence results for passport option with capped call payoff, using the full two factor equations, and forcing only one nonlinear iteration per timestep (explicit evaluation of nonlinear coefficients). Parameters are provided in Table 5.4. The timestep is halved at each grid refinement. "Diff" is the absolute value of the change in the solution as the grid is refined. "Ratio" is the ratio of successive differences.*

explicit evaluation of nonlinear coefficients in an uncertain volatility model lead to substantially slower convergence rates. The difference occurs because the trading strategy q in the passport problem is relatively constant. The trading strategy can be different in different regions, but the boundaries of these regions change very slowly. The only exception for the problems being tested is at the start of the solution process. Thus, it may be advantageous to solve the full nonlinear equations for the first few timesteps, and to then switch to an explicit evaluation of the coefficients (i.e. similar to Rannacher timestepping).

5.5 Application to Trader Compensation

5.5.1 Motivation

The classical passport option can be seen as insurance against trading losses. However, given their relatively high cost, it is unclear how often passport deals are used in practice. Recently, passport option theory has been applied to the problem of trader compensation [1]. In many cases, a trader's bonus is some percentage of his/her trading account (if positive). A rational trader will therefore seek to maximize his/her expected bonus by maximizing the expected value of the trading account. Mathematically, this means the trader needs to solve

$$V(S,w,t) = \max_{|q|\leq 1} E_p^{S,w,t}[e^{-r(T-t)}U(\max(w(T),0))], \tag{5.28}$$

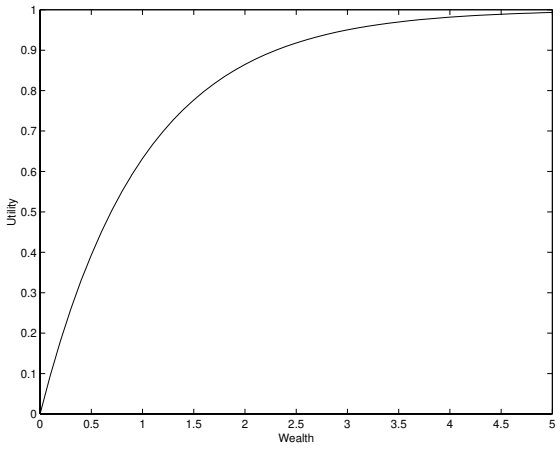


FIGURE 5.2: *Depiction of an exponential utility function:* $U(w) = 1 - e^{-w}$.

where U is the trader's utility function, and the expectation E_p is with respect to the real drift μ, as specified by equation (5.1). A utility function represents the level of "happiness" associated with different levels of wealth [59]. The actual function used will depend on an individual's risk preferences. Nevertheless, in general, utility functions tend to have the following two properties:

- $U'(w) \geq 0$: With a non-negative first derivative, more wealth is preferred to less.
- $U''(w) < 0$: Most investors are **risk averse**, meaning that their utility functions are concave. Practically, as wealth increases, the amount of extra happiness decreases (an increase from 1 to 2 gives more happiness than an increase from 2 to 3).

An example of an exponential utility function is shown in Figure 5.2. For a discussion of this, and other utility functions, see [59].

Applying Bellman's principle of dynamic programming to equation (5.28), one obtains

$$V(S, w, t) = \max_{|q| \leq 1} E_p[e^{-r\Delta t} V(S_{t+\Delta t}, w_{t+\Delta t}, t + \Delta t)]. \tag{5.29}$$

By the standard procedure of applying Ito's lemma to equation (5.29), it can be shown that V must satisfy the Hamilton-Jacobi-Bellman equation

$$-V_t = -rV + \mu S V_S + \max_{|q|\leq 1} \left\{ \left[(\mu - r) qS + rw \right] V_w + \frac{\sigma^2 S^2}{2} \left[V_{SS} + 2qV_{wS} + q^2 V_{ww} \right] \right\}, \quad (5.30)$$

where μ is the real world drift rate of the underlying asset. To be consistent with existing literature [1, 59], r_t and r_c have been set to the risk-free rate r when deriving equation (5.30). With these simplifications, and setting $r - \gamma$ to μ, equation (5.30) is identical to equation (5.2). However, the final condition (initial condition after setting $t = T - \tau$) is now

$$V(S, w, T) = U(\max(w(T), 0)). \quad (5.31)$$

Equation (5.31) simply means that the passport option payoff is modified by the utility of the trader.

In [1], a linear utility function was adopted, which makes the terminal conditions of the trader equation (5.31) equal to the passport option payoff. Reference [1] also extended the basic passport/trader bonus model to account for effects such as trader skill, a Sharpe ratio, dismissal conditions, and path-dependent bonus structures. The focus of those results, determined in part from the PDE solution of the pricing equations, was mostly on the expected value and distribution of the trader's bonus. To a lesser extent, distributions of the entire trading account (bank profit/loss) were shown. Note that distributions were obtained by simulating many asset paths, but always adopting the optimal trading strategy.

To contrast these results, we focus on the full distribution of the final trading account. This is similar to considering the point of view of a financial institution which employs a trader. If the trader has a positive balance in her trading account at the end of the year, the institution pays a bonus. On the other hand, if the trading account is negative, the financial institution takes the loss, and the trader effectively exercises the passport option.

The optimal strategy for the trader is, therefore, not optimal for the financial institution. In particular, since the trader has effectively a free option, this encourages risky trading activities.

To potentially avoid such behaviour, the utility of the trader should be aligned with the utility of the bank. As shown by equation (5.31), this is determined by the payoff and the utility function of the trader. The generated optimal trading strategy then changes, which changes the distribution of account balances. For example, by making the payoff turn negative at negative account balances, there is incentive to avoid this "penalty region". Technically, it means that an option holder would have some risk, and could owe money at option expiry. Alternatively, we could imagine that some portion of the trader's previous year's bonus is held in escrow until the end of the current year. If the current year trading account is negative, the trader may lose the withheld portion of the previous year's bonus. In fact, it could be argued that a trader should earn the value of the option she holds, before any bonus is paid. For the application to generating strategies, this means that the trading strategy will try to avoid the penalty region by adjusting the amount invested. Similar adjustments occur if the payoff is capped (or altered) for positive account balances.

To generate the account balance distributions, equation (5.30) is solved for the optimal trading strategy $q(S, w, t)$. Values of this strategy for each grid node (S, w) and timestep t are written to a data file. Any number of asset and wealth paths can then be simulated according to

$$dS = \mu S dt + \sigma S dZ, \tag{5.32}$$

and

$$dw = rwdt + q(dS - rSdt). \tag{5.33}$$

At each simulation timestep, the PDE data file is read to determine the appropriate position q (using linear interpolation when required). Time intervals of $1/250$ (approximately 1 day)

are used for the simulations. 100,000 simulations are done for each test case. Unless otherwise stated, other parameters are as follows: $r = 5\%, \mu = 10\%, T = 1, \sigma = 0.3, S_0 = 100$.

Testing begins with a simple linear utility function, and progresses to an exponential utility function. We then take a direct approach, and change the payoff to force specific trends in the trading strategy. This is done by adding caps and/or penalty regions. Theoretically, these payoffs could correspond to some utility function, but we do not attempt to back out what these utility functions would be. Finally, value-at-risk (VAR) values are computed for each of the preceding distributions. This allows for a more quantitative comparison of the resulting account distributions.

5.5.2 Linear Utility

We begin with a linear utility function, which makes equation (5.31) the regular passport option payoff. Our first test sets all interest rates to zero, meaning the asset price movement is controlled solely by diffusion. The final account balances after 100,000 simulations are shown in Figure 5.3.

As expected, the mean account balance is essentially zero (it approaches zero as the number of simulations approaches infinity). Nevertheless, it is interesting to note how the distribution has two symmetrical peaks: one for positive account balances, and one for negative account balances. The peak for positive account balances results from the trading strategy attempting to maximize the (positive) expected return. This shift to the right is balanced by an equal peak shifted to the left. The symmetry occurs from the symmetrical trading strategy, which for this example happens to be go long when behind ($q = 1$ when $w < 0$) and short when ahead ($q = -1$ when $w > 0$).

For what is perhaps a more realistic scenario, we set r to 5% and μ to 10%. A histogram of account balances is shown in Figure 5.4. The mean account balance is positive (about

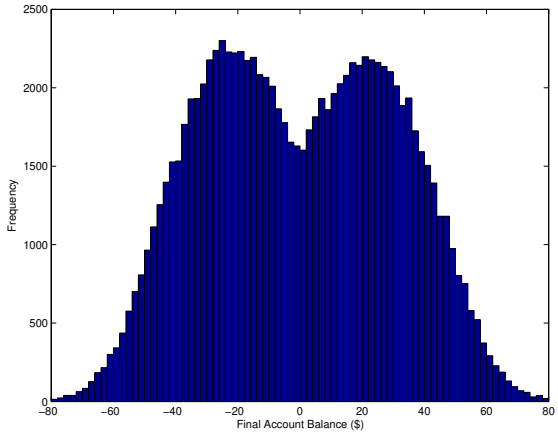


FIGURE 5.3: *Distribution of trading account when following the standard passport option trading strategy. Only diffusion affects the asset paths. The initial asset price was \$100. The peaks occur around* ± 20*, which correspond to returns of* $\pm 20\,\%$*.*

5.5) due to the positive expected drift rate. The optimal trading strategy is to always go long ($q = 1$).

5.5.3 Exponential Utility

Instead of using linear utility, we can use an exponential utility function given by

$$U(x) = -ae^{-bx} \tag{5.34}$$

(maximizing (5.34) is the same as maximizing $1 - ae^{-bx}$ which is also commonly seen in the financial literature for an exponential utility function).

The initial conditions for equation (5.30) become $V(S, w, T) = -ae^{-b\max(w,0)}$. Mathematically, this is equivalent to a linear utility with a modified payoff. Economically, we can say that as b increases, the investor becomes increasingly risk-averse. This viewpoint is evident in the results shown in Figure 5.5. For simplicity, we have set a to 1 in all tests. As before, we have shown the final account balances for 100,000 simulations. When b is large,

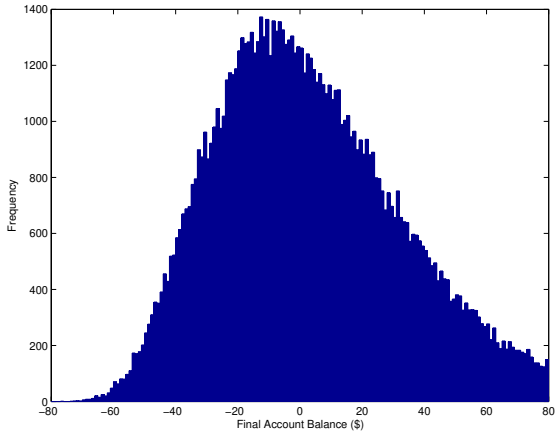


FIGURE 5.4: *Distribution of trading account when following the standard passport option trading strategy. The risk-free rate is 5% while the expected return is 10%. The initial asset price was $100.*

the payoff is reminiscent of a digital option (see Figure 5.5a). For a pure digital payoff in the passport option case, once the investor has made positive wealth, the optimal trading strategy is to stop trading: there is nothing to be gained and something to be lost. With a large b value, trading rapidly slows once a profit has been made. Hence, the final account balances are clustered just to the right of zero. As b decreases, the investor gains more incentive to keep trading at positive wealth balances. This shifts the positive peak to the right. For small b values, the optimal strategy is to buy and hold, making the final account distribution similar to the pure diffusion case shown in Figure 5.4.

5.5.4 Direct Modification of Payoff Functions

In this section, it is assumed that $U(x) = x$ (i.e. linear utility). The payoff function $V(w, S, T)$ will be directly modified to achieve different distributions.

In the zero drift linear utility example, the distribution of account balances shows that, on average, one can expect either a large gain or a large loss. In the positive drift linear utility example, the distribution is more bell shaped, but both large gains and losses can

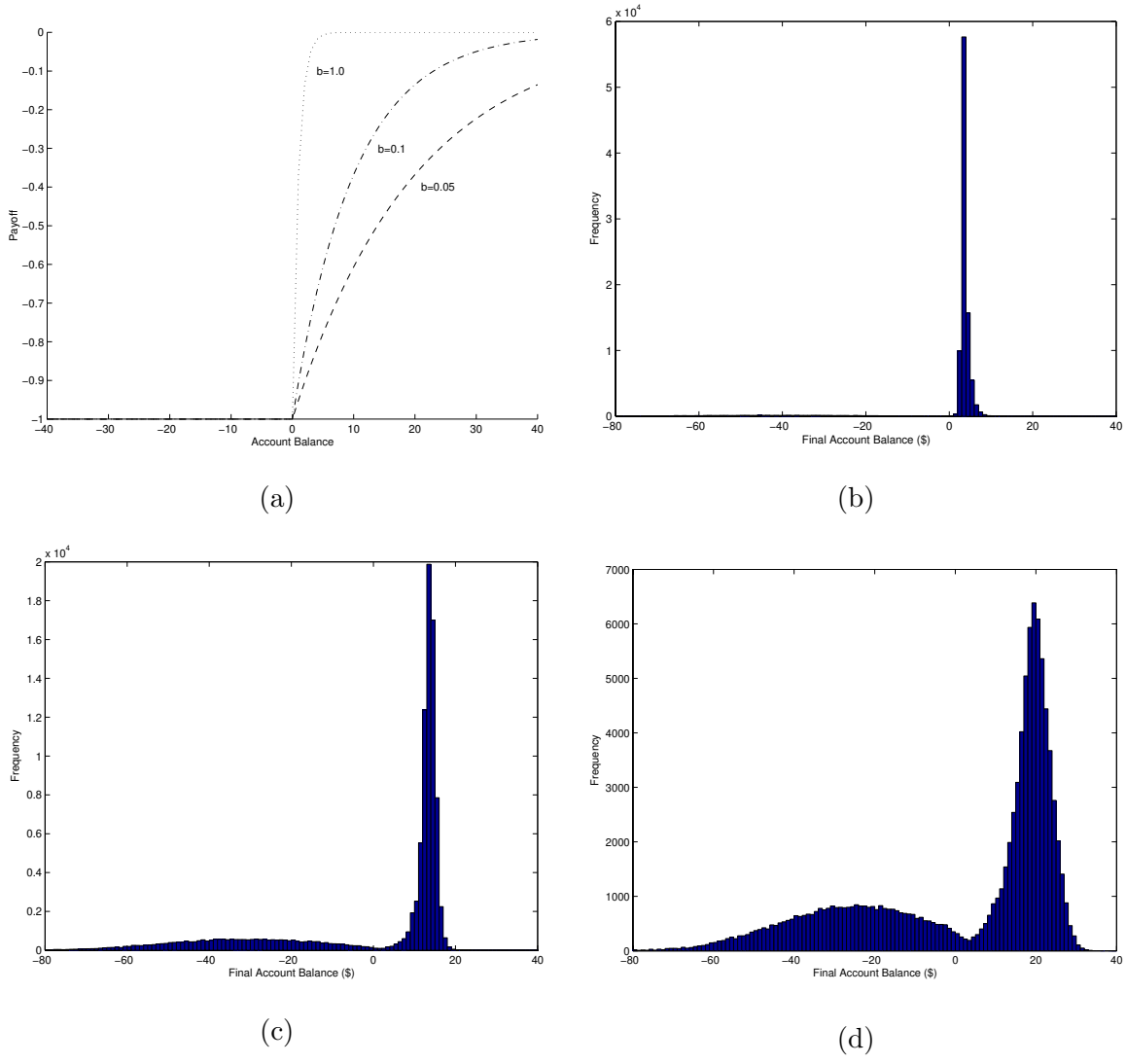


FIGURE 5.5: *Final account balance distributions when the usual passport payoff is transformed by an exponential utility function:* $V(S, w, T) = -e^{-b \max(w,0)}$ *Figure (a) shows the payoffs. Figures (b), (c) and (d) show account balance distributions when b is 1.0, 0.1 and 0.05, respectively.*

occur. Large losses can also occur with the exponential utility examples. While a large gain is usually desirable, one may want to control the expected losses. This can be done by adjusting the passport payoff. For our test, we leave the payoff at zero for trading account balances from 0 to -40, but then have the payoff linearly decrease as the account goes further negative. Precisely, the payoff $V(w, S, T)$ is set by the algorithm

if $w > 0$ **then**

$V(w, S, T) = w$

else if $w > -40$ AND $w \leq 0$ **then**

$V(w, S, T) = 0.0$

else

$V(w, S, T) = m(w + 40)$

end if

for m values of 0.1, 0.5 and 1 (m is the slope of the payoff in the penalty area). The payoffs are shown in Figure 5.6(a). Thus, a type of penalty is applied if the trading account goes below -$40. This kind of payoff is similar to imposing a dismissal boundary condition [1], in which the option is knocked-out if the trading account ever falls below a certain value. The knock-out corresponds to dismissing the trader, also known as the "in by nine, out by five" effect. The results of testing are shown in Figure 5.6(b),(c) and (d).

As the penalty for poor performance increases (the payoff slope increases for negative account balances) the distributions are seen to be cut off at the payoff turning point. This results from the trading position getting smaller as the turning point is approached. In some sense, the penalty for crossing the turning point is not worth the potential gain from a large position. On the other hand, the right hand (positive) side of the distributions remain unchanged. This is because the trading strategy for account balances far from the penalty region remains unchanged. Note that the mean of all distributions is still zero.

Similar effects can be produced to the right hand sides of the distributions by capping

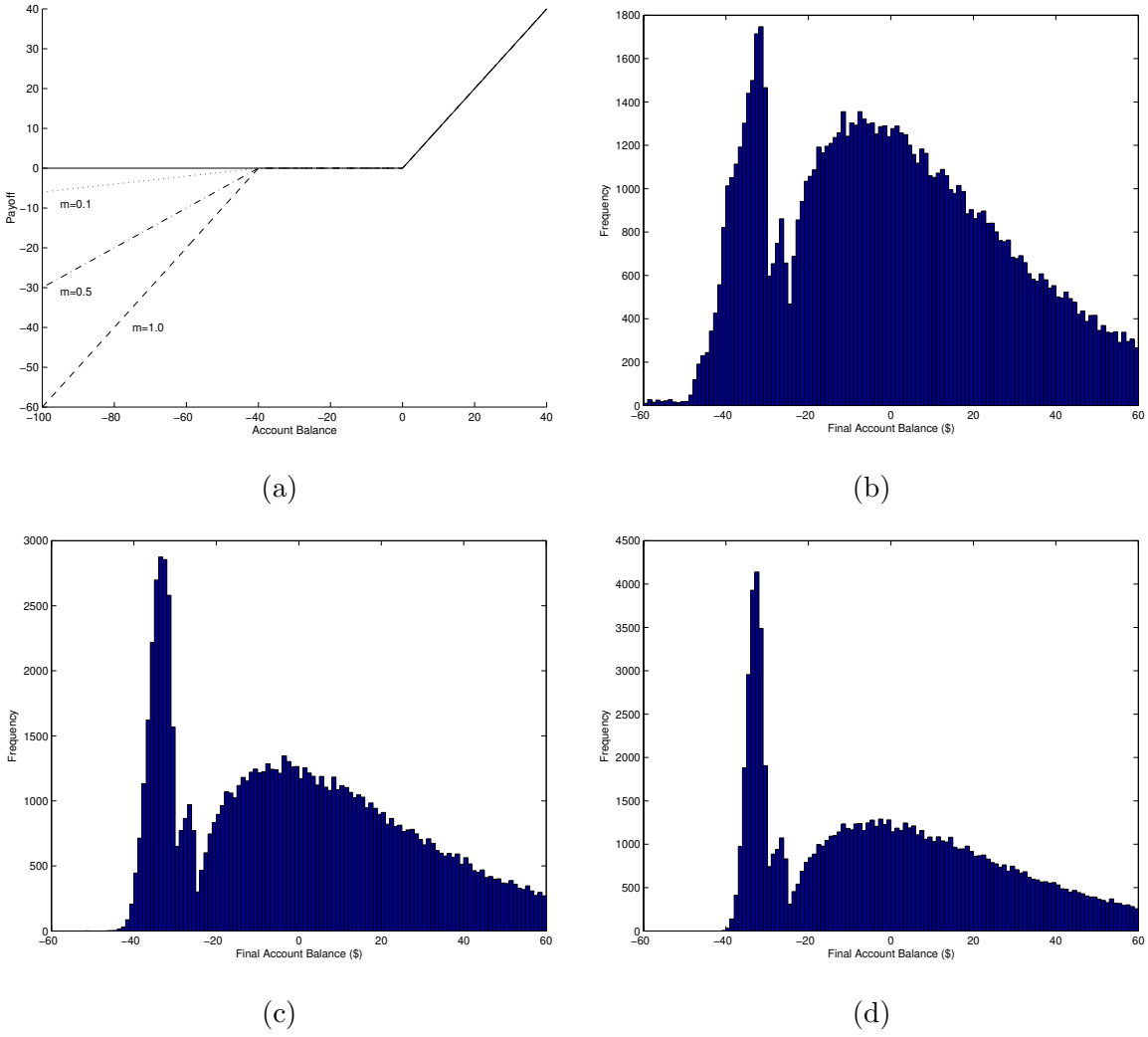


FIGURE 5.6: *Final account balance distributions when the payoff becomes negative for negative account balances (linear utility for an adjusted payoff). Figure (a) shows the payoffs. Figures (b), (c) and (d) show account balance distributions when the slope of the payoff in negative account regions is 0.1, 0.5 and 1, respectively.*

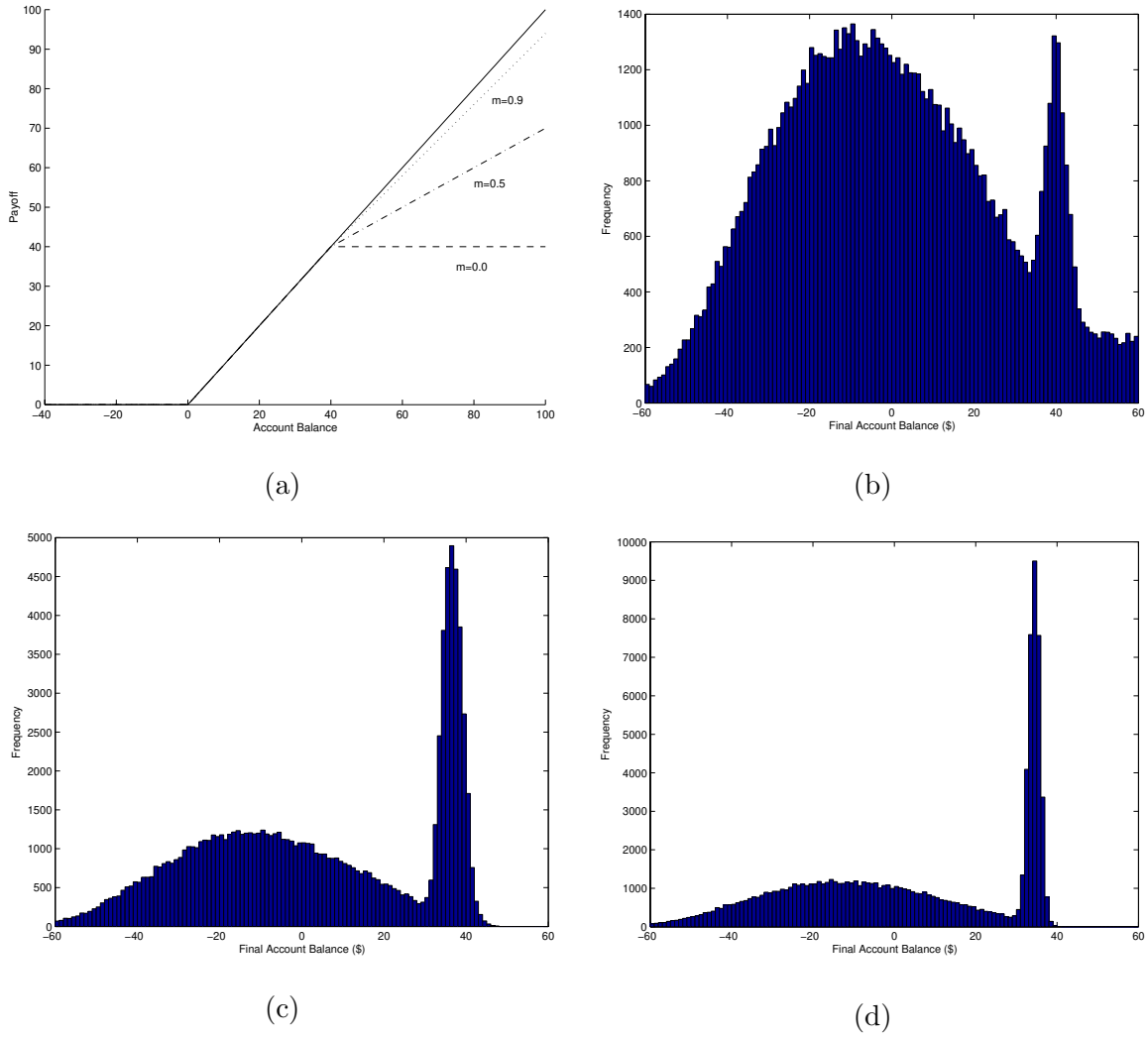


FIGURE 5.7: *Final account balance distributions when the payoff is capped or reduced for positive account balances (linear utility for an adjusted payoff). Figure (a) shows the payoffs. Figures (b), (c) and (d) show account balance distributions when the slope of the payoff for positive account regions is 0.9, 0.5 and 0, respectively.*

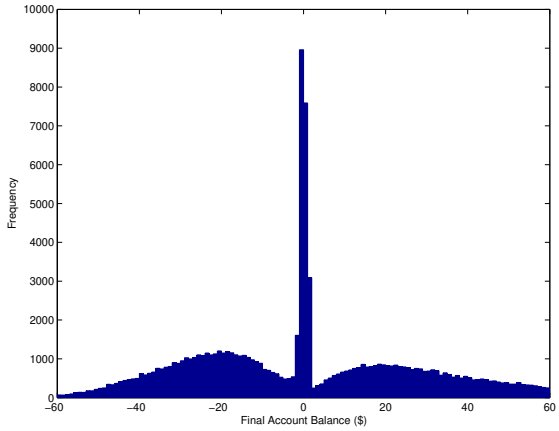


FIGURE 5.8: *Distribution of trading account when the payoff is set to model a typical trader compensation structure (linear utility for a modified payoff).*

the payoff. In these cases, the slope of the payoff decreases past a certain point, or even caps at a maximal value. The actual payoffs tested are shown in Figure 5.7(a). The resulting distributions are shown in Figure 5.7(b), (c) and (d). As expected, the distributions spike near the point where the cap (slope change) is introduced.

For a final test, a payoff that attempts to mimic a typical trader compensation structure is examined. For wealth values less than zero, the payoff is zero. This would be a 'firing' region. For wealth values from 0 to 10% of the initial asset price (in this case, $100*0.1 = 10$), the payoff is fixed at 0.5. This is meant to reflect a base salary level. At higher wealth levels, the payoff increases linearly with a slope of 0.1 (base level plus 10% of wealth generated during trading). The final account balances are shown in Figure 5.8. At positive wealth levels near zero, and close to expiry time, the optimal trading strategy is to do nothing – q approaches zero. This results from a fear of hitting the firing region. Hence, a peak occurs near wealth levels of zero in Figure 5.8. However, if wealth levels are away from zero, the optimal behaviour is similar to a regular passport option. Thus, the distribution is similar to Figure 5.4 in these regions.

5.5.5 VAR Calculations

To further compare the different methods for computing optimal trading strategies, value-at-risk (VAR) values are computed for each of the account distributions presented above. According to [60], VAR is commonly defined as "an estimate, with a given degree of confidence, of how much one can lose from one's portfolio over a given time horizon". For example, a 1-day 95% VAR of $\$10,000$ means that the probability of losing more than $\$10,000$ in one day is $1 - 0.95 = 0.05$ (or 5%). In this way, VAR values give an indication of down-side risk, or a measure of the tail end of a distribution. Typical confidence levels are between 95% and 99%. The time horizon is typically from 1 to 5 days, and is meant to reflect the amount of time needed to liquidate the portfolio.

In the computation of VAR values below, a slightly different version of VAR is used. Specifically, the expected loss of the passport option (trader compensation) over the entire life of the option is computed. The goal is to obtain a somewhat statistical measure of the tail end of the distributions shown in the previous section. Only 95% and 99% confidence intervals will be considered. The results are given in Table 5.12.

As a base level, the linear utility function results in a VAR(95%) value of -40.5 (recall that the initial asset price is \$100, and that the simulated time is one year). The tests with capped payoff have little effect on VAR values. This is to be expected, as the cap is applied in the positive payoff region, and is unlikely to affect the worst-case (negative) outcomes. On the other hand, adding penalty regions improves the VAR values by shifting them closer to zero. Some caution must be taken in interpreting the numbers though. If the penalty region is harsh (large slope), and begins sufficiently close to final account balances of zero, then VAR values will be close to zero (the distribution will be "jammed" up against a left boundary). However, VAR values at different confidence levels (e.g. 80% and 99%) will be almost identical, thereby losing practical meaning.

The most interesting results occur for the exponential utility function. As the power of

Test	VAR(95%)	VAR(99%)
linear utility	-40.5	-52.3
exponential utility: $b = 0.01$	-40.7	-53.2
exponential utility: $b = 0.05$	-46.2	-60.6
exponential utility: $b = 0.1$	-49.9	-65.6
exponential utility: $b = 0.5$	-44.9	-68.4
exponential utility: $b = 1.0$	-35.8	-67.0
capped: $m = 0.0$	-41.8	-54.5
capped: $m = 0.5$	-41.6	-53.8
capped: $m = 0.9$	-40.6	-52.3
penalty: $m = 0.1$	-38.3	-44.7
penalty: $m = 0.5$	-35.7	-38.6
penalty: $m = 1.0$	-34.4	-36.5
trader model	-40.8	-52.9

TABLE 5.12: *Value-at-risk values for the resulting distribution of account balances.*

the exponent initially increases, the VAR values worsen (expected loss increases). However, past a certain point (extra values have been added to the table to help identify this point), the trend reverses. Recall that the trader's utility function is applied to the regular passport option payoff. Hence, for wealth values less than zero, the resulting payoff is the same. The utility function therefore has little impact on the negative wealth regions (any impact that does occur diffuses from the positive payoff region). Consequently, VAR values are hard to predict for the exponential utility function. This raises some interesting questions about common trader compensation packages.

Chapter 6

Conclusions

As contingent claim contracts become more complicated, the appearance of nonlinear pricing models is sure to increase. In this thesis, techniques have been developed and guidelines established for dealing with the resulting nonlinear pricing equations.

If an implicit method is used to discretize a nonlinear PDE, we are faced with having to solve a set of nonlinear algebraic equations at each timestep. Provided that the discretization satisfies certain conditions, it has been shown that the fixed point iteration scheme is guaranteed to converge for problems with optimized nonlinear coefficients. Further, convergence is monotonic. Experiments indicate that for typical timesteps, convergence occurs in just over two iterations per timestep on average for the problems tested.

In the case of one factor uncertain volatility problems, it was proven that a fully implicit discretization is monotone, and hence converges to the viscosity solution of the PDE. On the other hand, Crank-Nicolson timestepping is only conditionally monotone. Numerical examples show that Crank-Nicolson timestepping can generate incorrect (i.e. not viscosity) solutions to the PDE, or even unstable results, if a timestep is used which results in a non-monotone discretization.

Numerical experiments further show that Crank-Nicolson timestepping can converge to

the viscosity solution if a small number (2-4) of fully implicit steps are taken at the beginning, followed by Crank-Nicolson thereafter. For continuous, but non-smooth payoffs, testing indicated convergence at a quadratic rate, which is an improvement over the linear convergence of a fully implicit method. Unfortunately, quadratic convergence (using Rannacher timestepping) could not be achieved for discontinuous payoffs (e.g./ digital options). In this case, fully implicit timestepping should be used, since convergence to the viscosity solution is guaranteed.

Another approach for solving nonlinear equations is to evaluate the nonlinear term explicitly. This avoids the need to solve nonlinear equations at each step, but is expected to only have a linear convergence rate. In fact, this approach appears to converge at a slower rate than implicit methods for one factor uncertain volatility problems. An explicit evaluation of nonlinear terms is also not guaranteed to converge to the viscosity solution, although no difficulties were noticed in the numerical tests.

When solving nonlinear equations numerically, it is important to optimize the discrete form of the equations. This was especially important for the two factor uncertain volatility problems. In these cases, it was shown how the discrete form of the equations to be optimized is equivalent to a specific numerical approximation of the second derivative values (up to a constant). Using a different approximation for the derivative values can cause the numerical scheme to fail, in the sense that an iterative solver may loop indefinitely between two or more states.

Numerical examples for two factor uncertain volatility problems were provided for a simple call on the max of two assets, as well as a butterfly type payoff. In both cases, implicit and Rannacher timestepping converged to the same (presumably correct) solution, while Crank-Nicolson was either converging to a non-viscosity solution, or had a slowly growing instability. For the call payoff, Rannacher timestepping showed quadratic convergence. Unfortunately, convergence was only linear for the butterfly payoff. Note how this is a

slightly worse situation than for the one factor uncertain volatility problems, in which convergence was linear for only discontinuous payoffs. In general, only linear convergence can be expected for Rannacher timestepping when pricing nonlinear equations.

When applied directly to an option payoff, the uncertain volatility model with reasonable volatility and correlation ranges can lead to large price spreads. It was shown how this spread can be greatly reduced with optimal static hedging. Once a static hedge is established, only the residual payoff needs to be priced. While the greatest spread reduction for max of two asset call options occured when the hedge was composed of other max of two asset call options, significant reduction also occured when hedging with one factor call options.

In the last example of nonlinear PDE pricing equations, passport options were examined. It was shown that a finite difference discretization of the one factor passport option equation for an arbitrary payoff does not always result in a positive coefficient discretization. Fortunately, a lack of positive coefficients at a few grid locations does not appear to affect the numerical solution. For a convex payoff, it was shown that a positive coefficient discretization can always be obtained. In this way, we are assured that a fully implicit discretization is stable. However, the positive coefficient condition is a weaker condition than monotonicity. The only known results concerning convergence of a numerical solution to the viscosity solution require monotonicity (see Theorem 2.2). It does not appear to be an easy task to design discretizations which are monotone for passport options.

When the payoff of passport options is convex, it can be shown that the position in the underlying asset, q, should always assume one of its extreme values. In these cases, a simple formula gives the value of q. Further, in these cases, all timestepping methods tested converged to the correct solution. However, when the payoff was not convex, or when the general formula was used to compute q, the Crank-Nicolson method either converged to a solution which did not agree with the analytic solution, or diverged. Thus, if quadratic

convergence is desired, Rannacher timestepping should be used.

Unfortunately, quadratic convergence using Rannacher timestepping was not observed for the full two factor equation when testing a non-convex payoff function. In this case, implicit timestepping may be the best approach.

When solving the nonlinear discrete equations, an iterative scheme must be employed. For passport options, using a standard Newton iteration (numerical Jacobian) for this task almost always fails, with the iterative scheme failing to converge. A non-smooth Newton iteration appears to converge for implicit or Rannacher timestepping, but the average number of iterations increases as the grid is refined. On the other hand, the frozen coefficient method always performs well for implicit and Rannacher timestepping, with an average number of iterations just above two. As an added bonus, the frozen coefficient method is relatively simple to implement. Similar results were seen for the uncertain volatility problems. For these reasons, the use of the frozen coefficient method is recommended for solving passport options and two factor uncertain volatility problems.

Finally, it was shown how passport option theory can be used in a trader compensation context. The approach was to show how the payoff can be adjusted to change the distribution of final account balances. With the normal passport option payoff, the trading strategy is to go long when behind and short when ahead when the drift is zero (or when the expected drift equals the risk-free rate). The resulting account balance distribution has two peaks: one for positive account balances and one for negative account balances. By making the payoff turn negative for negative account balances, or by decreasing the payoff slope for positive account balances, these peaks can be shifted or cutoff at prescribed values. While the mean of the distribution cannot change, changing the shape may be desirable for designing trader compensation structures in financial institutions.

6.1 Future Work

The current work could be extended in many ways, including the following:

- It is of particular interest to develop techniques for guaranteeing quadratic convergence of nonlinear option pricing PDEs as the mesh spacing and timestep are reduced, or obtaining a proof that quadratic convergence can not always be guaranteed. It is expected, however, that such a proof would be exceedingly difficult to obtain.

- It would be desirable to prove convergence of iterative schemes without the use of M-matrices. In a similar spirit, more work could be done to establish monotonicity properties of discrete schemes. If monotone schemes prove infeasible, then results for weaker conditions (positive coefficients, total variation diminishing, essentially non-oscillatory, etc.) would be useful. This would appear to require some advances in the basic theory of viscosity solutions.

- In the case of two factor uncertain volatility problems, a small optimization problem must be solved at each node for every timestep. Such a procedure can be extremely computationally expensive. Examining techniques to either speed up the optimization process or to avoid the optimization all together (to perhaps only achieve an approximate solution) would be helpful.

Bibliography

[1] H. Ahn, J. Dewynne, P. Hua, A. Penaud, and P. Wilmott. The end-of-the-year bonus: How to optimally reward a trader. *International Journal of Theoretical and Applied Finance*, 5(2):279–306, 2002.

[2] H. Ahn, A. Penaud, and P. Wilmott. Various passport options and their valuation. *Applied Mathematical Finance*, 6:275–292, 1999.

[3] L. Andersen, J. Andreasen, and R. Brotherton-Ratcliffe. The passport option. *Journal of Computational Finance*, 1(3):15–36, 1998.

[4] L. Andersen and R. Brotherton-Ratcliffe. The equity option volatility smile: An implicit finite-difference approach. *Journal of Computational Finance*, 1(2):5–37, 1998.

[5] M. Avellaneda and R. Buff. Combinatorial implications of nonlinear uncertain volatility models: The case of barrier options. *Applied Mathematical Finance*, 6:1–18, 1999.

[6] M. Avellaneda, A. Levy, and A. Parás. Pricing and hedging derivative securities in markets with uncertain volatilities. *Applied Mathematical Finance*, 2:73–88, 1995.

[7] M. Avellaneda and A. Parás. Managing the volatility risk of derivative securities: the Lagrangian uncertain volatility model. *Applied Mathematical Finance*, 3:21–52, 1996.

[8] G. Barles. Convergence of numerical schemes for degenerate parabolic equations arising

in finance. In L. C. G. Rogers and D. Talay, editors, *Numerical Methods in Finance*, pages 1–21. Cambridge University Press, Cambridge, 1997.

[9] G. Barles and H. M. Soner. Option pricing with transaction costs and a nonlinear Black-Scholes equation. *Finance and Stochastics*, 2:369–397, 1998.

[10] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81:637–659, 1973.

[11] S. Chan. The valuation of American passport options. Unpublished, School of Business, Grainger Hall, University of Wisconsin - Madison, July 10, 1999.

[12] T. H. F. Cheuk and T. C. F. Vorst. Complex barrier options. *Journal of Derivatives*, 4(1):8–22, Fall 1996.

[13] S. S. Clift and P. A. Forsyth. Linear and non-linear iterative methods for the incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 18:229–256, 1994.

[14] T. F. Coleman, Y. Li, and A. Verma. Reconstructing the unknown local volatility function. *Journal of Computational Finance*, 2:77–102, 1999.

[15] M. G. Crandall, H. Ishii, and P. L. Lions. User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society*, 27:1–67, July 1992.

[16] M. G. Crandall and P. L. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 277:1–42, 1983.

[17] F. Delbaen and M. Yor. Passport options. Unpublished, December 1, 1999.

[18] M. A. H. Dempster, A. Eswaran, D. G. Richards, and G. W. P. Thompson. Wavelet based PDE valuation of derivatives. 7th Annual CAP Workshop on Mathematical Finance, December 1, 2000.

[19] Y. d'Halluin, P. A. Forsyth, and K. R. Vetzal. Robust numerical methods for contingent claims under jump diffusion processes. University of Waterloo, Working paper, 2003.

[20] A. K. Dixit and R. S. Pindyck. *Investment Under Uncertainty*. Princeton University Press, 1994.

[21] N. G. Dokuchaev and A. V. Savkin. The pricing of options in a financial market with transaction costs and uncertain volatility. *Journal of Multinational Financial Management*, 8:353–364, 1998.

[22] W. H. Fleming and H. M. Sonar. *Controlled Markov Processes and Viscosity Solutions*, volume 25 of *Applications of Mathematics*. Springer-Verlag, New York, 1993.

[23] P. A. Forsyth and M. C. Kropinski. Monotonicity considerations for saturated-unsaturated subsurface flow. *SIAM Journal on Scientific Computing*, 18(5):1328–1354, 1997.

[24] P. A. Forsyth and K. R. Vetzal. Implicit solution of uncertain volatility/transaction cost option pricing models with discretely observed barriers. *Applied Numerical Mathematics*, 36:427–445, 2001.

[25] A. George and J. Liu. *Computer Solution of Large Sparse Positive Definite Systems*. Prentice Hall, Inc., Englewood Cliffs, N.J., 1981.

[26] V. Henderson and D. Hobson. Passport options wih stochastic volatility. *Applied Mathematical Finance*, 8:97–118, 2001.

[27] V. Henderson, D. Hobson, and G. Kentwell. A new class of commodity hedging strategies: A passport options approach. *International Journal of Theoretical and Applied Finance*, 5(3):255–278, 2002.

[28] S. Heston and G. Zhou. On the rate of convergence of discrete-time contingent claims. *Mathematical Finance*, 10(1):53–75, January 2000.

[29] S. L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies*, 6:327–343, 1993.

[30] J. Hull. *Option, Futures, and Other Derivatives*. Prentice-Hall, Inc., Upper Saddle River, N.J., third edition, 1997.

[31] T. Hyer, A. Lipton-Lifschitz, and D. Pugachevsky. Passport to success. *Risk Magazine*, 10(9):127–131, 1997.

[32] A. Jameson. Positive schemes and shock modelling for compressible flow. *International Journal for Numerical Methods in Fluids*, 20:743–776, 1995.

[33] H. O. Kreiss, V. Thomée, and O. Widlund. Smoothing of initial data and rates of convergence for parabolic difference equations. *Communications on Pure and Applied Mathematics*, 23:241–259, 1970.

[34] H. E. Leland. Option pricing and replication with transaction costs. *Journal of Finance*, 40:1283–1301, 1985.

[35] R. J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser, Basel, 1990.

[36] T. Lyons. Uncertain volatility and the risk free synthesis of derivatives. *Applied Mathematical Finance*, 2:117–133, 1995.

[37] T. Lyons and A. T. Smith. Uncertain volatility. *Risk*, 12:106–109, September 1999.

[38] C. Martini. The uncertain volatility model and American options. INRIA Rocquencourt, MATHFI project, Research report number 3697, May 1999.

[39] R. C. Merton. Theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4:141–183, 1973.

[40] C. W. Oosterlee. On multigrid for linear complementarity problems with application to American-style options. *Electronic Transactions on Numerical Analysis*, 15:165–185, 2003.

[41] J.-S. Pang and L. Qi. Nonsmooth equations: Motivation and algorithms. *SIAM Journal on Optimization*, 3:443–465, 1993.

[42] A. Penaud, P. Wilmott, and H. Ahn. Exotic passport options. *Asia-Pacific Financial Markets*, 6:171–182, 1999.

[43] D. M. Pooley, P. A. Forsyth, and K. R. Vetzal. Numerical convergence properties of option pricing PDEs with uncertain volatility. *IMA Journal of Numerical Analysis*, 23(2):241–267, April 2003.

[44] L. Qi and J. Sun. A nonsmooth version of Newton's method. *Mathematical Programming*, 58:353–367, 1993.

[45] L. Qi and G. Zhou. A smoothing Newton method for minimizing a sum of Euclidean norms. *SIAM Journal on Optimization*, 11:389–410, 2000.

[46] R. Rannacher. Finite element solution of diffusion problems with irregular data. *Numerische Mathematik*, 43(2):309–327, 1984.

[47] S. Shreve and J. Večeř. Options on a traded account: Vacation calls, vacation puts and passport options. *Finance and Stochastics*, 4:255–274, 2000.

[48] D. Sun and J. Han. Newton and quasi-Newton methods for a class of nonsmooth equations and related problems. *SIAM Journal on Optimization*, 7:463–480, 1997.

[49] H. A. Taha. *Operations Research: An Introduction*. Macmillan Inc., 866 Third Avenue, New York, New York 10022, fourth edition, 1982.

[50] D. Tavella and C. Randall. *Pricing Financial Instruments: The Finite Difference Method*. Wiley, New York, 2000.

[51] V. Thomée and L. B. Wahlbin. Convergence rates of parabolic difference schemes for non-smooth data. *Mathematics of Computation*, 28(125):1–13, 1974.

[52] J. Topper. A finite element implementation of passport options. Discussion Paper No. 224, ISSN 0949-9962, University of Hannover, Germany, 2001.

[53] J. Topper. Uncertain parameters and reverse convertibles. *RISK*, pages 4–7, January 2001.

[54] J. Topper. Worst case pricing of rainbow options. Discussion paper Number 217, Department of Economics, University of Hannover, version October 12, 2001.

[55] H. A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. 13:631–645, 1992.

[56] T. Vargiolu. Existence, uniqueness and smoothness for the Black-Scholes-Barenblatt equation. Università di Padova, Department of Pure and Applied Mathematics, Rapporto Interno n. 5, July 2001.

[57] J. Večeř. A new PDE approach for pricing arithmetic average Asian options. *Journal of Computational Finance*, 4(4):105–113, 2001.

[58] L. B. Wahlbin. A remark on parabolic smoothing and the finite element method. *SIAM Journal on Numerical Analysis*, 17:33–38, 1980.

[59] P. Wilmott. *Paul Wilmott on Quantitative Finance*, volume 1. John Wiley and Sons Ltd, Chichester, England, 2000.

[60] P. Wilmott. *Paul Wilmott on Quantitative Finance*, volume 2. John Wiley and Sons Ltd, Chichester, England, 2000.

[61] H. Windcliff, P. A. Forsyth, and K. R. Vetzal. Shout options: A framework for pricing contracts which can be modified by the investor. *Journal of Computational and Applied Mathematics*, 134:213–241, 1999.

[62] H. Windcliff, P. A. Forsyth, and K. R. Vetzal. Analysis of the stability of the linear boundary condition for the Black-Scholes equation. University of Waterloo, Working paper, 2003.

[63] O. C. Zienkiewicz and R. Taylor. *Finite Element Method*. McGraw-Hill, 4th edition, 1991.

[64] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique. *International Journal for Numerical Methods in Engineering*, 33:1331–1364, 1992.

[65] O. C. Zienkiewicz and J. Z. Zhu. Automatic directional refinement in adaptive analysis of compressible flow. *International Journal for Numerical Methods in Engineering*, 37:2189–2210, 1994.

[66] R. Zvan. *The Numerical Solution of Two-Factor Option Pricing Models*. PhD thesis, University of Waterloo, 2000.

[67] R. Zvan, P. A. Forsyth, and K. R. Vetzal. A finite volume approach for contingent claims valuation. *IMA Journal of Numerical Analysis*, 21(3):703–731, 2001.

[68] R. Zvan, P. A. Forsyth, and K. R. Vetzal. Negative coefficients in two factor option pricing models. University of Waterloo, Working paper, 2003.