Linear and Non-linear Monotone Methods for Valuing Financial Options Under Two-Factor, Jump-Diffusion Models

by

Simon Sivyer Clift

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

Waterloo, Ontario, Canada, 2007

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

The evolution of the price of two financial assets may be modeled by correlated geometric Brownian motion with additional, independent, finite activity jumps. Similarly, the evolution of the price of one financial asset may be modeled by a stochastic volatility process and finite activity jumps. The value of a contingent claim, written on assets where the underlying evolves by either of these two-factor processes, is given by the solution of a linear, two-dimensional, parabolic, partial integro-differential equation (PIDE). The focus of this thesis is the development of new, efficient numerical solution approaches for these PIDE's for both linear and non-linear cases.

A localization scheme approximates the initial-value problem on an infinite spatial domain by an initial-boundary value problem on a finite spatial domain. Convergence of the localization method is proved using a Green's function approach. An implicit, finite difference method discretizes the PIDE. The theoretical conditions for the stability of the discrete approximation are examined under both maximum and von Neumann analysis. Three linearly convergent, monotone variants of the approach are reviewed for the constant coefficient, two-asset case and reformulated for the non-constant coefficient, stochastic volatility case. Each monotone scheme satisfies the conditions which imply convergence to the viscosity solution of the localized PIDE.

A fixed point iteration solves the discrete, algebraic equations at each time step. This iteration avoids solving a dense linear system through the use of a lagged integral evaluation. Dense matrix-vector multiplication is avoided by using an FFT method. By using Green's function analysis, von Neumann analysis and maximum analysis, the fixed point iteration is shown to be rapidly convergent under typical market parameters. Combined with a penalty iteration, the value of options with an American early exercise feature may be computed.

The rapid convergence of the iteration is verified in numerical tests using European and American options with vanilla payoffs, and digital, one-touch option payoffs. These tests indicate that the localization method for the PIDE's is effective. Adaptations are developed for degenerate or extreme parameter sets. The three monotone approaches are compared by computational cost and resulting error. For the stochastic volatility case, grid rotation is found to be the preferred approach.

Finally, a new algorithm is developed for the solution of option values in the non-linear case of a two-factor option where the jump parameters are known only to within a deterministic range. This case results in a Hamilton-Jacobi-Bellman style PIDE. A monotone discretization is used and a new fixed point, policy iteration developed for time step solution. Analysis proves that the new iteration is globally convergent under a mild time step restriction. Numerical tests demonstrate the overall convergence of the method and investigate the financial implications of uncertain parameters on the option value.

Acknowledgments

In 2002, after a number of years in the financial industry, I had planned to take a one year break from my career. I wanted to refocus my life and start along a better direction. The break has been a little longer than planned. I have become a very happy person these five years and, as much as one must find one's own happiness in life, the people around me have made that easy.

I was pleased by the chance to work again with my supervisor, Peter Forsyth. He opened a position for me and, with that, a chance to fulfill a goal that I have had since I was a child to obtain a doctorate. Peter's work is always rigorous and of outstandingly high quality. It has been always rewarding, if often demanding, for me to try and meet the challenge of his high standards. I am very grateful to him for this opportunity and for his time, guidance, patience and humour.

Ken Vetzal is owed one case of good red wine for his good advice and support. I also thank Phelim Boyle for his occasional, but always apt and timely, words of wisdom.

Yann d'Halluin's work was the starting point for this thesis, along with the work of Dave Pooley, Rob Zvan and Heath Windcliffe. I thank them for their work, which formed such a good foundation on which to build. I thank my fellow students with whom I have studied concurrently, Amelie Belanger and Shannon Kennedy, for their companionship and many worthwhile discussions.

The David R. Cheriton School of Computer Science at the University of Waterloo has given me a chance to learn, work and teach again with its stellar faculty. It has been a great pleasure to work with Justin Wan, Bruce Simpson, Steven Mann, Craig Kaplan and Gladimir Baranoski as a teaching assistant for their courses.

My parents, Bruce and Tina, as well as my sister Elaine, have been behind me since day one and always a source of encouragement. Thanks especially to my father and mother for having me spend my childhood in Deep River, which helped give me a life-long love of learning and science.

Jane Peryer gave up her home and career in New Zealand to be with me, learned to paddle white water in a kayak and canoe, accepted my proposal of marriage on the rapids of the Elora Gorge, married me and gave me my first child, Hazel Peryer Clift, all while I worked on this degree. With all that, all the desires were fulfilled which I had for my life when I began this degree. She, along with her parents Bryan and Jennifer, and Hazel have my profound thanks for all they have done for me and always have my love.

For Jane, Hazel and the bump.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Originally, an option contract in the financial markets established a right, but not an obligation, to buy or sell an asset for an agreed "exercise" price at a future date. Today, the term "option" implies a wide class of financial contracts which have a value contingent on the evolution of the uncertain price of some risky underlying asset. From the seller's perspective, an option can be an opportunity to create a valuable service, comparable to writing an insurance contract. For the contract to have value to the seller, there must be a clear strategy to manage the risks entailed in its obligations. From the buyer's perspective, an option could be, for example, a means of reducing risk or a vehicle for speculation.

The Black–Scholes model [13] for the valuation of financial options is a fundamental result in the field and revolutionized how the financial community approached the contracts. This model assumes that the future price of a single risky asset, which underlies an option, can be described by geometric Brownian motion with drift. Brownian motion as an asset price model was not new in itself, having first been proposed by Bachelier [7] along with methods for estimating the present value of assets under this assumption. Black and Scholes showed, in this idealized setting, how a seller could use a process called "delta hedging" to neutralize the risk of an option by adding it to a portfolio containing a risk-free cash deposit and a dynamically adjusted quantity of the underlying asset. This construction supplied a pricing model which resulted in the same value for both buyer and seller, unifying the two perspectives.

Although it has many elegant theoretical properties and continues to serve as an important benchmark, simple drift and diffusion is often viewed as unable to capture significant risks seen in real markets. The Black–Scholes model must be extended to incorporate any additional risks, such as possible shocks and surprises in the market. This raises the question of determining the fair value of an option contract exposed to these new risks, which in turn leads to the focus of this thesis: creating flexible, efficient algorithms with which to compute that value numerically.

The fair value of an option under the Black–Scholes model can be expressed as the solution to a partial differential equation (PDE), where the asset price and the time to expiry are the independent variables. The underlying asset can also be modeled to allow sudden changes, which generate a finite number of jumps of a random magnitude from a known distribution and which occur at random, Poisson-distributed points in time. When the underlying asset evolves by such jumps as well as by diffusion, the option value may be expressed as the solution of a parabolic, partial, integro-differential equation (PIDE).

The single asset underlying the option value in the Black-Scholes model may be replaced by a pair of assets. An option may then pay out a value which is a function of the two assets, such as the maximum or minimum price, as introduced by Stulz [103]. The assets may be modeled with jumps, arriving at a finite rate, in addition to diffusion. The value of the option is then given by the solution to a two-dimensional, parabolic PIDE.

Rather than the constant volatility Brownian motion of the Black-Scholes model, the price an asset may be modeled with a magnitude of volatility which is itself a stochastic process. Such "stochastic volatility" models are intended to capture more realistic market behaviour for a single asset. An important example is that of Heston [60], where volatility follows a mean-reverting diffusion. Again, a finite number of jumps in price as well as jumps in the volatility may be added to the diffusion. In this case the value of the option is, again, the solution to a two-dimensional, parabolic PIDE where the spatial dimensions are asset price and volatility.

This thesis formulates a numerical approach and supporting theory for solving the two-dimensional PIDE's arising from these two problems. The linear case, when the parameters of the market model are fixed, is considered first and then a non-linear case, which arises when parameters are known only to within a deterministic range.

1.1 Motivation

Many techniques have been developed to compute the fair value of an option contract and a recent survey was published by Broadie and Detemple [21]. The ones that are particularly relevant to this thesis are reviewed at the start of each main chapter. A semi-analytic solution is available to value European options under many stochastic volatility models with jumps [41]. When a contract depends only on the final asset state and the option payoff, then an integration may be used to determine the value of a single contract with a given exercise price, current underlying asset state and expiry time. Monte Carlo methods integrate forward in time and often have problems computing option values that depend on optimal decision making.

The finite difference approach studied in this thesis computes all option values over a finite, discrete domain of expiry times and asset states. For even the minor, additional complications of American options, which can be exercised for their payoff at any time, or for barrier options, which pay when the underlying asset reaches a given barrier price at any time, the finite difference approach has distinct advantages. A robust, efficient finite difference method is a practical choice for many contracts where the value is not a simple function of the underlying at the expiry time.

1.1.1 Two-Asset Jump Diffusion

Spread options [25], for example on the price difference between the input and product commodities of an industrial process, are traded on exchanges such as the New York Mercantile Exchange (NYMEX). Such relatively simple, two-asset options allow industrial operations to

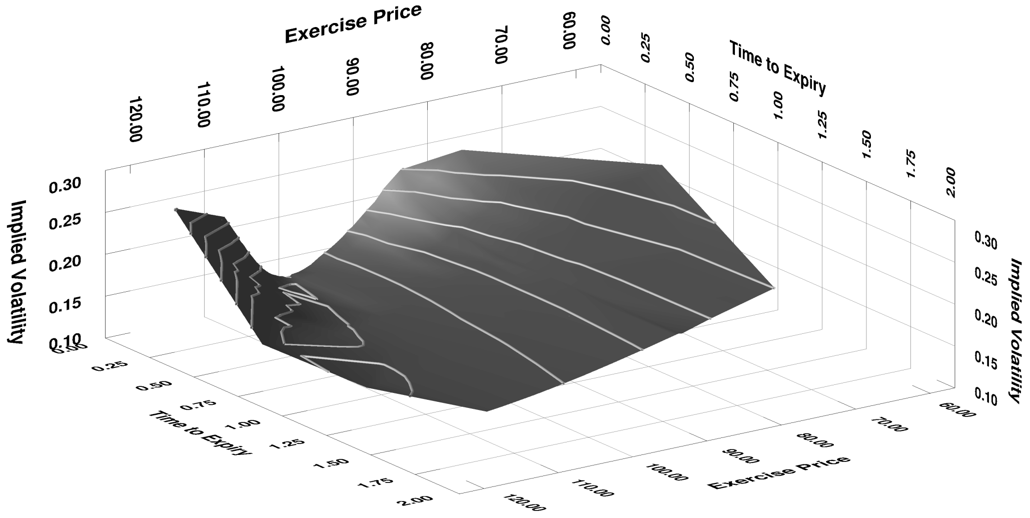

Figure 1.1.1: Volatility parameter for a Black–Scholes model (implied volatility) that generates the prices for options on the S&P 500 index quoted on April 14, 2005. Note that this implied volatility surface shows a "smile" for the short-term options and a "skew" for options with a longer expiry time. Such features are well replicated by prices generated from stochastic volatility models which also permit jumps.

hedge their market risk, which is a factor beyond their control. Options on more complex functions of two assets, such as the maximum of cash and two assets, are commonly known as "rainbow options". These are considered exotic and are not traded on exchanges, but rather between interested parties directly. Such options are of interest when, for example, an investor wishes to have or remove a risk on two related companies, such as two companies which form a duopoly.

It is easy to motivate the extension of the two-asset, diffusion model to a jump diffusion model: it is a generally accepted fact that financial markets do not evolve by perfect Brownian motion. It is an intuitively appealing idea that markets respond to unexpected, discrete events by sudden price changes. Consider, for example, weather events that affect electricity consumption or natural gas prices: these are a motivation for the trade of spread options at NYMEX. In the example of the duopoly, the awarding of a major contract to one company in favour of another, or a general setback to an industry overall, are economic events that could generate a price jump. Cont [33] provides an overview of how price jumps are an important factor in real markets. The two-asset, jump-diffusion models used as working examples in this thesis extend the one-asset, jump-diffusion processes of Merton [86] and Kou [72]. The resulting two-asset model permits jump events to be correlated between the two prices, positively or negatively.

1.1.2 Stochastic Volatility with Jumps

Numerous recent studies [8, 28, 55, 61, 99] justify the idea that stochastic volatility is an important factor in the evolution of market prices. Again, this idea is intuitively appealing: a sudden drop in the price of an equity asset may be associated with an increase in the uncertainty about the price and hence with an increase in volatility. Similarly, an increase in the price of a company's stock may be associated with a more certain financial outlook and hence a decrease in the volatility. Such effects manifest themselves in the surface of implied volatilities.

The implied volatility surface is formed by the Black–Scholes volatility parameter value required to replicate a set of quoted option prices over a range of exercise prices and expiry dates. The "smile" or "skew" of implied volatility refer to the tendency of this surface to have a positive second derivative or (typically) negative first derivative with respect to the option exercise price. Such a surface is shown in Figure 1.1.1 for the S&P 500 index, showing a smile in the implied volatility for short-term options and a skew for long-term options. Mathematically, these effects are well replicated using a stochastic volatility model, particularly when that model permits jumps in price and variance as well as diffusion (e.g. [100]). Calibration studies (e.g. [20, 24, 43, 44, 100]) confirm that such a process is a good description of the behaviour of a number of financial markets for which options are traded, particularly in equity markets.

One might hypothesize that there are more independent variables simultaneously affecting a market than those already noted, which may extend the numerical solution domain to higher dimensions. Regime switching approaches (e.g. [39]) can fall into this category when they introduce the notion of additional discrete or continuous states of market behaviour. Higher dimensional stochastic volatility models exist which propose, for example, that the rate of jump arrivals should be its own stochastic process [100]. Two drawbacks to the use of more dimensions exist. The first is the difficulty of computing a solution: in higher dimensions finite-difference methods lose their computational advantage over other methods of numerical integration such as Monte Carlo methods. The second is the question of measuring the initial state: only the asset price is considered directly observable in a market. Other state variables must be inferred. A two-factor diffusion model extended with jump processes is less affected by these problems yet provides a larger set of parameters and hence more flexibility with which to capture complex market behaviour.

1.1.3 Uncertain Parameter Models

A calibration approach usually underlies the decision of which asset price model to use for a particular market (e.g. [24]). Calibration determines the parameters for a given model and is a process that can be affected by the choice of financial data, the measure of goodness of fit of a model to that data and even by the initial guess used in the fitting algorithm [59]. Therefore, no matter which model is selected, the parameters will always be uncertain to some degree.

This leads to a final, practical problem studied in this work: valuing an option in a hedge portfolio when the actual market evolution follows the best-case or worst-case scenario allowed by a deterministic range of parameters. This is of particular interest as one explanation of the difference between the value of an option that a buyer will pay and that a seller will demand.

Both will wish to trade using the value of the option under their own “worst-case” scenario, which leads to the seller computing a higher, and the buyer a lower, option value. This problem is an example of one that can be expressed as a non-linear, Hamilton-Jacobi-Bellman PIDE, which is closely related to the form of the linear PIDE used to value options when parameters are known.

1.2 Summary Contributions

The main contributions made in this thesis to the study of financial options may be summarized as follows and focus on improving the algorithms by which options can be valued. The calibration techniques required to match models to markets are not addressed although, where such studies have been done, some of the resulting models are used in the numerical demonstrations. Detailed notes on the contributions are given in each main chapter.

- A method is presented for localizing the two-factor, jump-diffusion PIDE’s for option valuation from an initial-value problem on an infinite spatial domain to an initial–boundary value problem on a finite domain. New theorems, based on Green’s function analysis, are presented to bound the error due to this operation. The localized problem is discretized with a finite-difference approach and the stability of the discrete problem analyzed. A fixed point iteration algorithm which solves the discrete, algebraic equations at each time step is extended from previous work in the one-dimensional case to the two-dimensional case. The convergence of the fixed point iteration is proved analytically for a number of key cases. Both the convergence of the overall approach and the fixed point iteration are demonstrated numerically.

- Fully monotone variants of the discretization are formulated for the stochastic volatility case, which has spatially dependent coefficients, by modifying the standard approach for the constant coefficient, two-asset case. These satisfy the key conditions which imply convergence to the viscosity solution of the problem. The cost and error of each is compared under the fixed point solution algorithm to determine the most favourable approach for valuing a variety of option contracts under real market conditions. Of the approaches studied, grid rotation is the best in terms of cost and error.

- Using the two-asset case as a working example, a new fixed point, policy iteration is formulated for the solution of the discrete, non-linear algebraic equations for a time step of the Hamilton-Jacobi-Bellman PIDE for the case when the underlying process has jump parameters which are known only within a deterministic range. The convergence of the new algorithm is proved and shown to be faster than the existing approach. Numerical demonstrations of the new algorithm compute the best-case and worst-case value of a two-asset option under uncertain jump parameters, showing the method is robust, accurate and converges at the rate predicted by theory.

1.3 Outline and Overview

Chapters 2 through 4 contain the detailed development, theoretical proofs and numerical demonstrations in the thesis. Each starts with an overview of its contents, a review of key pieces of prior work, and a synopsis of the specific contributions of the chapter to the field of study. Chapter 5 summarizes the conclusions and results of the thesis.

In Chapter 2 the basic theory and algorithms are developed for localizing the infinite domain problem, discretizing the PIDE and solving the discrete algebraic equations at each time step. The problem of valuing options under two-asset, jump-diffusion is studied as the working example. Numerical demonstrations confirming the theoretical results are presented, based on realistic, but synthetic financial markets.

In Chapter 3 the Heston stochastic volatility model with finite-activity jumps is described. The solution method is adapted from that used for two-asset options. This chapter emphasizes monotone approaches, demonstrating that the methods reviewed for constant coefficient, two-asset problems also apply to these stochastic volatility problems, which have spatially variable coefficients. Numerical demonstrations compare the cost and error in each monotone method under a set of market cases that are drawn from real financial studies. These market cases demonstrate the numerical approach under some important cases where parameters approach or take on limiting values.

Chapter 4 studies the case where the option parameters, specifically those governing the jump portion of the asset price evolution, are known only within a deterministic range. The new fixed point policy iteration is developed, which advances each time step of the resulting, discretized, Hamilton-Jacobi-Bellman PIDE. The convergence of the iterative algorithm is proved, and demonstrated for a two-asset contract where the selection of the parameters that generate the best- and worst-case scenarios is clearly non-trivial.

Appendices A and B contain a number of fundamental, standard equations for reference. Appendices C through F contain the details of some of the longer proofs. Notation will be defined as it is introduced; for reference, Appendix G summarizes that notation.

Chapter 2

The Basic Theory and Approach using Two-Asset Jump Diffusion Models

2.1 Introduction

This chapter establishes the general approach to the numerical solution of the two-dimensional, parabolic PIDE's studied in this thesis. The simplest of these three problems is the PIDE for the value of a two-asset option, which may be formulated to have constant coefficients and no degeneracies. This first practical case is studied in detail so that the following two chapters may focus on the adaptations to the approach that are required to deal with the remaining two problems.

The localization technique introduced in this chapter is used to approximate the spatially infinite PIDE with a spatially finite problem. This method requires the PIDE be computed in a core domain nested inside a larger finite region. The integral term of the PIDE is a correlation and is non-local; using a finite domain of integration is equivalent to assuming a zero solution value outside it. Thus, in the larger region, an approximate option value is computed which permits a more accurate computation of the integral term of the PIDE in the core. The sizes of the two domains are specified using two error tolerance parameters. Although this approach is proved convergent as the finite domains increase in size, without reference to any solution method, it fits well with the numerical method used to compute the correlation.

The discretization of the finite-domain problem is formed using standard finite difference, interpolation and quadrature approaches. The algebraic identity relating a discrete, correlation integral to the fast Fourier transform (FFT) is exploited to permit the fast computation of what would otherwise be a dense matrix-vector multiply. The conditions are reviewed under which the discretization is monotone. Stability is examined in detail under both the l_∞ and l_2 norms. Stability under the l_∞ norm, along with consistency and monotonicity, is a key condition for convergence to the viscosity solution. In Chapter 4 these considerations become crucial to ensure a correct solution in the non-linear case.

The fixed point iteration for the solution of the algebraic equations for a single time step is described in detail in this chapter and proved convergent in a number of key cases. It permits the use of an implicit time step, which does not suffer from step size restrictions due to stability considerations, without actually having to compute a dense linear system solution. The FFT technique replaces the dense matrix-vector multiply that would otherwise be required for the integral term. The fixed point iteration may be combined with a penalty iteration to compute the value of options with an American, early-exercise feature. This algorithm will carry over to the stochastic volatility case in Chapter 3 with no changes, but requires important changes in Chapter 4 to deal with uncertain parameters.

It is worth noting that the technique for solving the jump-diffusion problem can be implemented as an extension to a conventional two-factor, finite-difference approach for the pure diffusion, PDE case. The fixed point iteration may be programmed as a variation on an existing penalty iteration method for American options. The algorithm introduces two FFT's at each iteration of the solution method for the discrete algebraic equations at each time step.

The numerical demonstrations in this chapter are based on two synthetic markets. The first uses bi-variate, Normally distributed jumps [86] and the second uses bi-variate exponentially distributed jumps [72, 80]. Convergence of the localization technique is demonstrated numerically by solving over progressively larger domains. The convergence with respect to grid and time step spacing is demonstrated for the case of a conventional finite-difference grid and a monotone finite difference discretization. The fixed point iteration is demonstrated to be convergent, both for European options and when it is combined with a penalty iteration to solve for the price of an American option.

2.1.1 Previous Work

The study of two-asset option valuation, under diffusion alone or under diffusion with jumps, has been undertaken by a number of researchers. The following work led directly to the new developments presented in this chapter.

A semi-analytic valuation method for options under the broad class of affine price processes was developed by Duffie et al. [41] which computes the integral of Fourier transformed quantities. It applies to European options on one asset under a variety of processes, which include one asset jump-diffusion. This was simplified by Lewis [78, 79] and further development and analysis was carried out by Lee [76]. Da Fonseca et al. extend this for multiple assets [46], an approach used in this work as a benchmark valuation technique. Carr and Madan [26] developed a similar approach which uses a Fast Fourier Transform and Dempster and Hong [35] adopted such an approach for spread options. These approaches can be related to wavelet methods, developed for implicit solutions of single factor infinite activity jump-diffusion problems with European options by Matache et al. [82, 83, 84].

Binomial lattice methods are the equivalent of explicit finite difference methods [113] and an approach to the one-dimensional jump-diffusion problem using this technique is discussed by Amin [4]. Recent work in finite difference approaches by Briani et al. [18, 19] and Cont et al. [34] use explicit time stepping for the integral operator introduced by the jump process. Two-asset American claims under jump diffusion were priced using a Markov chain approach by Martzoukos [81], an approach which can be viewed as essentially an explicit finite difference method. The jump terms were handled using an extension of the method in [4]. For single asset problems, Andersen and Andreasen [5] developed an operator splitting approach for European options which was unconditionally stable and second order in time. An implicit, finite difference approach for single asset options was explored by d'Halluin, Forsyth, et al. [37, 38]. This method was demonstrated to be quadratically convergent versus grid spacing and time step size. A similar approach, which uses an iterative method to solve the implicit discretized PIDE and which also uses an FFT to carry out the dense matrix-vector multiply, was developed by Almendral and Oosterlee [2].

Pham [91] developed a modern, viscosity solution approach to the valuation of American options under diffusion processes with jumps. Approximations, such as that of Whaley and Barone-Adesi [107] in the pure diffusion case, have also been made for jump-diffusion cases, for instance that of Mulinacci [88]. However, the free boundary problem arising from the American early exercise constraint is usually solved numerically. Zhang [110] developed a semi-implicit approach for American options using a traditional linear complementarity solver for constant diffusion processes with Normally distributed jumps. Zvan, Forsyth et al. [48] developed a fully implicit approach that uses a penalty method within a finite difference approach, which was further developed [37, 38] to an implicit method for American options under one-factor jump-diffusion models. An alternative is the integral equation approach of Tsavalis et al. [104] or Chiarella et al. [29], however, these require assumptions about the form of the free boundary, which penalty methods do not.

The two-asset, correlated Brownian motion model [103] is a simple extension of the one-asset Black–Scholes model [13, 85]. In this chapter, the finite difference jump-diffusion work of [37, 38] is adapted to the work on two-factor option valuation of Zvan, Forsyth et al. [48, 111, 112] to produce a similarly quadratically convergent method.

This new two-asset technique retains the advantages of being able to price options with general types of payoffs and barriers for American as well as European options. The approach of Briani et al. [18] for a two-dimensional case with jumps has similarities to the method presented here. However, the method developed in this thesis may be used in a fully implicit form and also computes the jump terms implicitly. It can be used with any distribution of jumps that obeys mild, conventional restrictions, as well as discontinuous, exponential distributions, and also computes without splitting the linear system along axes, so can solve correlated two-dimensional diffusion problems.

2.1.2 Overview of this Chapter

Section 2.2 reviews the equations governing option valuation over two assets with jump-diffusion. The localization of the equations from an infinite to a finite domain, along with control methods for the resulting error, are discussed in Section 2.3. The discretization method discussed in Section 2.4 is studied in Section 2.5 to determine the theoretical conditions for stability and monotonicity. Section 2.6 describes the fixed point iteration used to advance the solution by one time step and provides theory that shows it should be rapidly convergent under normal parameter ranges. The combination of the fixed point iteration with a penalty iteration for valuing American options is also discussed. Section 2.7 gives a number of numerical examples to demonstrate the techniques and confirm the theory presented in the chapter.

2.1.2.1 Synopsis of Contributions in this Chapter

The following lists the specific contributions to the field of option valuation made in this chapter.

- A localization approach is presented in Section 2.3 for the two-asset, jump-diffusion case.

  Theorems 2.3.7 and 2.3.8 use a Green's function approach to demonstrate that the localization of the option valuation PIDE's from an infinite initial value problem, to a finite, initial–boundary value problem has bounded error when non-zero boundary conditions are present. These present sufficient conditions for convergence with non-zero, non-constant boundary conditions.

  Numerical demonstrations in Section 2.7.2 confirm the effectiveness of the approach.

- Using von Neumann analysis, the scheme used in this study is shown to be unconditionally stable in the l_2 norm for Crank–Nicholson time-stepping in Theorem 2.5.11.

- The discrete 1D Fourier transform integral method of d'Halluin [36] is extended in Section 2.4.3 to two-asset jump-diffusion. An alignment method for the jump PDF on the discrete Fourier transform grid, discussed in Section 2.4.3.4, simplifies the use of exponential jump distributions and improves on the results in [36], which suffered impaired convergence for such distributions.

- The convergence of the functional form of the fixed point time step iteration under the l_∞ norm for the semi-discretized equations is demonstrated in Theorem 2.6.1 by Green's function analysis. Using an analogous formulation to the Green's function approach, convergence and the rate of convergence of the discrete fixed point iteration is shown in Theorem 2.6.2 for the case when the iteration matrix is an M-matrix. By using von Neumann analysis, Theorem 2.6.3 demonstrates that the convergence rate of the fixed point iteration under the l_2 norm is similar to that under the l_∞ norm.

- Further numerical demonstrations in Sections 2.7.3 through 2.7.5 show, for both European and American contracts, the convergence rate both of the fixed point iterative algorithm and of the overall approach. Typically 3 to 5 fixed point iterations, or combined fixed point and penalty iterations, are required for each time step. The overall approach is demonstrated to be up to quadratically convergent in grid spacing and time step for the linear problem. The convergence under a monotone spatial discretization scheme is also shown.

2.2 Governing Equations

This section presents the equations governing the value of an option contract under two-asset jump-diffusion. The value will be computed after transforming the asset prices from price to logarithm of price (log-price) scaling. For reference, some equations are given in price scaling.

2.2.1 The Finite Activity Jump Diffusion Model

In this chapter, the value of an option is computed that uses, as independent variables, the time to expiry τ and two asset prices S_1 and S_2 where

$$\begin{aligned} t &\in [t_0, t_0 + T] \\ \tau &\in [0, T] \\ (S_1, S_2) &\in [0, \infty] \times [0, \infty] \ . \end{aligned} \tag{2.2.1}$$

Time to expiry is $\tau = T - t$, where t is time in the forward direction. The contract of duration T is written at $t = t_0$ with $\tau = T$ time remaining, when the two initial asset prices may be observed in the market. Appendix A.1 describes the risk neutral price processes assumed for (S_1, S_2). A logarithmic transform is applied to the asset prices

$$\begin{aligned} (x_1, x_2) = (\log(S_1), \log(S_2)) &\in \Omega_\infty \\ \Omega_\infty = [-\infty, \infty] &\times [-\infty, \infty] \ . \end{aligned} \tag{2.2.2}$$

The asset prices in logarithmic scaling evolve by a correlated, constant coefficient, Brownian motion and correlated, finite activity jumps where the jumps are independent of the diffusion. The value of a European option $U(x_1, x_2, \tau)$ over $\Omega_\infty \times [0, T]$ is found by taking expectations under the risk-neutral price process (see Appendix A.1 [18, 33]) and solving

$$\begin{aligned} \frac{\partial U}{\partial \tau} &= \mathcal{L}U + \lambda \mathcal{H} U \\ U(x_1, x_2, 0) &= \mathcal{I}(x_1, x_2) \ . \end{aligned} \tag{2.2.3}$$

Linear operator $\mathcal{L}$ represents the differential and source terms due to the Brownian motion and discounting by r and is defined in this chapter as

$$\begin{aligned} \mathcal{L}U = &\left(r - \frac{{\sigma_1}^2}{2}\right) \frac{\partial U}{\partial x_1} + \left(r - \frac{{\sigma_2}^2}{2}\right) \frac{\partial U}{\partial x_2} \\ &+ \frac{{\sigma_1}^2}{2} \frac{\partial^2 U}{\partial {x_1}^2} + \rho \sigma_1 \sigma_2 \frac{\partial^2 U}{\partial x_1 \partial x_2} + \frac{{\sigma_2}^2}{2} \frac{\partial^2 U}{\partial {x_2}^2} - rU \end{aligned} \tag{2.2.4}$$

with constant coefficients

$\sigma_1, \sigma_2 > 0$	the volatility of the Brownian motion of the two assets,
$\lvert\rho\rvert \leq 1$	the correlation between the diffusion of x_1 and x_2, and
$r \geq 0$	represents a risk-free rate of return.

For simplicity, dividends are not included in Equation 2.2.4. The operator $\lambda\mathcal{H}$ represents the effects of finite activity price jumps in two assets together, generated by a Poisson process. In this chapter,

$$\begin{aligned}\lambda\mathcal{H}U = \lambda \iint_{-\infty}^{\infty} g(J_1,J_2)\Big[&U\left(x_1+J_1,x_2+J_2,\tau\right) - U \\ &- \left(e^{J_1}-1\right)\frac{\partial U}{\partial x_1} - \left(e^{J_2}-1\right)\frac{\partial U}{\partial x_2}\Big]\, dJ_1\, dJ_2 \\ g = g(J_1,J_2) \in \mathbb{R}&\end{aligned} \tag{2.2.5}$$

where the mean jump arrival rate is $\lambda > 0$ and the jump magnitudes (J_1, J_2) are distributed according to a probability density function $g(J_1, J_2)$. $\mathcal{I}(x_1, x_2)$ represents initial conditions, usually the option payoff, which are discussed in Section 2.2.3 below.

This chapter proceeds under the standard assumption that $g(J_1, J_2)$ is independent of (x_1, x_2, τ). It is also assumed that $g(J_1, J_2)$ satisfies the technical conditions of [54] §II.1.2 Definition 1.6 (see also [33] Proposition 3.18) in particular, that the integrals for the second, third and fourth term of $\mathcal{H}U$ may be written, respectively, as

$$\iint_{-\infty}^{\infty} g(J_1,J_2)\, U\, dJ_1\, dJ_2 = U \tag{2.2.6}$$

$$\iint_{-\infty}^{\infty} g(J_1,J_2)\left[\left(e^{J_1}-1\right)\frac{\partial U}{\partial x_1}\right] dJ_1\, dJ_2 = \kappa_1 \frac{\partial U}{\partial x_1} \tag{2.2.7}$$

$$\iint_{-\infty}^{\infty} g(J_1,J_2)\left[\left(e^{J_2}-1\right)\frac{\partial U}{\partial x_2}\right] dJ_1\, dJ_2 = \kappa_2 \frac{\partial U}{\partial x_2} \tag{2.2.8}$$

$$\kappa_1, \kappa_2 \in \mathbb{R}$$

$$|\kappa_1|, |\kappa_2| < \infty$$

where values κ_1, κ_2 correct for the mean drift in each asset due to jumps. The first term of $\mathcal{H}U$ is written separately as

$$\begin{aligned}\mathcal{J}U &= \iint_{-\infty}^{\infty} g(J_1,J_2)\, U\left(x_1+J_1,x_2+J_2,\tau\right)\, dJ_1\, dJ_2 \\ &= \iint_{-\infty}^{\infty} g(J_1-x_1,J_2-x_2)\, U\left(x_1,x_2,\tau\right)\, dJ_1\, dJ_2\end{aligned} \tag{2.2.9}$$

which are equivalent forms of a correlation product. Specific formulations of $g(J_1, J_2)$ with Normal and exponentially distributed jumps, which are analogous to the one-dimensional jump models of Merton [86] and Kou [72] respectively, are given in Appendix B.

An American option may be exercised for its terminal payoff at any time. The American option value is the solution to a linear complementarity problem [91, 110]

$$\frac{\partial U}{\partial \tau} \geq \mathcal{L}U + \lambda\mathcal{H}U \tag{2.2.10}$$

$$U \geq \mathcal{I}(x_1,x_2) \tag{2.2.11}$$

$$U(x_1,x_2,0) = \mathcal{I}(x_1,x_2) \tag{2.2.12}$$

where at least one of Equations 2.2.10 or 2.2.11 must hold with equality.

2.2.2 Notes on the PIDE in Price Scaling

In "price scaling", the value of the option $\tilde{U}(S_1, S_2, \tau)$ in time to expiry τ and two space dimensions is determined by solving a PIDE analogous to Equation 2.2.3. It is defined on $(S_1, S_2, \tau) \in [0, \infty] \times [0, \infty] \times [0, T]$. Note the analogue of the differential term due to Brownian motion, which in price scaling becomes

$$\tilde{\mathcal{L}}\tilde{U} = r\,S_1 \frac{\partial U}{\partial S_1} + r\,S_2 \frac{\partial U}{\partial S_2} + \frac{{S_1}^2 {\sigma_1}^2}{2} \frac{\partial^2 U}{\partial {x_1}^2} + S_1 S_2\, \rho \sigma_1 \sigma_2 \frac{\partial^2 U}{\partial x_1 \partial x_2} + \frac{{S_2}^2 {\sigma_2}^2}{2} \frac{\partial^2 U}{\partial {x_2}^2} - rU\ . \quad (2.2.13)$$

This has zero differential coefficients normal to the natural lower boundary of the problem at $S_1 = 0$ or $S_2 = 0$. These lines correspond to $x_1 = -\infty$ and $x_2 = -\infty$ in log-price scaling, a feature which shall be exploited when the problem is localized to a finite domain.

2.2.3 Contract Types and Initial Conditions

The numerical examples in this chapter value various option contracts, which differ in their payoffs. The PIDE initial conditions $\mathcal{I}(x_1, x_2)$ are defined by the terminal payoff equation of the option contract and a strike price K. This chapter uses, as the working examples,

$$\mathcal{I}(x_1, x_2) = \begin{cases} \max\big(0, K - \min(e^{x_1}, e^{x_2})\big) \\ \qquad\qquad = \max\big(0, K - \min(S_1, S_2)\big) & \text{a put on the minimum, or} \\ \\ \max\big(0, \max(e^{x_1}, e^{x_2}) - K\big) \\ \qquad\qquad = \max\big(0, \max(S_1, S_2) - K\big) & \text{a call on the maximum} \end{cases} \quad (2.2.14)$$

of the two underlying assets. These examples are not specific to the numerical solution technique, which can be used for any payoff that can be bounded linearly (see Assumption 2.3.1 below).

2.3 Localization

The numerical solution technique requires truncating the infinite domain at finite boundaries. In this section, equation localization is discussed along with the associated convergence issues in the context of pricing European options (Equation 2.2.3). The same approach is used for localizing American options, which can be justified in this case by numerical experiment.

The localization method which follows is easy to visualize and implement. Essentially, the computational domain is divided into an inner or core region, and an outer region. In the inner region, the full PIDE is solved. In the outer region, all terms involving the integral are set to zero and a parabolic PDE is solved. This will be justified on the basis of the properties of the Green's function of the PIDE [54]. As well, the integral term $\mathcal{H}$ is asymptotically zero in regions where the solution is asymptotically linear in (S_1, S_2); linearity in asset price is a common assumption for far-field boundary conditions in finance [108]. The outer region then acts as a buffer zone, so that the integral terms in the inner region have enough information for a sufficiently accurate evaluation.

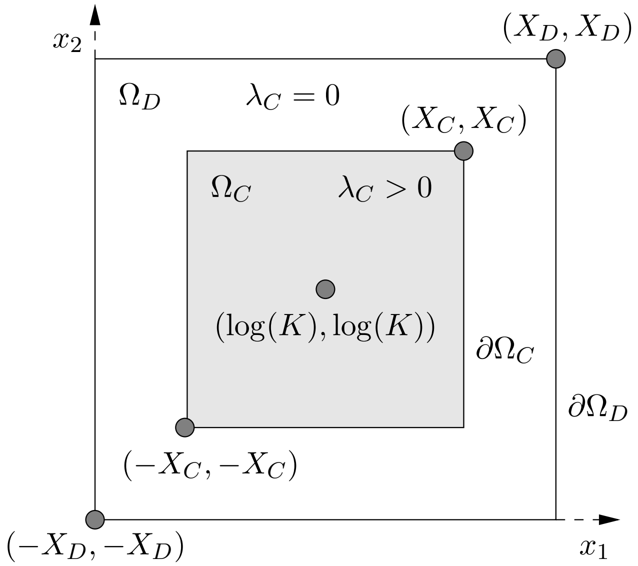

Figure 2.3.1: Domains $\Omega_C \subset \Omega_D$ truncate the infinite domain Ω_∞. In Ω_C coefficient $\lambda_C > 0$ and $\lambda_C = 0$ outside of Ω_C. Thus the PIDE 2.3.2 is computed with the jump component in Ω_C only.

2.3.1 Truncation to Nested, Finite Domains Ω_C and Ω_D

Over (x_1, x_2) define an interior domain nested in a finite domain $\Omega_C \subset \Omega_D \subset \Omega_\infty$, as shown in Figure 2.3.1, with upper and lower bounds

$$\begin{aligned} \Omega_C &= [-X_C, X_C] \times [-X_C, X_C] \\ \Omega_D &= [-X_D, X_D] \times [-X_D, X_D] \ ; \quad 0 < X_C < X_D \\ \partial\Omega_D &= \quad \{x_1 = [-X_D, X_D], x_2 = \{-X_D, X_D\}\} \\ &\quad \cup \{x_2 = [-X_D, X_D], x_1 = \{-X_D, X_D\}\} \ . \end{aligned} \tag{2.3.1}$$

In general, the upper and lower limits need not be equal nor the domains square. The approximate option value $V(x_1, x_2, \tau)$ on the localized domain $\Omega_D \times [0, T]$ with boundary conditions $B(x_1, x_2, \tau)$ is determined by solving

$$\begin{aligned} \frac{\partial V}{\partial \tau} &= \mathcal{L}V + \lambda_C \mathcal{H}_D V \\ V(x_1, x_2, 0) &= \mathcal{I}(x_1, x_2) \\ V(x_1, x_2, \tau) &= B(x_1, x_2, \tau) \ ; \quad (x_1, x_2) \in \partial\Omega_D \ . \end{aligned} \tag{2.3.2}$$

Operator $\mathcal{L}$ is defined by Equation 2.2.4 and

$$\lambda_C = \begin{cases} \lambda & \text{for} \ \ (x_1, x_2) \in \Omega_C \\ 0 & \text{for} \ \ (x_1, x_2) \in \Omega_D \setminus \Omega_C \ \ \text{(outside of } \Omega_C). \end{cases}$$

The domain of integration in Equation 2.2.5 is truncated, and Equations 2.2.6, 2.2.7 and 2.2.8 are applied to obtain

$$\begin{aligned}
\mathcal{H}_D V &= \iint_{(x_1+J_1, x_2+J_2)\in\Omega_D} g(J_1,J_2) V\left(x_1+J_1, x_2+J_2, \tau\right)\, dJ_1\, dJ_2 \\
&\quad - \iint_{-\infty}^{\infty} g(J_1,J_2) \left[V(x_1,x_2,\tau) + \left(e^{J_1}-1\right)\frac{\partial V}{\partial x_1} + \left(e^{J_2}-1\right)\frac{\partial V}{\partial x_2}\right] dJ_1\, dJ_2 \\
&= \iint_{(J_1,J_2)\in\Omega_D} g(J_1-x_1, J_2-x_2) V\left(J_1, J_2, \tau\right)\, dJ_1\, dJ_2 \\
&\quad - V(x_1,x_2,\tau) - \kappa_1 \frac{\partial V}{\partial x_1} - \kappa_2 \frac{\partial V}{\partial x_2} \\
&= \mathcal{J}_D V - V(x_1,x_2,\tau) - \kappa_1 \frac{\partial V}{\partial x_1} - \kappa_2 \frac{\partial V}{\partial x_2}
\end{aligned} \tag{2.3.3}$$

.

The first term of $\mathcal{H}_D V$ is written in two ways, corresponding to the two forms of $\mathcal{J}U$ in Equation 2.2.9, and denoted $\mathcal{J}_D V$. In the first form, the integration for a point $(x_1, x_2) \in \Omega_C$ must be performed over $(x_1 + J_1, x_2 + J_2) \in \Omega_D$; the value of V is not defined outside of Ω_D so the integration limit for J depends on (x_1, x_2). The second form of $\mathcal{J}_D V$ is equivalent, but since $g(J_1 - x_1, J_2 - x_2)$ is defined over $(J_1 - x_1, J_2 - x_2) \in \Omega_\infty$, the integration limit is simpler and independent of the point (x_1, x_2) being computed. The second form also illustrates one reason why λ_C is set so that $\lambda_C \mathcal{H}_D V(x_1, x_2) = 0 \;\forall\; (x_1, x_2) \notin \Omega_C$. For $(x_1, x_2) \in \Omega_C$ the range of evaluation of $g(J_1 - x_1, J_2 - x_2)$ with $(J_1, J_2) \in \Omega_D$ is not severely truncated in any given direction compared to the infinite integral $(J_1, J_2) \in \Omega_\infty$ used for the second and third terms. If size of Ω_C and Ω_D is set so that this truncation occurs when the value of the PDF $g(J_1 - x_1, J_2 - x_2)$ is small, then the finite evaluation range is expected to have a small impact on the solution over Ω_C.

For a complete review of the possible boundary condition assumptions and their implications see [108]. The upper boundary may be approximated by a time-dependent value

$$V(x_1,x_2,\tau) = a_0 e^{-r\tau} + a_1 e^{x_1} + a_2 e^{x_2} \quad \text{on} \begin{cases} x_1 = X_D & x_2 = [0, X_D] \\ x_2 = X_D & x_1 = [0, X_D] \end{cases} \tag{2.3.4}$$

where a_0, a_1, a_2 are determined by the option contract and Equation 2.3.4 is enforced as a Dirichlet boundary condition. On the lower boundaries,

$$\frac{\partial V}{\partial \tau} = \begin{cases} \left(r - \dfrac{{\sigma_2}^2}{2}\right)\dfrac{\partial V}{\partial x_2} + \dfrac{{\sigma_2}^2}{2}\dfrac{\partial^2 V}{\partial {x_2}^2} - rV & \text{on } x_1 = -X_D,\ x_2 \in (-X_D, X_D) \\ \left(r - \dfrac{{\sigma_1}^2}{2}\right)\dfrac{\partial V}{\partial x_1} + \dfrac{{\sigma_1}^2}{2}\dfrac{\partial^2 V}{\partial {x_1}^2} - rV & \text{on } x_2 = -X_D,\ x_1 \in (-X_D, X_D) \text{ and} \\ -rV & \text{at } (x_1, x_2) = (-X_D, -X_D)\ . \end{cases} \tag{2.3.5}$$

This follows from the differential terms at the lower boundaries in price-scaling (Equation 2.2.13), matching those on the lower boundaries $S_1 = 0$ or $S_2 = 0$. This also assumes that, in price scaling, the derivatives normal to the boundary of the option value are bounded. Note that these lower boundary conditions do not rely on interior solution values, so may be treated as time-dependent, Dirichlet boundary conditions when it is convenient to do so.

The lower boundary is in domain Ω_D, thus $\lambda_C = 0$ which eliminates the integral terms. Boundary $\partial\Omega_D$ must be spaced sufficiently far from Ω_C that the error from boundary approximations are well controlled (see Sections 2.3.3 and 2.7.2).

2.3.2 Convergence Estimates for European Options

This choice of localization to finite domains and the accompanying approximations is convenient for several reasons. First, if $X_D > X_C$ are both sufficiently large then one expects that the error due to approximating $\mathcal{H}$ by $\mathcal{H}_D$ and λ by λ_C will be negligible in Ω_C, particularly near the strike. Secondly, this localization allows an efficient, FFT-based computation of the integral term $\mathcal{H}_D V$. Finally, the numerical computation of the PDE in $\Omega_D \setminus \Omega_C$ is inexpensive and assumed to be a better approximation than, for example, simply using the initial condition $\mathcal{I}(x_1, x_2)$ in this region.

The following assumptions are required in order to bound the error due to this localization.

Assumption 2.3.1 *The initial condition (the option payoff) $\mathcal{I}(x_1, x_2)$ can be bounded by*

$$\mathcal{I}(x_1, x_2) \leq c_1 + c_2\left(e^{x_1} + e^{x_2}\right)$$

for some constants c_1 and c_2 and the jump distribution $g(J_1, J_2)$ must be such that $|\mathcal{H}\mathcal{I}| < \infty$ for $|x_1, x_2| < \infty$.

A payoff which is linearly valued in price scaling, such as those listed in Section 2.2.3, satisfies Assumption 2.3.1 for the Normal and exponentially distributed jumps used in the numerical examples.

Assumption 2.3.2 *The infinite domain solution $U(x_1, x_2, \tau) \in \Omega_\infty$ to Equation 2.2.3 satisfies the condition*

$$|\mathcal{H}U| \leq c_3 + c_4\left(e^{x_1} + e^{x_2}\right) \quad \textit{for} \quad x \in \Omega_\infty \setminus \Omega_C \tag{2.3.6}$$

for constants c_3 and c_4.

Besides being finite, the action of the jump operator on the option value must be bounded by a plane in price scaling. Again, this will be satisfied by the numerical examples, where the jump density functions are independent of (x_1, x_2) and the value $U(x_1, x_2, \tau)$ also satisfies a bound of the form given in Assumption 2.3.1. This limitation could apply, for example, to jump density functions which increase the jump magnitude with the logarithm of the asset price.

Assumption 2.3.3 *The artificial boundary condition $B(x_1, x_2, \tau)$ of Equation 2.3.2 is bounded by the growth in the exact, infinite domain solution $U(x_1, x_2, \tau) \in \Omega_\infty$, i.e. by*

$$|B(x_1, x_2, \tau) - U(x_1, x_2, \tau)| \leq c_5 + c_6\, U(x_1, x_2, \tau) \tag{2.3.7}$$

for some constants c_5 and c_6.

Note that Equation 2.3.7 is satisfied even if $B(x_1, x_2, \tau) = 0$.

Assumption 2.3.4 *The PIDE's 2.2.3 and 2.3.2 must satisfy the conditions in Garroni and Menaldi [54] §I, II. In particular, the diffusion coefficients must be bounded on Ω_∞ and the operator $\mathcal{L}$ must be uniformly elliptic, so that a smooth, classical, bounded solution exists. With these conditions, the solution can be represented by convolutions of Green's functions and Poisson functions, as in [54] §IV.*

Note that the PIDE for the option value formulated in price scaling over the infinite domain does not satisfy Assumption 2.3.4: the differential operator does not have bounded coefficients on the infinite domain and is not uniformly elliptic [54] because of the zero diffusion coefficients in Equation 2.2.13 on $S_1 = 0$ and $S_2 = 0$. Hence this thesis uses log-price scaling: the domain is bounded away from the $S_1 = 0$ and $S_2 = 0$ axes and the diffusion coefficients are constant.

Assumption 2.3.5 *The initial and boundary conditions are smooth and have finite first and second derivatives with respect to x_1 and x_2 (see [54] §II.1.1).*

The initial conditions in Equation 2.2.14 do not meet Assumption 2.3.5, however, an arbitrarily close, but smooth approximation to $\mathcal{I}(x_1, x_2)$ may be made to satisfy the theory (independent of the numerical solution approach). Typically, this regularization is done using a mollification of the initial condition, with which the resulting error in the final solution may also be bounded to an arbitrarily small value. See [51, 52] for the classical mollification method for PDE's and a survey by Lamm [75] for its application to integral equations. Recently this approach has been applied in practice to financial problems by Friz and Gatheral [53] and in theory by Jakobsen et al. [68] (particularly Lemma 3.1).

Remark 2.3.6 *Assumptions 2.3.1 to 2.3.5 are taken as fulfilled for the following theorems and the remaining discussion of the localization of the continuous operators.*

The error due to the solution V of Equation 2.3.2 over $(x_1, x_2) \in \Omega_D$ must now be shown to satisfy $|V(x_1, x_2, \tau) - U(x_1, x_2, \tau)| \to 0$ as $\Omega_D, \Omega_C \to \Omega_\infty$ where U is the solution to Equation 2.2.3 over Ω_∞. This is done in two parts.

Theorem 2.3.7 *Let U be the solution to PIDE 2.2.3. Let V be the solution of an initial-value PIDE*

$$\begin{gathered} \frac{\partial V}{\partial \tau} = \mathcal{L}V + \lambda_C \mathcal{H}_D V \\ V(x_1, x_2, 0) = \mathcal{I}(x_1, x_2)\ ; \quad (x_1, x_2) \in \Omega_\infty \end{gathered} \tag{2.3.8}$$

where, for this equation

$$\lambda_C = \begin{cases} \lambda & \textit{for} \;\; (x_1, x_2) \in \Omega_C \\ 0 & \textit{for} \;\; (x_1, x_2) \in \Omega_\infty \setminus \Omega_C \;\; \textit{(outside of } \Omega_C\textit{)} \end{cases}$$

and $\mathcal{H}_D V$ is defined in Equation 2.3.3. This is similar to localized PIDE 2.3.2, but embedded in Ω_∞ and without a boundary condition.

*Define the **cutoff error** $E_c = U - V$. The value of $E_c(x_1, x_2, \tau)$ for a fixed $(x_1, x_2) \in \Omega_\infty$ due to the approximation of λ by λ_C and $\mathcal{H}$ by $\mathcal{H}_D$ obeys*

$$\lim_{\Omega_C, \Omega_D \to \Omega_\infty} |E_c(x_1, x_2, \tau)| = 0 \; . \tag{2.3.9}$$

Proof. See Appendix C.1. □

Theorem 2.3.8 *Let Y be the solution to Equation 2.3.2 with the approximate boundary condition $Y(x_1, x_2, \tau) = B(x_1, x_2, \tau), (x_1, x_2) \in \partial\Omega_D$. Let W be the solution to Equation 2.3.2 when the boundary is set to $W(x_1, x_2, \tau) = V(x_1, x_2, \tau), (x_1, x_2) \in \partial\Omega_D$, where V is the exact value from the solution of Equation 2.3.8. Define the error due to approximating the exact boundary condition $V(x_1, x_2, \tau)$ with the approximate boundary condition $B(x_1, x_2, \tau)$ on $\partial\Omega_D$ as $E_b = W - Y$.*

The error $E_b(x_1, x_2, \tau)$ has the limit

$$\lim_{\Omega_D \to \Omega_\infty} |E_b(x_1, x_2, \tau)| = 0 \; . \tag{2.3.10}$$

Proof. See Appendix C.2. □

Approximating Equation 2.2.3 by Equation 2.3.2 causes errors which tend to zero as $\Omega_C, \Omega_D \to \Omega_\infty$. These limits appear at first to be disappointingly weak, however, actual bounds would depend on the exact form of the jump size distribution $g(J_1, J_2)$. Using a different localization technique, the limit for this error in one dimension were estimated in [34] using a probabilistic approach. Defining the computational domain in price scaling by $[0, S^*]$, the localization error was estimated to be [34]

$$\text{LocalizationError} \leq \frac{1}{(S^*)^\alpha}, \qquad \alpha > 0 \tag{2.3.11}$$

which is similar to the limit in Equations 2.3.9 and 2.3.10.

The above estimates of the localization error are overly pessimistic. To see this, note that in many cases large regions of the payoff and the solution are asymptotically linear in price scaling as $S_1, S_2 \to \infty$. Consider such a region $\Omega_P \subset \Omega_D$ where, in log-price scaling,

$$\begin{aligned} &V(x_1, x_2) = c_0 + c_1 \, e^{x_1} + c_2 \, e^{x_2} \\ &(x_1, x_2) \in \Omega_P \; . \end{aligned}$$

Note that for $(x_1, x_2) \in \Omega_P$

$$\frac{\partial V}{\partial x_1} = c_1 e^{x_1} \ , \quad \frac{\partial V}{\partial x_2} = c_2 e^{x_2} \ .$$

Examine the integral term $\mathcal{H}V$ of Equation 2.2.5. By limiting the integral so that it is taken only over $(x_1, x_2), (x_1 + J_1, x_2 + J_2) \in \Omega_P$ then

$$\begin{aligned}
\mathcal{H}V(x_1, x_2) &\simeq \mathcal{H}_P V(x_1, x_2) \\
&= \iint_{(x_1,x_2),(x_1+J_1,x_2+J_2)\in\Omega_P} g(J_1, J_2) \cdot \\
&\qquad \Bigg[V(x_1 + J_1, x_2 + J_2) \\
&\qquad\quad - V(x_1, x_2, \tau) - \left(e^{J_1} - 1\right) \frac{\partial V}{\partial x_1} - \left(e^{J_2} - 1\right) \frac{\partial V}{\partial x_2} \Bigg] \, dJ_1 \, dJ_2 \\
&= \iint_{(x_1,x_2),(x_1+J_1,x_2+J_2)\in\Omega_P} g(J_1, J_2) \cdot \\
&\qquad \Bigg[c_0 + c_1 \, e^{(x_1+J_1)} + c_2 \, e^{(x_2+J_2)} \\
&\qquad\quad - (c_0 + c_1 \, e^{x_1} + c_2 \, e^{x_2}) \\
&\qquad\quad - \left(c_1 e^{(x_1+J_1)} - c_1 e^{x_1} \right) - \left(c_2 e^{(x_2+J_2)} - c_2 e^{x_2} \right) \Bigg] \, dJ_1 \, dJ_2 \\
&= 0 \ .
\end{aligned}$$

In such regions one may expect that the error due to setting $\lambda_C = 0$ and dropping the integral term, or due to limiting the region of integration of $\mathcal{H}_D$, will be small.

2.3.3 Sizing Ω_D and Ω_C for Error Control

If Ω_D is sufficiently large then the solution in significant regions outside of Ω_C tends to approach the limiting assumptions $\mathcal{H}V \to 0$ discussed in Section 2.3.2. The error due to setting $\lambda_C = 0$, or due the bounded region of integration of $\mathcal{J}_D V$, tends to be small when Ω_D and Ω_C are chosen appropriately.

The sizes of Ω_C and Ω_D can be dictated by the asset price evolution process in a way which controls numerical error. It is important for Ω_C to be sufficiently large to ensure the solution near (x_1, x_2) is minimally affected by error in the $\Omega_D \setminus \Omega_C$ region. PDF $g(J_1, J_2)$ is truncated when the PIDE is localized; a sufficiently large domain Ω_D will limit the integral error due to this truncation. The following rules of thumb for sizing are adapted from [38] for the x_1 direction; the x_2 direction is formulated identically.

Let $(x_1, x_2)_p$ be a point within a region of particular interest, typically a small region around the $(\log(K), \log(K))$ point. It is straightforward to compute or approximate, usually by Fourier transform inversion [41, 78], the marginal distribution $f_1(x_1; (x_1, x_2)_p, T)$ of the expected value of an asset price path starting from $(x_1, x_2)_p$ and evolving to option expiry time T. Width Δx_C

is computed based on a tolerance parameter ϵ_c such that

$$\int_{-\Delta x_C}^{\Delta x_C} f_1(x_1 + z; (x_1, x_2)_p, T)\, dz > (1 - \epsilon_c) \tag{2.3.12}$$

and the size of Ω_C is chosen to ensure $(x_1 \pm \Delta x_C, x_2) \in \Omega_C \ \forall \ (x_1, x_2)_p$. The reasoning behind this rule is to set Ω_C sufficiently large that the influence of solution details outside Ω_C have an influence approximately proportional to ϵ_c at expiry time for points near the strike.

Let $\Delta x_D = X_D - X_C$, the distance from a point on the upper Ω_C boundary $(x_1, x_2) \in \partial\Omega_C$ to the upper $\partial\Omega_D$ boundary in the positive x_1 direction. Distance Δx_D is chosen based on the marginal jump distribution $g_1(x_1)$ of $g(x_1, x_2)$ in the x_1 direction and the initial value $\mathcal{I}$ such that a given integral evaluation error tolerance ϵ_d satisfies (where Assumption 2.3.1 holds)

$$\begin{aligned} &\max_{(x_1,x_2)\in\partial\Omega_C} \left[\int_{\Delta x_D}^{\infty} g_1(z)\, \mathcal{I}(x_1 + z, x_2)\, dz \right] \\ &\qquad \simeq \max_{(x_1,x_2)\in\partial\Omega_D} \left[\mathcal{I}(x_1, x_2) \right] \int_0^{\infty} g_1((\Delta x_D + z))\, e^{\alpha z}\, dz \\ &\qquad \simeq \max_{(x_1,x_2)\in\partial\Omega_D} \left[\mathcal{I}(x_1, x_2) \right]\, \epsilon_d \ . \end{aligned} \tag{2.3.13}$$

Constant α is the slope of the initial condition at the upper boundary in the positive x_1 direction. Typically $\alpha = 0$ in the case of a put and $\alpha = 1$ for a call. The lower bound spacing along x_1 is computed analogously, with $\alpha = 0$. The use of $\mathcal{I}$ is as an approximation to V for the purposes of computing the domain size. The selection of $\max_{(x_1,x_2)\in\partial\Omega_D} [\mathcal{I}(x_1, x_2)]$ as a multiplying factor ensures that the error control accounts for an artifact of the numerical solution approach, explained in Section 2.4.3.3, which computes the integral using a method that treats Ω_D as periodic and may cause the maximum value of V to factor into the error near the boundary. In the case of one-sided or one-dimensional jump distributions, Δx_D is never allowed to go to zero so that, in particular, the lower boundary condition Equations 2.3.5 never need to be altered to allow for an integral term.

2.4 Discretization

Recall Equation 2.3.2, the localized problem posed on the finite domain $(x_1, x_2) \in \Omega_D$. To simplify the explanation of the discretization, a linear differential operator $\mathcal{G}$ is used which contains only partial differential terms. Separating out the three terms of $\mathcal{H}$, rewrite Equation

2.3.2 over $\Omega_D \times [0,T]$ as

$$\begin{aligned}
\frac{\partial V}{\partial \tau} &= \mathcal{G}V - (r+\lambda_C)V + \lambda_C \mathcal{J}_D V \\
V(x_1,x_2,0) &= \mathcal{I}(x_1,x_2) \\
V(x_1,x_2,\tau) &= B(x_1,x_2,\tau) \quad ; \quad (x_1,x_2) \in \partial\Omega_D \\
\mathcal{G}V = &\quad \left(r - \frac{{\sigma_1}^2}{2} - \lambda_C\, \kappa_1\right) \frac{\partial V}{\partial x_1} \\
&+ \left(r - \frac{{\sigma_2}^2}{2} - \lambda_C\, \kappa_2\right) \frac{\partial V}{\partial x_2} \\
&+ \frac{{\sigma_1}^2}{2}\frac{\partial^2 V}{\partial {x_1}^2} + \rho\sigma_1\sigma_2 \frac{\partial^2 V}{\partial x_1 \partial x_2} + \frac{{\sigma_2}^2}{2}\frac{\partial^2 V}{\partial {x_2}^2} \\
\mathcal{J}_D V &= \int_{(J_1,J_2)\in\Omega_D} g(J_1-x_1, J_2-x_2) V(J_1,J_2,\tau)\, dJ_1\, dJ_2
\end{aligned} \tag{2.4.1}$$

where the boundary conditions $B(x_1,x_2,\tau)$ are unchanged. Note that $\mathcal{J}_D$ is chosen to match the second form of $\mathcal{H}_D$ in Equation 2.3.3.

Equation 2.4.1 can now be semi-discretized in time by the theta method with constant time step weight $0 \le \theta \le 1$

$$\begin{aligned}
\frac{W^{n+1} - W^n}{\Delta\tau} &= (1-\theta)\left[\mathcal{G} - (r+\lambda_C) + \lambda_C \mathcal{J}_D\right] W^{n+1} \\
&\quad + \theta\left[\mathcal{G} - (r+\lambda_C) + \lambda_C \mathcal{J}_D\right] W^n
\end{aligned} \tag{2.4.2}$$

where $W^n = W(x_1,x_2,\tau_n)$ is the solution to the semi-discretized problem. This form will be used later in Section 2.6.1. Only the cases $\theta = 1/2$ and $\theta = 0$ are considered, which are the second order Crank–Nicolson time step and the first order fully implicit time step respectively.

2.4.1 General Discrete Form

The discrete equations are first written in a general form using matrices and vectors, with the details for the differential and integral terms to follow in Sections 2.4.2 and 2.4.3. The general form permits the application of some useful general stability results.

Discretization is performed over a finite difference (FD) grid of N points in space and N_τ points in time

$$(x_1,x_2)_\mathsf{i} \in [-X_D, X_D) \times [-X_D, X_D) \ , \qquad \mathsf{i} = 1 \dots N \tag{2.4.3}$$

$$\tau_n \in [0,T] \ , \qquad n = 0 \dots N_\tau \ , \quad \tau_n = n\Delta\tau \tag{2.4.4}$$

and form a vector of solution values at these points[1]

$$\mathsf{w} \in \mathbb{R}^N \ , \quad \text{with elements } \mathsf{w}_\mathsf{i} \simeq W((x_1,x_2)_\mathsf{i}) \ .$$

[1] The compact notation $W((x_1,x_2)_\mathsf{i}, \tau_n)$, or $W((x_1,x_2)_\mathsf{i})$ where the τ_n is either implied or not relevant, will be used instead of $W(x_{(1,\mathsf{i})}, x_{(2,\mathsf{i})})$ or $W(x_{(1,\mathsf{i})}, x_{(2,\mathsf{i})}, \tau_n)$ to avoid double subscripting. Also, $f(y_{(1,\mathsf{i})} + z_{(1,\mathsf{j})}, y_{(2,\mathsf{i})} + z_{(2,\mathsf{j})})$ will be denoted $f((y_1,y_2)_\mathsf{i} + (z_1,z_2)_\mathsf{j})$ where a sum of two co-ordinates is required.

This requires a boundary condition enforcement vector

$$\mathsf{b} \in \mathbb{R}^N \ , \quad \text{with elements } \mathsf{b}_\mathsf{i} \simeq b((x_1, x_2)_\mathsf{i}) \ .$$

Vector b can be seen as encoding the Dirichlet boundary condition nodes on $\partial\Omega_D$, where the option value is known at time steps n and $n+1$, after these nodes are eliminated from the solution vector (see Appendix E) and hence from the discrete equations. Note that vector b is not a direct representation of the values of $B(x_1, x_2, \tau)$. The linear differential operator is discretized to form a matrix G such that

$$\mathcal{G}W \simeq \mathsf{Gw} \ ; \quad \mathsf{G} \in \mathbb{R}^{N \times N}, \tag{2.4.5}$$

and the integral operator discretized to form a matrix J such that

$$\lambda_C \mathcal{J}_D W \simeq \lambda \mathsf{I_c J w} \ ; \quad \mathsf{I_c}, \mathsf{J} \in \mathbb{R}^{N \times N} \tag{2.4.6}$$

$$[\mathsf{I_c}]_{\mathsf{ij}} = \begin{cases} 1 & \text{if } \mathsf{i} = \mathsf{j} \text{ and } (x_1, x_2)_\mathsf{i} \in \Omega_C \\ 0 & \text{otherwise.} \end{cases} \tag{2.4.7}$$

Using $\lambda \mathsf{I_c}$ to replace λ_C, the discrete form of the spatial operator terms of Equation 2.4.2 can be written using a matrix

$$\mathsf{T} = -\left[\mathsf{G} + \lambda \mathsf{I_c}(\mathsf{J} - \mathsf{I}) - r\,\mathsf{I}\right] \tag{2.4.8}$$

where I is the identity matrix.

The discrete version of the time step Equation 2.4.2 can now be written to match the formulation of [73, 77, 102]. The full, general, discrete system is thus

$$\left[\mathsf{I} + (1-\theta)\Delta\tau\ \mathsf{T}\right] \mathsf{w}^{n+1} = \left[\mathsf{I} - \theta\Delta\tau\ \mathsf{T}\right] \mathsf{w}^n + \mathsf{b} \tag{2.4.9}$$

which is the form required to apply some of the stability analyses of [73, 77] in Section 2.5 below. To use these analyses, the theta method time discretization is written in the form

$$\begin{aligned} \mathsf{w}^{n+1} &= \varphi(z)\mathsf{w}^n + \left[\mathsf{I} + (1-\theta)\Delta\tau\ \mathsf{T}\right]^{-1} \mathsf{b} \\ z &= \Delta\tau\ \mathsf{T} \end{aligned}$$

that uses a rational polynomial defined as in [73][2] to be

$$\varphi(z) = \left[1 + (1-\theta)z\right]^{-1}\ \left[1 - \theta z\right] \ . \tag{2.4.10}$$

Equation 2.4.9 can also be written in a compact form which will be used in some definitions in Section 2.5 as

$$\begin{gathered} \mathsf{M}_\mathsf{i}^{n+1}(h, \mathsf{w}_\mathsf{i}^{n+1}, \mathsf{w}_\mathsf{j}^{n+1}, \mathsf{w}_\mathsf{i}^{n}, \mathsf{w}_\mathsf{j}^{n}) = \\ \left[\mathsf{I} + (1-\theta)\Delta\tau\ \mathsf{T}\right] \mathsf{w}^{n+1} - \left[\mathsf{I} - \theta\Delta\tau\ \mathsf{T}\right] \mathsf{w}^n - \mathsf{b} = 0 \ ; \quad \mathsf{i}, \mathsf{j} \in 1 \ldots N \ , \quad \mathsf{i} \neq \mathsf{j} \end{gathered} \tag{2.4.11}$$

where grid spacing is controlled by h and $\Delta x_1, \Delta x_2, \Delta\tau = \mathrm{O}(h)$ are representative grid spacings in space and time.

[2] In [73] the values of θ and $(1-\theta)$ are reversed to the sense in which they are used here.

2.4.2 Finite Difference Form of G, the Partial Differential Terms

The finite difference grid (**FD grid**) is rectangular and defined on domain Ω_D. The grid line intersections define the N solution points $(x_1, x_2)_\mathsf{i}$ in log-price scaling. The grid has a fine spacing between points near the location $(x_1, x_2) = (\log(K), \log(K))$ and, to save computational effort, grid spacing increases in regions away from the strike where the high resolution is not required. Previous work [112] has shown that the original grid should be specified so as to accurately capture the details of the option contract, in particular, the payoff and any barriers need to be accurately represented.

The sparse matrix G is formed over the FD grid using second order finite differences for the diffusion term. A second order, central finite difference is used for the drift terms where possible (see Section 2.5 below). Appendix E gives the details of the discretization assuming constant grid spacing. In the interests of brevity, the details for non-constant spacing are omitted, since this is completely standard. The cross-partial derivatives are discretized with a seven-point formula, using the non-constant spacing versions of Equation E.1.5 or E.1.6 when the diffusion correlation $\rho > 0$ or $\rho < 0$ respectively. Under certain conditions, detailed in Remark 2.5.13 below, the first-order approximation to the first partial derivatives is required.

The finite difference approximation is $\mathrm{O}(h^2)$ for constant grid spacing h when central differencing is used for the first order terms. This matches the $\mathrm{O}((\Delta\tau)^2)$ obtained when $\theta = 1/2$ in Equation 2.4.9. For non-constant spacing, if changes in the grid spacing are $\mathrm{O}(h)$, the second order convergence is preserved.

2.4.3 Discrete Form of J, the Correlation Integral

In this context, it is not necessary to achieve a high accuracy evaluation in the jump integral term; second order accuracy at each time step is good enough. This section considers the motivation for the method which evaluates the jump integral term, then proceeds with the details of the numeric quadrature.

Consider a simple form of J created when a second order, trapezoidal rule is used for the approximation in Equation 2.4.6 of $\mathcal{J}_D$. This may be written in the following form over the FD grid to form a dense matrix J

$$[\mathsf{I_c J w}]_\mathsf{i} = \begin{cases} [\mathsf{Jw}]_\mathsf{i} = \sum_{\mathsf{j}=1}^{N} f'_g((x_1,x_2)_\mathsf{j} - (x_1,x_2)_\mathsf{i})\,\mathsf{w_j} & \text{for } (x_1,x_2)_\mathsf{i} \in \Omega_C \text{ and} \\ \qquad\qquad\qquad\qquad\qquad\qquad 0 & \text{for } (x_1,x_2)_\mathsf{i} \notin \Omega_C\ . \end{cases} \tag{2.4.12}$$

Function $f'_g((x_1, x_2)_\mathsf{j} - (x_1, x_2)_\mathsf{i})$ is formed by integrating $g((z_1, z_2) - (x_1, x_2)_\mathsf{i})$ over (z_1, z_2) inside the cell centered on node $(x_1, x_2)_\mathsf{j}$, as will be explained in detail in Section 2.4.3.1 below. The result is second order accurate over the grid of N points and J naturally satisfies the conditions

$$\mathsf{J_{ij}} \geq 0\ , \quad \text{and} \quad \sum_\mathsf{j} \mathsf{J_{ij}} \leq 1\ . \tag{2.4.13}$$

Note that option value W is only given at points w_j, located at nodes $(x_1, x_2)_\mathsf{j}$ on the grid. On a general FD grid the set of points $(x_1, x_2)_\mathsf{j} - (x_1, x_2)_\mathsf{i}$, around which g is integrated, may differ for each $(x_1, x_2)_\mathsf{i}$.

Using Equation 2.4.12 would mean that Equation 2.4.9, although useful for theoretical analysis, could not be used for a practical algorithm. Matrix J (and hence T) is dense and solving Equation 2.4.9 could require the solution to a dense linear system and a dense matrix multiply at each time step. The dense linear system solution can be avoided by using the iterative method described in Section 2.6, however, this still leaves the dense matrix-vector multiply.

There are fast methods which, under certain conditions, can be used to carry out Equation 2.4.12 (see [38]). If the sum is performed over a rectangular grid with constant spacing, then Equation 2.4.12 can be performed by exploiting the algebraic identity which uses the discrete Fourier transform (DFT) and in turn, the fast Fourier transform ([17] §13). To exploit this approach, a two dimensional version of the method used in [38] is created. This method requires an interpolation of the original FD grid of values onto a DFT grid, a summation corresponding to Equation 2.4.12 by FFT, then an interpolation of the result back to the FD grid. Since all stages are second order accurate the approach satisfies the accuracy requirement.

There are several algorithms which can be used to determine the FFT of input data on unequally spaced grids [42, 96, 106]. These eliminate the need for interpolation between grids. However, some previous tests [36] indicate that these approaches were no more efficient than the simple interpolation strategy used here. Previously, experiments were conducted with a Fast Gauss Transform [22] for the case when g is a Normal distribution, which also does not require a regular grid. However, this method did not appear to be any more efficient than FFT–based methods, at least for the order of accuracy required here.

2.4.3.1 DFT Domain and Grid

To apply the discrete Fourier transform to Equation 2.4.12, and hence allow the use of an FFT, the sum must be computed over a rectangular grid with constant grid line spacing. A rectangular domain Ω_D^* is defined with dimensions such that $\Omega_C, \Omega_D \subset \Omega_D^*$. The **DFT grid** over Ω_D^* is defined with $N_F = N_1 \times N_2$ nodes at the grid line intersections and it covers Ω_D^* with identically sized cells centered on those nodes. Nodes are denoted

$$(x_1, x_2)_\mathsf{k} \in \Omega_D^* \ , \quad \mathsf{k} = 1 \ldots N_F \ . \tag{2.4.14}$$

The integers N_1 and N_2 are chosen so that, for a DFT grid spacing (h_1, h_2) which is of the order of the size of the FD grid spacing at the strike node, the DFT grid overlaps the FD grid. In general $N_1 \neq N_2$ and $h_1 \neq h_2$.

Define the vector of solution values

$$\mathsf{x} \in \mathbb{R}^{N_F} \ , \quad \text{with elements } \mathsf{x}_\mathsf{k} \simeq W((x_1, x_2)_\mathsf{k})$$

at nodes of the DFT grid. The matrix form of the integral operator of Equation 2.4.6 over the DFT grid is

$$\mathcal{J}_D W \simeq \mathsf{J}_\mathsf{r} \mathsf{x} \ ; \quad \mathsf{J}_\mathsf{r} \in \mathbb{R}^{N_F \times N_F} \ . \tag{2.4.15}$$

As in Equation 2.4.12, the correlation is integrated over the DFT grid with a second order, cell-centered quadrature rule. The first line of Equation 2.4.12 then becomes, for a point $(x_1, x_2)_\mathsf{k} \in \Omega_C$ on the DFT grid,

$$[\mathsf{J_r x}]_\mathsf{k} = \sum_{\mathsf{l}=1}^{N_F} f_g((x_1,x_2)_\mathsf{l} - (x_1,x_2)_\mathsf{k})\, \mathsf{x_l} \tag{2.4.16}$$

where f_g is given by

$$f_g((x_1,x_2)_\mathsf{l} - (x_1,x_2)_\mathsf{k}) = \int_{-h_1/2}^{+h_1/2} \int_{-h_2/2}^{+h_2/2} g((x_1,x_2)_\mathsf{l} - (x_1,x_2)_\mathsf{k} + (z_1,z_2))\, dz_1\, dz_2 \ . \tag{2.4.17}$$

This ensures that the conditions of Equation 2.4.13 are satisfied[3]. The actual numerical computation uses a periodic form of f_g, explained below in Section 2.4.3.3.

The DFT grid must be sufficiently fine that the solution and the PDF are adequately represented by their discrete form.[4] An inaccurate representation of, in particular, the PDF can produce useless results. This happens, typically, when the spatial grids are too coarse (examples appear later in Section 3.5.1). This problem is usually addressed by refining the DFT grid, as well as by the techniques in Sections 2.4.3.4 and 3.4.2 below.

2.4.3.2 Interpolating Between the FD and DFT Grids

A mapping is formed between a price vector $\mathsf{w} \in \mathbb{R}^N$ over the nodes of the FD grid and the price vector $\mathsf{x} \in \mathbb{R}^{N_F}$ on the nodes of the DFT grid. The mapping can be written as a $N_F \times N$ sparse matrix L so that $\mathsf{x} = \mathsf{Lw}$. The entries of L interpolate using a local, bi-linear, Lagrange interpolation over the FD grid. The interpolation stencil is chosen at each node so that

$$0 \le \mathsf{L_{ij}} \le 1 \quad \text{and} \quad \sum_\mathsf{j} \mathsf{L_{ij}} \le 1 \ .$$

Where DFT grid point $(x_1,x_2)_\mathsf{i} \in \Omega_D^*$ but $(x_1,x_2)_\mathsf{i} \notin \Omega_D$ then $\mathsf{L_{ij}} = 0\ \forall \mathsf{j}$ to set $\mathsf{x_i} = 0$ (rather than extrapolate). A bi-linear interpolation is applied in the other direction using $N \times N_F$ matrix K to compute $\mathsf{w} = \mathsf{Kx}$.

To apply the discrete integral term over the FD grid, approximate Equation 2.4.12 by

$$\mathsf{I_c J w} \simeq \mathsf{I_c}\,[\mathsf{K} \cdot \mathsf{J_r} \cdot \mathsf{L}]\ \mathsf{w} \tag{2.4.18}$$

where $\mathsf{J_r}$ of Equation 2.4.15 computes over the DFT grid. If $h_1, h_2 = \mathrm{O}(h)$ is the grid spacing on the DFT grid, then Equation 2.4.18 is an $O(h^2)$ approximation to Equation 2.4.12, which is

[3] Where no analytic CDF is available and the PDF g is sufficiently smooth, Equation 2.4.17 can be computed using a standard, high-accuracy numerical technique. The values $f_g((x_1,x_2)_\mathsf{l} - (x_1,x_2)_\mathsf{k})$ repeat from one matrix row to the next, and need only be evaluated once during the option valuation process, so this does not incur an undue computational cost. Where the PDF is non-smooth, as with the Marshall-Olkin Bi-variate Exponential Distribution (see Section B.2) the integral must be done directly by evaluating the cumulative distribution.

[4] In signal processing terms, the error due to "aliasing" is low.

as good as the error due to the finite difference operators. Note that if $\mathsf{J_r}$ satisfies the conditions of Equation 2.4.13, then the construction of L and K preserves this result so

$$[\mathsf{K} \cdot \mathsf{J_r} \cdot \mathsf{L}]_{\mathsf{ij}} \geq 0 \quad \text{and} \quad \sum_{\mathsf{j}} [\mathsf{K} \cdot \mathsf{J_r} \cdot \mathsf{L}]_{\mathsf{ij}} \leq 1 \ .$$

2.4.3.3 Fast Fourier Computation of the Integral Term

The details of the reduction of the cell-centered quadrature rule of Equation 2.4.16 to an operation involving the DFT is described in detail in standard texts (e.g. [17] §13). However, to exploit this method requires a new approximation F which is a Toeplitz matrix formed on a periodic domain. In general, $\mathsf{F} \neq \mathsf{J_r}$ because, in effect, $f_g((x_1,x_2)_{\mathsf{k}})$ in Equation 2.4.16 is replaced with a periodic function

$$\begin{aligned} f_g^{\circ}((x_1,x_2)_{\mathsf{k}}) &= f_g((x_1,x_2)_{\mathsf{k}}) \quad \forall \, (x_1,x_2)_{\mathsf{k}} \in \left(-\frac{N_1}{2}h_1, -\frac{N_2}{2}h_2\right) \times \left(\frac{N_1-1}{2}h_1, \frac{N_2-1}{2}h_2\right) \\ f_g^{\circ}((x_1,x_2)_{\mathsf{k}}) &= f_g^{\circ}((x_1,x_2)_{\mathsf{k}} + (a\,N_1h_1, b\,N_2h_2)), \quad \forall \, a,b \text{ integers.} \end{aligned}$$

The DFT in the periodic grid setting is denoted $\mathcal{D}$ and its inverse as $\mathcal{D}^{-1}$ (see Appendix F). For grid points $(x_1,x_2)_{\mathsf{k}}$, $\mathsf{k} = 1 \ldots N_F$ the Fourier transform form of Equation 2.4.16 is given by the identity

$$[\mathsf{F}\,\mathsf{x}]_{\mathsf{k}} = \frac{1}{N_F} \left\{ \mathcal{D}^{-1} \left[\mathcal{D}(\mathsf{x}) \cdot \overline{\mathcal{D}(f_g^{\circ})} \right] \right\}_{(x_1,x_2)_{\mathsf{k}}} \tag{2.4.19}$$

where $\mathcal{D}(\mathsf{x})$ and the complex conjugate of $\mathcal{D}(f_g^{\circ})$ are multiplied at each node on the Fourier-space grid. By using an FFT to compute the DFT on the N_F nodes, an $\mathrm{O}({N_F}^2)$ dense matrix multiplication is reduced to an $\mathrm{O}(N_F \log(N_F))$ operation. The scaling factor $1/N_F$ is a side effect of the form chosen for the DFT (Equation F.1.2).

Any solution using F will (typically near $\partial\Omega_D^*$) have been contaminated by values where $f_g^{\circ}((x_1,x_2)_{\mathsf{k}} - (x_1,x_2)_{\mathsf{l}}) \neq f_g((x_1,x_2)_{\mathsf{k}} - (x_1,x_2)_{\mathsf{l}})$ because of the periodicity implied in the FFT operation. Fortunately, the grid nesting strategy already dictates the retention only of the more accurate values in the core domain $\Omega_C \subset \Omega_D^*$ and discards the rest of the computation. The final form of the approximation to Equation 2.4.12 is given by

$$\mathsf{I_c J w} \simeq \mathsf{I_c} \left(\mathsf{K} \cdot \mathsf{F} \cdot \mathsf{L}\right) \mathsf{w} \tag{2.4.20}$$

where F is computed by applying Equation 2.4.19.

The approximation in Equation 2.4.20 is used in the iterative method described in Section 2.6.1 to solve time step Equation 2.4.9. The “wrap-around” error in the computation of the integral using the DFT approximation of Equation 2.4.19 may cause the maximum of the option value W to factor into the error. The control of the wrap-around error is already accounted for in the domain sizing methods discussed in Section 2.3.3.

2.4.3.4 Grid Alignment of the PDF

Let $g'(x_1 - z_1, x_2 - z_2) = g(x_1, x_2)$ where (z_1, z_2) is an arbitrary shift of the jump PDF function. Then, by using a change of integration variable

$$\begin{aligned}\mathcal{J}U(x_1, x_2) &= \iint_{-\infty}^{\infty} g(J_1, J_2)\, U(x_1 + J_1, x_2 + J_2)\, dJ_1\, dJ_2 \\ &= \iint_{-\infty}^{\infty} g'(J_1 - z_1, J_2 - z_2)\, U(x_1 + J_1, x_2 + J_2)\, dJ_1\, dJ_2 \\ &= \iint_{-\infty}^{\infty} g'(J_1', J_2')\, U(x_1 + J_1' + z_1, x_2 + J_2' + z_1)\, dJ_1'\, dJ_2' \\ &= \mathcal{J}'U(x_1 + z_1, x_2 + z_2)\end{aligned} \tag{2.4.21}$$

where $\mathcal{J}'$ computes with g' as the PDF. This holds in the discrete periodic case as well, to within $\mathrm{O}(h^2)$ where $h_1, h_2 = \mathrm{O}(h)$ is the DFT grid spacing, provided the domain has been constructed so that (z_1, z_2) is small relative to the size of Ω_D^*.

This fact may be used to align discontinuities in a jump PDF to fall exactly between DFT grid nodes. A translation of the PDF can be corrected when the result for point (x_1, x_2) is interpolated back from the correlation by simply interpolating at $(x_1 + z_1, x_2 + z_2)$. The DFT-based procedure is equivalent to the cell-centered integration rule. If PDF discontinuities can be aligned to fall on cell edges (not nodes) then the integration captures the discontinuity location exactly. The result, although still accurate to $\mathrm{O}(h^2)$, tends to be much more accurate in absolute value, significantly improving the results when h specifies a coarse grid. This is particularly convenient for jumps of exponential types, where the continuous marginal probability distribution is defined with a peak point and the two-dimensional probability distribution is the linear combination of a PDF in each of the four quadrants around the peak.

2.5 Stability, Monotonicity and Consistency

Definition 2.5.1 *Modern stability analysis (for example [73, 77, 102]) defines general categories of stability under an arbitrary norm $\|\cdot\|$ using a rational polynomial $\varphi(z)$ (as in Equation 2.4.10). Nomenclature varies somewhat, so this thesis uses the following names for three cases of interest.* ***Algebraic stability*** *holds if*

$$\|\varphi(\Delta\tau \mathsf{T})^n\| \le c\, p^\alpha\, n^\beta$$

where the linear system has order $p \ge 1$, for time step $n \ge 1$, with $c, \alpha, \beta > 0$ independent of n and p. ***Strong stability*** *holds if*

$$\|\varphi(\Delta\tau \mathsf{T})^n\| \le c$$

for $c > 0$, which becomes ***strict stability*** *if $0 < c \le 1$.*

Definition 2.5.2 *An alternate definition of stability, used for analysis of the viscosity solution to a system of equations (e.g. [47, 66]) is that the time step method of Equation 2.4.9 is unconditionally strictly stable in the l_∞ norm when*

$$\left\|\mathsf{v}^{n+1}\right\|_\infty \le c \cdot \max\Big[\left\|\mathcal{I}(x_1,x_2)\right\|_\infty, \left\|B(x_1,x_2,\tau)\right\|_\infty\Big]$$

where the constant c is independent of grid spacing Δx_1, Δx_2 or $\Delta\tau$.

Definitions 2.5.1 and 2.5.2 are clearly different when applied to non-linear PIDE's.

Definition 2.5.3 *Equation 2.4.9, denoted as $\mathsf{M}_\mathsf{i}^{n+1}$ in Equation 2.4.11, represents a* ***monotone*** *scheme [47, 66] when, for all $\epsilon_\mathsf{j}^{n+1}, \epsilon_\mathsf{i}^{n}, \epsilon_\mathsf{j}^{n} \ge 0$ and $\mathsf{i} \ne \mathsf{j}$*

$$\mathsf{M}_\mathsf{i}^{n+1}(h, \mathsf{w}_\mathsf{i}^{n+1}, \mathsf{w}_\mathsf{j}^{n+1} + \epsilon_\mathsf{j}^{n+1}, \mathsf{w}_\mathsf{i}^{n} + \epsilon_\mathsf{i}^{n}, \mathsf{w}_\mathsf{j}^{n} + \epsilon_\mathsf{j}^{n}) \le \mathsf{M}_\mathsf{i}^{n+1}(h, \mathsf{w}_\mathsf{i}^{n+1}, \mathsf{w}_\mathsf{j}^{n+1}, \mathsf{w}_\mathsf{i}^{n}, \mathsf{w}_\mathsf{j}^{n}) \ . \tag{2.5.1}$$

Definition 2.5.4 *Equation 2.4.9, denoted as $\mathsf{M}_\mathsf{i}^{n+1}$ in Equation 2.4.11, is a* ***consistent*** *scheme if, for any smooth function $\phi((x_1,x_2),\tau)$ with a vector $\bar\phi$ where $\bar\phi_\mathsf{i}^n = \phi((x_1,x_2)_\mathsf{i},\tau_n)$ over the $\mathsf{i} = 1\ldots N$ finite difference points, it satisfies*

$$\left|\left(\frac{\partial\phi}{\partial\tau} - \mathcal{G}\phi - \lambda_C\mathcal{H}_D\phi\right)_\mathsf{i}^{n+1} - \mathsf{M}_\mathsf{i}^{n+1}(h, \bar\phi_\mathsf{i}^{n+1}, \bar\phi_\mathsf{j}^{n+1}, \bar\phi_\mathsf{i}^{n}, \bar\phi_\mathsf{j}^{n})\right| \to 0$$
$$\text{as} \quad h \to 0 \ .$$

Theorem 2.5.5 *The discrete Equation 2.4.11 is consistent with PIDE 2.3.2.*

Proof. This follows from Taylor series analysis of the techniques used to create the discretization. □

Definition 2.5.6 *If a matrix A has elements $\mathsf{A}_{\mathsf{ii}} > 0$ and $\mathsf{A}_{\mathsf{ij}} \le 0$ for $\mathsf{i} \ne \mathsf{j}$ and every row sum is non-negative with at least one row sum positive in each connected part of A, then A is an* ***M-matrix*** *(see [93, 98]).*

Remark 2.5.7 *If matrix A is an M-matrix then A^{-1} exists and has elements $\left[\mathsf{A}^{-1}\right]_{\mathsf{ij}} \ge 0$ [93, 98].*

Definition 2.5.8 *If a matrix A has elements $\mathsf{A}_{\mathsf{ii}} \ge 0$ and $\mathsf{A}_{\mathsf{ij}} \le 0$ for $\mathsf{i} \ne \mathsf{j}$ and each row sum is non-negative then A is* ***M-compatible****.*

Remark 2.5.9 *The sum of an M-compatible matrix and an M-matrix is an M-matrix.*

As will be shown below, if T is an M-matrix, then Crank–Nicolson time stepping is unconditionally algebraically stable in the l_∞ norm and, under a time step restriction, strictly stable in the sense both of Definition 2.5.1 and 2.5.2. In Definition 2.5.1, stability implies that at time step n the error $E_n = \overline{W}_n - W_n$ due to a perturbed solution $\overline{W}_n$ can be bounded in terms of the initial error $E_0 = \overline{W}_0 - W_0$ where $\overline{W}_0$ is a perturbed initial solution. This definition is useful for analyzing the fixed point iteration for time step solution in Section 2.6 below and is similar to the one used in von Neumann analysis.

Crank–Nicolson time stepping will be shown to be unconditionally stable in the l_2 norm in the sense of von Neumann analysis. Under von Neumann analysis the conditions are determined under which the finite difference and integral operator reduce, rather than amplify, the l_2 norm of a perturbation error $E_n = \overline{W}_n - W_n$ as it propagates to time step $n + 1$.

2.5.1 Monotonicity and Stability in the l_∞ Norm

2.5.1.1 European Options

Theorem 2.5.10 *If* T *is an M-matrix then the time step method of Equation 2.4.9 is* ***unconditionally algebraically stable*** *in the* l_∞ *norm for* $\theta = 1/2$*. The time step method is* ***monotone*** *and* ***unconditionally strictly stable*** *in the* l_∞ *norm for* $\theta = 0$*.*

Equation 2.4.9 is ***monotone*** *and* ***strictly stable*** *in the* l_∞ *norm for* $\theta = 1/2$ *if the time step is bounded using the maximum diagonal of* T *such that*

$$\frac{\Delta\tau}{2} \max_{\mathsf{i}}(\mathsf{T}_{\mathsf{ii}}) < 1 \ .$$

Proof. For stability in the sense of Definition 2.5.1 see Kraaijevanger et al. [73]. For strict stability in the sense of Definition 2.5.2 this result is proved by simple maximum analysis (e.g. as an extension of the result of [37]) which includes the boundary conditions. Monotonicity follows from the definition of an M-matrix. □

The finest grid spacing in the problem determines the maximum Crank–Nicolson time step for which the solution is strictly stable in the l_∞ norm. A fully implicit time stepping provides a monotone, consistent and stable method, which are the conditions required in [11, 18, 33, 34] to demonstrate convergence to the viscosity solution.

2.5.1.2 American Options

For American options, if T in Equation 2.4.8 is an M-matrix, then it is straightforward to extend the analysis in [37] to show that fully implicit time stepping ($\theta = 0$) coupled with a penalty method [48] is unconditionally stable and monotone in the sense of Definitions 2.5.2.

2.5.2 Stability in the l_2 Norm

The von Neumann stability analysis examines a problem over a periodic domain to determine conditions for the stability of its discrete operators under the l_2 norm ([40] §8.3 and [97] §6.8). It can be related to algebraic stability, in that the necessary conditions for both can be equivalent [101] §3 in some situations.

Theorem 2.5.11 *Consider a periodic formulation of Equation 2.3.2, discretized with a finite difference approximation on a grid with constant spacing. The problem is formed with constant coefficients for the drift and diffusion terms, $-1 \leq \rho \leq 1$ and $\lambda_C = \lambda \geq 0$ constant. Either the cross-partial finite difference of Equations E.1.5 or E.1.6 are used to form the approximation G_s to $\mathcal{G}$ on a seven-point stencil, or Equation E.1.7 used to form G_n over a nine-point stencil.*

The time step Equation 2.4.9 is unconditionally von Neumann stable in the l_2 norm for $\theta = 0$ and for $\theta = 1/2$.

Proof. See Appendix F, in particular Appendix F.5 Remark F.5.1. □

2.5.3 Monotonicity and the Finite Difference Approximation

Ensuring that T of Equation 2.4.9 is an M-matrix is the main obstacle to guaranteeing Theorem 2.5.10 holds. In the non-linear case considered in Chapter 4, this is an essential requirement for convergence to the viscosity solution.

Remark 2.5.12 *Let the conditions*

- $r > 0$,
- $(-\mathsf{G})$ *is M-compatible,*
- $\mathsf{J} \geq 0$ *and*
- $(\max_{\mathsf{i}} \sum_{\mathsf{j}} \mathsf{J}_{\mathsf{ij}}) \leq 1$ *hold.*

Then, from Definition 2.5.6, T *of Equation 2.4.8 is an M-matrix.*

A discretization that satisfies the conditions in Remark 2.5.12 will generate positive off-diagonal and negative diagonal coefficients in G, $(\mathsf{J} - \mathsf{I})$ and hence $(-\mathsf{T})$, so is also called a **positive coefficient discretization** [47].

The discrete integral operator matrix $\mathsf{J} \geq 0$ is explicitly constructed in Section 2.4.3.3 to have a maximum row sum $(\max_{\mathsf{i}} \sum_{\mathsf{j}} \mathsf{J}_{\mathsf{ij}}) \leq 1$ and thus ensures that $-(\mathsf{J} - \mathsf{I})$ is M-compatible. Since $r > 0$ holds, the M-compatibility of $(-\mathsf{G})$ remains as the last condition of Remark 2.5.12 left to satisfy. To ensure $(-\mathsf{G})$ is compatible with an M-matrix a number of conditions may be applied or approaches taken, listed as follows with the names that will be used to denote them.

1. **"Central differencing as much as possible"**
   The second order accurate, central difference FD stencil for first partial differential terms makes negative off-diagonal contributions to G. Suppose, for example, the discrete derivative terms in G for one direction along an axis at solution point $\mathsf{v_b}$, with equidistant, adjacent nodes $\mathsf{v_a}$ and $\mathsf{v_c}$, result in a diffusion term

$$\beta_1 \mathsf{v_a} - (\beta_1 + \beta_2)\mathsf{v_b} + \beta_2 \mathsf{v_c}, \quad \beta_{1,2} > 0$$

   and by central differencing and first order differencing the drift terms result in the expressions

$$-\alpha_1 \mathsf{v_a} + \alpha_1 \mathsf{v_c} \quad \text{and} \quad -\alpha_2 \mathsf{v_b} + \alpha_2 \mathsf{v_c}\ , \quad \alpha_1, \alpha_2 > 0$$

respectively. Central differencing will be used as much as possible to minimize discretization error. Only when $\beta_1 - \alpha_1 < 0$ will the first order difference be used instead to preserve the M-compatibility of $(-\mathsf{G})$. This approach is used in both x_1 and x_2 directions. Further details are omitted, since this is standard and completely analogous to the one dimensional case as described in [38, 113].

Remark 2.5.13 *It is taken as given in the remainder of this work that central differencing is applied as much as possible, with first order differencing where required, to the discrete drift term for proofs which require* $(-\mathsf{G})$ *to be M-compatible.*

2. **"Skewed Grid"**
   An approach to eliminating negative diffusion coefficients is to transform the problem into a skewed co-ordinate system as proposed by Hull and White [62, 112]

$$(\psi_1, \psi_2) = (\sigma_2 x_1 + \sigma_1 x_2, \sigma_2 x_1 - \sigma_1 x_2) \tag{2.5.2}$$

   which will result in a zero diffusion correlation in the transformed system. This is, in effect, a $\pi/4$ rotation of the grid followed by a scaling in either the x_1 or x_2 direction. See Appendix E.2 for details of the transformed partial differential terms.

3. **"Spacing Restricted Grid"**
   An alternative to Item 2 is to enforce a spacing restriction on the FD grid. For the following theorem, assume that the discretization is performed on a rectangular grid with constant grid line spacing. Denote the grid spacing as $(\Delta x_1, \Delta x_2)$ in the (x_1, x_2) direction.

Theorem 2.5.14 *The finite difference discretization of the second partial differential terms of* $\mathcal{G}$ *is compatible with an M-matrix if the following constraints hold when* $\rho \neq 0$.

(a) For cross-partial derivatives, select the seven-point FD stencil which includes $V(x_1 + \Delta x_1, x_2 + \Delta x_2)$ *and* $V(x_1 - \Delta x_1, x_2 - \Delta x_2)$ *if* $\rho > 0$ *(Equation E.1.5), and the seven-point stencil which includes* $V(x_1 + \Delta x_1, x_2 - \Delta x_2)$ *and* $V(x_1 - \Delta x_1, x_2 + \Delta x_2)$ *if* $\rho < 0$ *(Equation E.1.6).*

(b) For each point on the grid

$$\frac{{\sigma_1}^2}{\Delta x_1} \geq \left| \frac{\rho \sigma_1 \sigma_2}{\Delta x_2} \right| \text{ and } \frac{{\sigma_2}^2}{\Delta x_2} \geq \left| \frac{\rho \sigma_1 \sigma_2}{\Delta x_1} \right|$$

which can be written as

$$\left| \rho \frac{\sigma_2}{\sigma_1} \right| \leq \frac{\Delta x_2}{\Delta x_1} \leq \left| \frac{1}{\rho} \frac{\sigma_2}{\sigma_1} \right| . \tag{2.5.3}$$

With these conditions and item 1 above, $(-\mathsf{G})$ *is compatible with an M-matrix.*

Proof. See [11] and [89] §9.4. □

The variable-spaced finite difference approach used here has the same restriction, except applied to both spacings $\Delta x_1^{\pm}$ and $\Delta x_2^{\pm}$ on the $\pm x_1$ and $\pm x_2$ sides of the central point of the stencil.

Remark 2.5.15 *Consider the finite difference approximation G_n to $\mathcal{G}$ on a nine-point stencil with the four-point second order cross-partial derivative given by Equation E.1.7. This formulation results in negative off-diagonal coefficients in G_n for $\rho \neq 0$. Thus this discretization does not result in $(-\mathsf{G})$ M-compatible for correlation $\rho \neq 0$ and Theorem 2.5.10 cannot be shown to hold by the M-matrix approach.*

4. **"Rotated Grid"**
   A third method to generate a monotone diffusion term discretization is to rotate the finite difference grid by [112]
   $$\theta_r = \frac{1}{2}\tan^{-1}\left(\frac{2\rho\sigma_1\sigma_2}{{\sigma_1}^2 - {\sigma_2}^2}\right) \tag{2.5.4}$$
   which will result in a zero correlation in the diffusion tensor of the rotated system. The cross-partial derivative is thus eliminated and the FD approximation to the second partial derivatives is always compatible with an M-matrix. See Appendix E.3 for details of the transformed partial differential terms.

There are two notable alternative approaches to the problem of monotone diffusion discretization [10]: Generalized Finite Differences [14, 15] and edge swapping with a finite volume method [113]. These are not considered since they may result in a non-local discretization that does not respect option contract payoff features or price barriers.

2.5.4 Stability Summary

The numerical approach can deviate from the theoretical conditions for stability. The FD method employed for grids with non-constant spacing has the same conditions for monotonicity as Theorem 2.5.14. However, the structure of a computationally efficient FD grid is not always such that these grid spacing conditions are met. Such a grid may not be able to guarantee the conditions for M-compatibility at every point in the solution domain, although the conditions will often be met locally in the region of interest near the strike price of the option. In turn, this has implications for both European and American options: the conditions for l_∞ norm stability cannot be globally guaranteed. This issue was studied in [113] for the pure diffusion case, where it was shown that if the option value is Lipschitz continuous then coefficients in the discretization which are not M-compatible cause at worst an $O(h)$ error. The convergence of the method was demonstrated numerically. Note, also, that the contribution to the linear system from the integral term works in favour of stability: it tends to correct, rather than worsen, the problem of the differential term not being M-compatible.

The von Neumann stability analysis fails to apply to the method for European options where the finite difference grid spacing is not constant in each direction. Jump rate λ_C is non-constant, although the analysis demonstrates stability where either $\lambda = 0$ or $\lambda > 0$ over the entire domain. Nonetheless, the von Neumann analysis is interesting because it provides a less restrictive result than the l_∞ norm analysis, indicating strict l_2 norm stability regardless of the ratio of grid spacing between the axes, the time step and the choice of discretization for the cross-partial derivatives.

The numerical demonstrations in Section 2.7 show the numerical method with Crank–Nicolson time stepping is quadratically convergent for European and American options despite the violations of the theoretical stability conditions that have been identified.[5] Monotone spatial discretization can be guaranteed by, for instance, employing a rotated co-ordinate system which eliminates the correlation in the Brownian motion. However, as shown in Section 2.7.4, this approach can result in higher errors in the solution. Chapter 3 investigates this issue fully in the context of stochastic volatility problems taken from actual markets, comparing error control and computational cost for all three fully monotone schemes. Following from that comparison, the grid rotation approach will be used in Chapter 4 to ensure that this important condition for convergence to the viscosity solution condition holds for a non-linear problem.

2.6 Solution of the Discrete Equations

The previous sections show how PIDE 2.2.3 has been localized to PIDE 2.3.2 and discretized. This section focuses on how the time step Equation 2.4.9 is solved using a fixed point iteration and how American option values are computed using a penalty method. Algorithm 2.6.1 gives the resulting combined fixed point, penalty iteration.

2.6.1 Fixed Point Iterative Solution for One Time Step

As a motivation for the fixed point iteration scheme consider the semi-discretized Equation 2.4.2. A fixed point iteration scheme, whereby the integral terms are handled in an iterative manner, avoids having to use the Green's and Poisson functions of the PIDE. Rather, only the Green's and Poisson functions of the differential terms are required.

2.6.1.1 Semi-Discrete Form

Let Z^k be the k-th iterate towards a solution W^{n+1} of Equation 2.4.2. One step of the iteration is given by

$$\begin{aligned}\left[1-(1-\theta)\Delta\tau\left(\mathcal{G}-r-\lambda_C\right)\right]Z^{k+1} &= (1-\theta)\Delta\tau\,\lambda_C\mathcal{J}_D Z^k \\ &\quad + \left[1+\theta\Delta\tau\left(\mathcal{G}+\lambda_C\mathcal{J}_D-r-\lambda_C\right)\right]W^n\end{aligned} \tag{2.6.1}$$

which is repeated until convergence.

[5] In one case, constructed inadvertently with an extreme spatial grid spacing change, oscillations did appear even though implicit time steps were taken and upwind weighting was used for the drift terms. A sufficiently poor choice of grid spacing can indeed cause oscillations in the solution.

Theorem 2.6.1 *Let $E^k = W^{n+1} - Z^k$ be the error in the solution to the semi-discretized Equation 2.4.2 at iteration k of the functional iteration given in Equation 2.6.1. The iteration is convergent to zero as*

$$\|E^{k+1}\|_\infty \leq \frac{(1-\theta)\Delta\tau\,\lambda}{1+(1-\theta)\Delta\tau\,(r+\lambda)}\|E^k\|_\infty \ .$$

Proof. See Appendix D.1. □

2.6.1.2 Fully Discrete Form

Now consider the solution of the fully discrete problem in Equation 2.4.9. To avoid having to invert any matrix such as T of Equation 2.4.8 formed by a sum containing the dense matrix J, use the discrete version of the iteration of Equation 2.6.1.

Let z^k be the k-th iterate towards a solution w^{n+1} of Equation 2.4.9. The fixed point iteration is given by

$$\begin{aligned}\{\mathsf{I} - (1-\theta)\Delta\tau\,[\mathsf{G} - r\,\mathsf{I} - \lambda\mathsf{I_c}]\}\,\mathsf{z}^{k+1} &= (1-\theta)\Delta\tau\,\lambda\mathsf{I_c}\mathsf{J}\mathsf{z}^k \\ &\quad + \{\mathsf{I} + \theta\Delta\tau\,[\mathsf{G} + \lambda\mathsf{I_c}(\mathsf{J}-\mathsf{I}) - r\,\mathsf{I}]\}\,\mathsf{w}^n + \mathsf{b}\ .\end{aligned} \tag{2.6.2}$$

The computation $\mathsf{I_c}\mathsf{J}\mathsf{z}^k$ is done using the FFT-based method of Equations 2.4.19 and 2.4.20.

The iteration is repeated until an update is obtained which satisfies a relative tolerance of ϵ_u such that

$$\max_{\mathsf{i}\in 1\ldots N} \frac{\left|\mathsf{z}^{k+1} - \mathsf{z}^k\right|_\mathsf{i}}{\max\left(1, |\mathsf{z}^k|_\mathsf{i}\,, |\mathsf{z}^{k+1}|_\mathsf{i}\right)} < \epsilon_u \ .$$

The constant of one in this convergence test is set assuming that the typical option value is of the order of $\mathsf{w} = 1$. This ensures that the convergence requirement does not become extreme for grid points with small option values. Section 2.7.5 discusses the actual amount of computation required for each time step.

Theorem 2.6.2 *Let $\mathsf{e}^k = \mathsf{w}^{n+1} - \mathsf{z}^k$ be the error in the solution to Equation 2.4.9 at iteration k of the fixed-point iteration given in Equation 2.6.2. If $\mathsf{J} \geq 0$ has maximum row sum $(\max_\mathsf{i} \sum_\mathsf{j} \mathsf{J_{ij}}) \leq 1$ and $(-\mathsf{G})$ is M-compatible with $\|\mathsf{G}\cdot\underline{1}\|_\infty = 0$, then the error in the iterative solution z^{k+1} in Equation 2.6.2 is convergent to zero as*

$$\|\mathsf{e}^{k+1}\|_\infty \leq \frac{(1-\theta)\Delta\tau\,\lambda}{1+(1-\theta)\Delta\tau\,(r+\lambda)}\|\mathsf{e}^k\|_\infty \ .$$

Proof. $\|\mathsf{G}\cdot\underline{1}\|_\infty = 0$ should hold for any consistent finite difference approximation. See Appendix D.2. □

Theorem 2.6.3 *Consider a periodic formulation of Equation 2.3.2, discretized with a finite difference approximation. Let* $(-\mathsf{G})$ *be formed either by the 7-point or 9-point finite difference stencil on a grid with constant spacing, as in Theorem 2.5.11.*

Then in the sense of von Neumann analysis the iterative solution to Equation 2.4.9 by Equation 2.6.2 is unconditionally convergent in the l_2 *norm (i.e. regardless of whether* $(-\mathsf{G})$ *is M-compatible) at a rate which is rapid if* $\lambda\Delta\tau \ll 1$.

Proof. See Appendix F, in particular Appendix F.5 Remark F.5.2. □

For most practical situations, $\lambda\Delta\tau \ll 1$ and convergence should be rapid. To summarize, this section has shown the following results.

- The functional fixed-point iteration Equation 2.6.1 for the semi-discrete Equation 2.4.2 is convergent in the l_∞ norm. This suggests that, for a sufficiently fine grid, the discrete iteration should also be convergent.
- If $(-\mathsf{G})$ is M-compatible then by maximum analysis the discrete fixed point iteration is convergent in the l_∞ norm.
- By von Neumann analysis, the iteration is convergent in the l_2 norm for a periodic problem with constant grid spacing, with no restrictions on grid spacing ratio or time step, using any standard second-order finite difference approximation of the differential operators.

2.6.2 American Options by Penalty Iteration

To solve the discrete, localized version of the linear complementarity problem of Equations 2.2.10 and 2.2.11, the penalty iteration of [48, 111] is applied. In [37] this method was shown to be l_∞ stable for jump-diffusion processes provided that the discrete diffusion portion of the process was M-compatible and that iterations of the form of Equation 2.6.2 should be rapidly convergent. If the discretized diffusion operator is an M-matrix then for fully implicit time stepping the method is l_∞ stable and monotone, as proved by maximum analysis. Since a consistent scheme is used for the differential and integral terms, one may expect convergence to the viscosity solution of the localized problem [3, 18, 19, 37, 91]. Note that the concept of a viscosity solution permits non-smooth solutions.

Define a penalty vector $\mathsf{c}^k \in \mathbb{R}^N$ for iterate z^k with elements

$$(\mathsf{c}^k)_\mathsf{i} = \begin{cases} \mathit{Large} & \text{if } \mathsf{w}^*_\mathsf{i} > \mathsf{z}^k_\mathsf{i} \\ 0 & \text{if } \mathsf{w}^*_\mathsf{i} \leq \mathsf{z}^k_\mathsf{i} \end{cases} \quad \forall \mathsf{i} = 1 \ldots N \tag{2.6.3}$$

where $\mathsf{w}^*_\mathsf{i} = \mathcal{I}((x_1, x_2)_\mathsf{i})$ is the vector of option payoff values, the minimum value of an American option at any time. A value of *Large* $\simeq 10^5$ is usually sufficiently large to impose the condition without causing numerical inaccuracy.

To impose the American constraint, Equation 2.6.2 is modified by adding the term

$$\left(\mathsf{c}^k\right)' \mathsf{I} \left(\mathsf{w}^* - \mathsf{z}^{k+1}\right)$$

to the RHS, where $\left(\mathsf{c}^k\right)'$ denotes the transpose of c^k. Vector c^k is updated before each linear system is constructed and solved, iterating from $\mathsf{z}^0 = \mathsf{w}^n$ for solutions z^{k+1} for $k = 0$ to convergence. The penalty iteration is incorporated into the fixed point iteration Equation 2.6.2 without adding another level of iteration. The resulting combined penalty, fixed point iteration for non-linear system solution is given as Algorithm 2.6.1.

Intuitively, the penalty iteration may be thought of as the adaptive imposition of a Dirichlet free boundary condition. The same approach can be used to impose a maximum value on an option.

2.6.3 Linear System Solution

Each iteration towards the solution of a time step requires solving the linear system given in Equation 2.6.2. In the case of American options, the linear system also contains a penalty constraint as in Step 5 of Algorithm 2.6.1. For one-factor options, a direct solution method based on Gaussian elimination is suitable. For two-factor options a direct method would be unacceptably expensive, thus a preconditioned, Krylov-subspace, iterative method is used. Following the results of [113, 37, 98] and after some experimentation, Bi-CGStab was selected combined with an ILU(2) preconditioner and RCM re-ordering. A relative point-wise update to $\mathsf{z}_\mathsf{i}^{k+1}$, at each node i, of no more than $\epsilon_l \cdot \max(1, \mathsf{v}_\mathsf{i}^n)$ must be attained in both Bi-CGStab update vectors, hence in the total solution update, before it is accepted as a solution.

2.7 Numerical Demonstrations

For the numerical demonstrations, the value of three option contracts were computed with two different types of jump-diffusion. The convergence was tested as $\Omega_C, \Omega_D \to \pm\infty$. Up to quadratic convergence to the solution of both European and American options is shown as the number of discrete solution nodes is increased in time and log-price.

2.7.1 Two-Asset Sample Problems: Two Markets, Three Contracts

Two jump PDF functions were chosen: the bi-variate Normal and the Marshall-Olkin Bi-variate Exponential Distribution (MOBED). Both are described in detail in Appendix B. As noted previously, these correspond in form to the well-known, one-asset models of Merton [86] and of Kou [72]. Table 2.7.1 lists the model coefficients, which are of a magnitude that is plausible for a real market.[6]

As a demonstration, the value of a European call on the maximum of two assets and a European and American put on the minimum of two assets were computed. Each option had a strike $K = 100$, an expiry of $T = 1$ and a risk free rate $r = 0.05$. A high-accuracy, semi-analytic solution, computed using the method described in [46], was used for comparison with the numerical solution of the European options.

[6] Compare the values in Table 3.5.1 of Chapter 3 which are drawn from statistical fits to actual markets.

	Jump Distribution	
Diffusion Parameter	**Normal**	**MOBED**
σ_1	0.12	0.12
σ_2	0.15	0.15
ρ	0.30	0.30
Jump Parameter		
λ	1.20	1.00
$\check{\mu}_1$	-0.10	0.00
$\check{\mu}_2$	0.10	0.00
$\check{\rho}$	-0.40	
$\check{\sigma}_1$	0.12	
$\check{\sigma}_2$	0.15	
$\check{p}_1$		0.40
$\check{p}_2$		0.60
$\check{\eta}_{p,1}$		1/0.20
$\check{\eta}_{p,2}$		1/0.18
$\check{\eta}_{q,1}$		1/0.15
$\check{\eta}_{q,2}$		1/0.14
$\check{\eta}_{pp}$		1/0.15
$\check{\eta}_{qq}$		1/0.16
$\check{\eta}_{pq}$		1/0.12
$\check{\eta}_{qp}$		1/0.15

Table 2.7.1: The two parameter sets used for the numerical examples in this chapter. These are not taken from actual market values, but represent parameter values in a plausible range for a real market process. See Appendix B for the definitions of the jump distribution coefficients. Solutions were computed with $r = 0.05$, strike $K = 100$ and expiry $T = 1$.

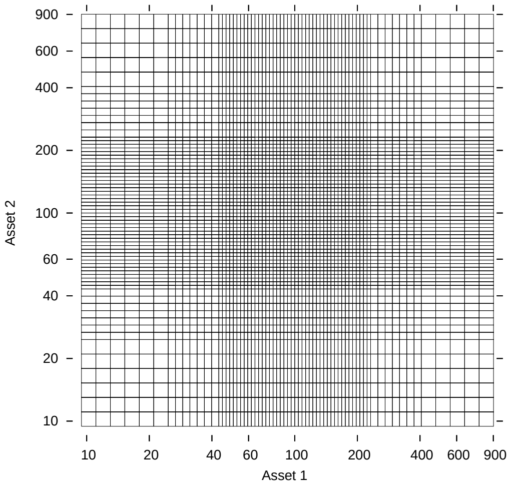

Figure 2.7.1: This coarse grid in log-price scaling over Ω_D shows grid line concentration in the Ω_C region around the strike of $K = 100$. The actual grids were extended very coarsely to approximately $S_1, S_2 = 11 \ldots 900$ to capture enough of the solution to control the error from the jump-diffusion computation.

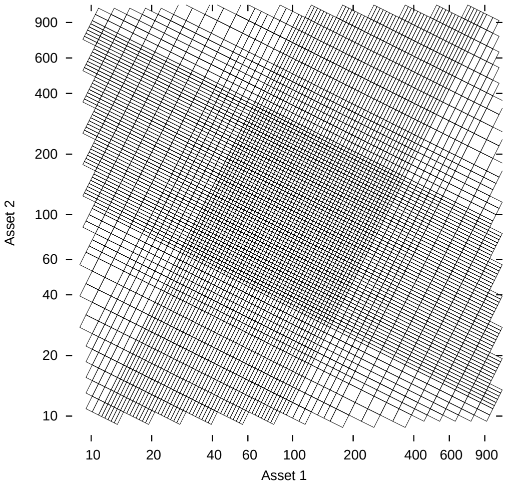

Figure 2.7.2: This coarse grid in log-price scaling over Ω_D, with node concentration in the Ω_C region around the strike of $K = 100$, is rotated by -26.57 degrees around the strike node. This ensures that the spatial finite difference approximation to the problem specified in Table 2.7.1 generates an M-matrix.

Algorithm 2.6.1 Solve one time step using a simultaneous fixed point, penalty iteration.

```
FixedPointIteration( $\mathsf{w}^n$, $\mathsf{w}^*$, $\theta$, $\Delta\tau$, $\mathsf{T}$, $\mathsf{G}$, $\mathsf{J}$, $\lambda\mathsf{I_c}$, $\mathsf{b}$, $\epsilon_u$ )
where
        $\mathsf{w}^n$                      the price at time step $n$
        $\mathsf{w}^*$                      the minimum option value (usually the payoff)
        $\theta$                      time step weight
        $\Delta\tau$                     time step size
        $\mathsf{T}, \mathsf{G}, \mathsf{J}, \lambda\mathsf{I_c}$              the discrete PIDE to be solved, Equation 2.4.8
        $\mathsf{b}$                      boundary condition imposition vector
        $\epsilon_u$                     required solution update tolerance

1.  Set $\mathsf{z}^{k=0} = \mathsf{w}^n$.

2.  For $k = 0, 1, 2, 3, \ldots$ until convergence (tested in Step 6)

3.      For American options: set $\mathsf{c}^k$ using Equation 2.6.3.
        For European options: set $\mathsf{c}^k = 0$.

4.      Set     $\mathsf{N}^k = [\mathsf{I} - (1-\theta)\Delta\tau\,(\mathsf{G} - r\,\mathsf{I} - \lambda\mathsf{I_c})] + (\mathsf{c}^k)'\mathsf{I}$
                $\mathsf{y}^k = [(1-\theta)\Delta\tau\;\lambda\,\mathsf{I_c}\mathsf{J}]\,\mathsf{z}^k + [\mathsf{I} - \theta\Delta\tau\;\mathsf{T}]\,\mathsf{w}^n + \mathsf{b} + (\mathsf{c}^k)'\mathsf{I}\,\mathsf{w}^*$
        where
                $\mathsf{I_c}\mathsf{J}\mathsf{z}^k \simeq \mathsf{I_c}\,(\mathsf{K}\cdot\mathsf{F}\cdot\mathsf{L})\,\mathsf{z}^k$.        (see Equation 2.4.20)

5.      Solve   $\mathsf{N}^k\,\mathsf{z}^{k+1} = \mathsf{y}^k$    using ILU(2) preconditioned, Bi-CGStab.
                                                        (see Section 2.6.3)

6.      If $\max_{\mathsf{i}} \dfrac{\left|\mathsf{z}^{k+1} - \mathsf{z}^k\right|_{\mathsf{i}}}{\max\left(1, \left|\mathsf{z}^k\right|_{\mathsf{i}}, \left|\mathsf{z}^{k+1}\right|_{\mathsf{i}}\right)} < \epsilon_u$ then the iteration is finished.

    End For

Return the solution vector $\mathsf{w}^{n+1} = \mathsf{z}^{k+1}$.
```

The region near $(x_1, x_2) = (\log(K), \log(K))$ was discretized with a constant grid spacing $(\Delta x_1, \Delta x_2)$. Crank–Nicolson time stepping was used with a constant time step $\Delta\tau$, so that convergence could be demonstrated with respect to a well controlled amount of computational effort. Variable grid spacing [94] or time stepping [112] can provide computational savings, but this thesis does not investigate these issues. Figure 2.7.1 shows the coarsest grid used for the demonstrations of Section 2.7.3 below. Over $\Omega_D \setminus \Omega_C$ solutions are expected to be mostly piecewise linear in price scaling, hence that region remains only coarsely resolved. Regions of the problem generated discrete equations which did not result in an M-matrix. However, the overall problem solution proceeded with no actual numerical oscillations detected in the region of interest around the strike. This is consistent with previous efforts [113] for pure diffusion models.

The DFT grid spacing was set to $(\Delta x_{f,1}, \Delta x_{f,2}) = (2\Delta x_1, 2\Delta x_2)$, twice the constant FD grid spacing $(\Delta x_1, \Delta x_2)$ near the strike node at $(\log(K), \log(K))$. The smallest integers N_1 and $N_2 = 2^a\, 3^b\, 5^c\, 7^d$, $a, b, c, d \in \mathbb{Z}$ where $a \geq 1$, $b, c, d \geq 0$ were selected (as dictated by the choice of FFT solution package[7]) such that the DFT grid of $N_F = N_1 \times N_2$ nodes overlapped the finite difference grid fully. Recall that $N_1 \neq N_2$ in general, although in the following experiments the two values were equal. This approach resulted in a number of DFT grid nodes which was of the same order as the number of FD grid nodes.

The error tolerance parameters used for the grid refinement demonstrations were

- $\epsilon_c = 10^{-3}$, which must be satisfied by the size of Ω_C (Equation 2.3.12),
- $\epsilon_d = 10^{-4}$, which dictates the minimum distance from $\partial\Omega_C$ to $\partial\Omega_D$ (Equation 2.3.13),
- $\epsilon_u = 10^{-7}$ for a relative update to a time-step solution by fixed point or combined fixed point, penalty iteration (Section 2.6.1) and
- $\epsilon_l = 10^{-8}$ for a relative update by Bi-CGStab to a linear system solution (Section 2.6.3).

2.7.2 Convergence as $\Omega_C, \Omega_D \to \Omega_\infty$

As a partial demonstration of Theorem 2.3.7, four solutions were computed for the European call on the maximum of two assets using grids of increasing size. The four ranges for Ω_C and Ω_D are given in price scaling in the left column of Table 2.7.2. Each is roughly symmetric around the strike node at $(x_1, x_2) = (\log(K), \log(K))$. The Ω_D size is not a round number because it is extended so that the grid retains a constant, but coarse spacing near the boundary in each direction.

In order to focus on the effect of the localization error, all grids had a spacing at the strike of $\Delta x_1 = \Delta x_2 = 0.02$ and each larger grid was formed as a simple extension of the previous one. In other words, this does not reduce discretization error in an attempt to converge to the exact solution, but rather examines the effect of the localization error for a fixed grid spacing. The time step was $\Delta\tau = 0.02$ and the DFT grids had a spacing of $\Delta x_{f,1} = \Delta x_{f,2} = 0.04$.

[7] The FFTW library, available at `http://fftw.org`, implements an efficient Winograd transform algorithm.

Grid Range		**Normal**			**Difference vs. Largest Ω_D**		
(Price Scale)	S_2	$S_1 = 90$	100	110	90	100	110
$\Omega_C : 60 \to 165$	90	11.3294	16.5689	23.4341	-0.0590	-0.0455	-0.0434
$\Omega_D : 39.1 \to 256$	100	15.7163	20.4293	26.7527	-0.0582	-0.0485	-0.0530
DFT 60 × 60	110	21.5055	25.4602	31.0294	-0.0807	-0.0771	-0.0893
$\Omega_C : 50 \to 200$	90	11.3818	16.6109	23.4749	-0.0066	-0.0034	-0.0025
$\Omega_D : 19.0 \to 525$	100	15.7710	20.4765	26.8047	-0.0034	-0.0012	-0.0009
DFT 90 × 90	110	21.5845	25.5368	31.1180	-0.0018	-0.0005	-0.0007
$\Omega_C : 25 \to 400$	90	11.3884	16.6144	23.4775	0.0001	0.0001	0.0001
$\Omega_D : 9.4 \to 903$	100	15.7746	20.4779	26.8058	0.0001	0.0001	0.0001
DFT 112 × 112	110	21.5865	25.5375	31.1190	0.0003	0.0003	0.0003
$\Omega_C : 20 \to 500$	90	11.3884	16.6143	23.4774			
$\Omega_D : 8.7 \to 1147$	100	15.7745	20.4778	26.8057			
DFT 126 × 126	110	21.5862	25.5372	31.1187			

Table 2.7.2: To show the effects of extending the domain, the value of the European call on the maximum of two assets is computed using the Normal jumps model. The Ω_C and Ω_D ranges are given in the left column in price scaling, along with the DFT grid size. The FD grid has a spacing of $\Delta x_1 = \Delta x_2 = 0.02$ near the strike and the time step was fixed at $\Delta\tau = 0.02$. Crank–Nicolson time stepping was used. Points at $S_1, S_2 = \{90, 100, 110\}$ are given for each grid. The three larger grids were each formed by extending the next smallest grid with additional lines. The error is measured against the solution over the largest domain $\Omega_D : 8.7 \to 1147$. As the upper and lower limits of Ω_C and Ω_D are extended, the difference between grids tends to diminish.

Table 2.7.2 shows the option values for the problem under Normally distributed jumps and also shows the difference measured against the solution on the largest domain. Note that the first two grids are, using the ϵ_c and ϵ_d specified above, somewhat too small around a small region of interest from $S_1, S_2 = 80 \to 125$. Since the convergence theorem does not address specific cases, it may only be noted that the error tends to diminish as the domains are extended and that the difference between the computations over the two largest domains is of the order of the ϵ_c and ϵ_d parameters.

Theorem 2.3.8 notes that, as the domain size increases, a perturbed boundary condition should generate a smaller error in the solution. The computations of Table 2.7.2 were repeated with a +50% lower Dirichlet boundary condition error. No significant difference to the values reported in Table 2.7.2 was noted. Recall that in the $\Omega_D \setminus \Omega_C$ region only the diffusion equation is solved. The rapid decay of error from the boundary condition is consistent with the results of [69, 112]. These results indicate that the propagation of that error into the interior domain

European

	TS	FD	TS	FD	DFT	FP	per	Linear	per
Grid	$\Delta\tau$	$\Delta x_{1,2}$	N_τ	N	$N_{1,2}$	Iters	TS	Iters	FP
1	0.04	0.04	25	4489	60	106	4.2	318	3.0
2	0.02	0.02	50	17689	112	180	3.6	540	3.0
3	0.01	0.01	100	70225	224	320	3.2	960	3.0

American

1	0.04	0.04				108	4.3	324	3.0
2	0.02	0.02				202	3.4	606	3.0
3	0.01	0.01				338	3.4	1014	3.0

Table 2.7.3: For each grid refinement, the time step and finite difference (FD) grid spacing near the strike are listed. The number of time steps N_τ, FD grid nodes N and FFT grid nodes in each dimension N_1, N_2 are listed. The total number of fixed point (FP) and linear solver iterations required to complete the solution of the European and American put options on the minimum of two assets are given for the test case with Normally distributed jumps. Crank–Nicolson time stepping was used. Also shown are the average number of fixed point iterations required to advance each time step and the average number of linear solver iterations required to solve each fixed point iteration. The values for the MOBED test case and for the European call on the maximum of two assets were approximately equal to these values.

by the jump process is too small to note.

2.7.3 Convergence with Grid Refinement

In this section, convergence is examined with respect to grid and time step refinement using $\Omega_D = [9.4, 903] \times [9.4, 903]$, $\Omega_C = [25, 400] \times [25, 400]$ (in price scaling). The error due to the domain size, using Table 2.7.2, was $\sim 10^{-3}$ smaller than the best absolute error in the experiments which follow. The coarsest FD grid has a spacing of $\Delta x_1 = \Delta x_2 = 0.04$ in a region around the strike as shown in Figure 2.7.1. For each of the next two grids the grid spacing was refined by two, doubling the number of finite difference grid lines in each direction. At each grid refinement, the time step is halved from the coarse grid value of $\Delta\tau = 0.04$. The DFT grid had a spacing of $\Delta x_{f,1} = \Delta x_{f,2} = 0.08$ on the coarsest grid, which was also refined along with

Grid		**MOBED**			**Absolute Error** ϵ		
	S_2	$S_1 = 90$	100	110	90	100	110
1	90	7.3114	11.5232	18.2544	0.3161	0.2377	0.1863
	100	11.8378	15.0514	20.6943	0.3448	0.2841	0.2859
	110	18.6118	20.8044	24.9160	0.4007	0.4045	0.3534
2	90	7.0680	11.3398	18.1083	0.0728	0.0543	0.0402
	100	11.5751	14.8357	20.4730	0.0820	0.0684	0.0646
	110	18.3068	20.4956	24.6456	0.0956	0.0957	0.0830
3	90	7.0140	11.2993	18.0787	0.0188	0.0138	0.0106
	100	11.5138	14.7842	20.4253	0.0208	0.0169	0.0168
	110	18.2358	20.4246	24.5835	0.0246	0.0247	0.0209
Convergence Exponent α Grid 2 to 3	90				2.0	2.0	1.9
	100				2.0	2.0	1.9
	110				2.0	2.0	2.0

Table 2.7.4: Numerical solution of the European call on a maximum of two assets with the parameters given in Table 2.7.1. The MOBED model results are reported with an absolute error. Note that the approximate asymptotic convergence, reported here for the two finest grids, approaches the ideal quadratic $\mathrm{O}((\Delta\tau)^2, (\Delta x)^2)$ as the grid and time step is refined. Crank–Nicolson time stepping was used. The error was computed by comparison with the semi-analytic solution [46].

each FD grid. Table 2.7.3 lists the grid spacing and time step sizes, as well as the time step and grid node counts at each resolution. The objective was to form a coarse grid scenario that generated an error of the order of 1% to 3% for European options.

The grid sizes and the computed results for the European call on the maximum are given in Table 2.7.4 for the MOBED jumps model. The semi-analytic solution, computed using Fourier methods [46], allows a highly accurate measure of the error in the numerical scheme. If ϵ is the error in the numerical solution, the working assumption is that $\epsilon = c\Delta {x_1}^{\alpha}$ and $\Delta x_2, \Delta\tau = \mathrm{O}(\Delta x_1)$ (see Appendix E.4). Table 2.7.4 reports the approximate convergence exponent α. The asymptotic convergence of the price is roughly quadratic, which is computed from the error values on the two finest grids.

2.7.4 Convergence Using a Rotated Grid

As noted in Section 2.5.3, rotation of the coordinate system and finite difference grid by θ_r in Equation 2.5.4 will result in a monotone spatial discretization. Fully implicit time stepping results in a fully monotone, consistent and stable method, which are required for convergence to the viscosity solution [18]. As well, both the fixed point and penalty iteration are guaranteed to be globally convergent. From a theoretical point of view, this is highly beneficial.

A set of computations were performed with rotated grids, the coarsest of which is shown in Figure 2.7.2. These grids were formed by taking the FD grids of Section 2.7.3 and slightly extending the region where the grid was most highly refined. Since the payoff features are aligned with the axes, this extension was intended to ensure that the region around the strike remained as well covered with fine grid points as with the unrotated grids. The grids were then extended at the boundaries, rotated around $(x_1, x_2) = (\log(100), \log(100))$ by $\theta_r = -26.57$ degrees, then nodes outside of the original Ω_D were removed. The DFT grids used were the same as the unrotated case. The lower boundary condition specified in Equation 2.3.5 was not applied on the rotated grid. Instead, a Dirichlet boundary condition was imposed on all of $\partial\Omega_D$ using the initial conditions. To estimate the effect this had on the solution, the unrotated grid tests were repeated with this new lower boundary condition. The absolute difference in the solution at the strike was less than 10^{-6}. Thus, for this problem, this boundary condition approximation appears to be acceptable for the rotated grid case.

Previous research [113] has shown that, for the pure diffusion case, rotating the co-ordinate systems and grids produces a solution which is less accurate than the unrotated computation for the same grid spacing. In the rotated co-ordinates, initial conditions and barriers cannot be represented exactly. The grid rotation ensured that there was a node at the strike $(S_1, S_2) = (100, 100)$ in common with the original grid, but other nodes did not line up with points of non-smoothness in the payoff.

Comparing Table 2.7.5 with the results in Table 2.7.4 for the conventional grid, the error is slightly larger for the rotated grid at the same mesh size: approximately 1.5 times that of the conventional solution over all grid resolutions. The solutions generated by the conventional grids generated no actual instabilities, so using grid rotation to ensure that the discretization of the diffusion terms yields an M-matrix seems unjustified for this problem in a practical setting.

2.7.5 Convergence of American Options

The demonstrations of convergence with grid refinement were repeated with an American put on the minimum of two assets for both the Normal and MOBED jump distributions. These used the same domain, set of grids and time step sizes as in Section 2.7.3. Crank–Nicolson time stepping was used. The results are given in Table 2.7.6 for nine points, all of which were outside the region where the American minimum value constraint is imposed. In this case the exact solution is not available. Consequently, the error is assumed to be $\epsilon = c\Delta {x_1}^{\alpha}$ with $\Delta x_2, \Delta\tau = \mathrm{O}(\Delta x_1)$ and the convergence exponent was computed by examining the ratio of the difference in the computed option values for three mesh sizes. The α exponent is reported at the same point on the three grids and is roughly two in the MOBED case. In the Normal case the convergence is reported well above two. However, when the experiment was repeated with one further grid refinement, the convergence over the three finest grids for the Normal distribution was between $\alpha = 1.6$ and 1.8.

Figure 2.7.3 shows the results over the core of Ω_C in price scaling for the option using MOBED jumps. The two disconnected, dark regions on the surface represent areas where the

Rotated	**MOBED**	**Absolute**
Grid	Value at $(100, 100)$	**Error** ϵ
1	15.1901	0.4228
2	14.8714	0.1041
3	14.7933	0.0260
Convergence Exponent α,		
Grid 2 to 3		1.8

Table 2.7.5: Numeric solution to the European call on a maximum of two assets with the parameters given in Table 2.7.1. The MOBED model results are reported with the absolute error. The FD grid was rotated by $\theta_r = -26.57$ degrees to guarantee that the spatial finite difference approximation is monotone. Crank–Nicolson time stepping was used. Convergence is asymptotically quadratic, but there is a modest increase in absolute error when compared with Table 2.7.4.

penalty method has imposed the American minimum constraint on the solution, usually called the "early exercise" region. In this region the numerical error is controlled by *Large* of Equation 2.6.3 and is not significant at the grid nodes.

The spatial location of the free boundary between the constrained and unconstrained region is resolved to within the grid spacing. Along the $S_1 = 100$ line the boundary of the American payoff, for the MOBED problem, is at $S_2 = 78.7$ at the coarsest grid resolution with $\Delta x_1 = 0.04$ and $S_2 = 80.3$ for the two finer grids. Along the $S_2 = 100$ line, the boundary is at $S_1 = 81.9$ for all three grids. In both the Normal and MOBED test cases, each refinement of the grid placed the boundary within $(\Delta x_1, \Delta x_2)$ of its location on the coarser grids.

2.7.6 Computational Cost

Table 2.7.3 shows the total number of fixed point and linear solver iterations required for the entire solution of the put on the minimum of two assets in both the European and American cases with Normally distributed jumps. As predicted for the European case in Section 2.6.1, the number of fixed-point iterations required to advance a single time step diminishes with $\Delta\tau$: an average of 4.2 iterations were required with $\Delta\tau = 0.04$ but only 3.2 when $\Delta\tau = 0.01$. For an American option, on average, between 4.3 and 3.4 iterations of the combined fixed point, penalty algorithm were required. For both options, 3.0 linear solver iterations were required to converge to the solution of the equation in Step 5 of Algorithm 2.6.1. The demonstrations have shown that the rapid convergence indicated by the analyses in Section 2.6.1 is achieved in the actual tests, that the penalty iteration incurs only modest additional work and, also, that each resulting linear system is easy to solve.

Grid		**Normal**			**MOBED**		
	S_2	$S_1 = 90$	100	110	90	100	110
1	90	19.5633	16.9359	15.3132	14.7744	12.2635	11.3460
	100	15.8647	12.5443	10.5400	12.0048	7.9551	6.3010
	110	13.5349	9.6916	7.4036	10.9220	5.9661	3.8029
2	90	19.4776	16.8221	15.1976	14.6689	12.1644	11.2836
	100	15.7799	12.4401	10.4135	11.8671	7.8509	6.2074
	110	13.4611	9.5726	7.2579	10.8112	5.8345	3.6758
3	90	19.4619	16.8007	15.1755	14.6437	12.1421	11.2698
	100	15.7648	12.4207	10.3894	11.8350	7.8252	6.1854
	110	13.4488	9.5510	7.2305	10.7861	5.8031	3.6467
Convergence Exponent α	90	2.5	2.4	2.4	2.1	2.1	2.2
	100	2.5	2.4	2.4	2.1	2.0	2.1
	110	2.6	2.5	2.4	2.1	2.1	2.1

Table 2.7.6: Numeric solution to the American put on a minimum of two assets with the parameters given in Table 2.7.1. Crank–Nicolson time stepping was used. The Normal and MOBED jumps model values are reported. Note that convergence is roughly quadratic as the grid and time step is refined.

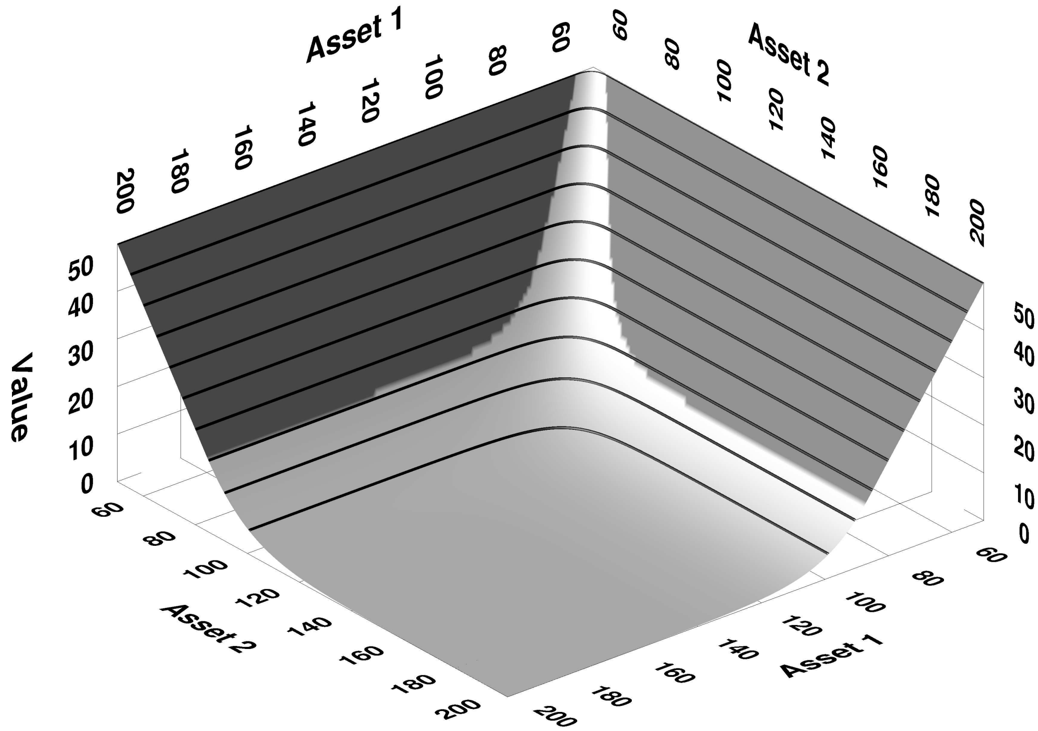

Figure 2.7.3: The solution surface for the value of an American put over the minimum of Assets 1 and 2 with MOBED jumps and parameters from Table 2.7.1. The two, disconnected, darker regions denote areas where the American minimum constraint is imposed.

2.8 Chapter Summary

The governing PIDE for valuing a two-asset, European option is a linear, infinite-domain, initial-value problem. This chapter has presented an approach to localizing this to a finite-domain, initial-boundary value problem. The convergence of the approach was proved rigorously using a Green's function analysis. Numerical tests confirmed that the localization approach is effective.

The discretization of the localized PIDE was presented. It employed finite-differences, a second-order accurate interpolation and numerical quadrature and Crank–Nicolson time stepping. An efficient method for evaluating the integral term was presented, which exploits the algebraic identity relating a correlation to a Fourier transform. The stability of the discrete equations was investigated in both the l_∞ norm by maximum analysis and the l_2 norm by von Neumann analysis. Stability by maximum analysis requires a monotone discretization in space. Three standard methods were reviewed which create a monotone, finite-difference discretization when the coefficients of the diffusion terms are constant. A restriction in $\Delta\tau$ is required for the maximum analysis when the time step method is second order accurate. The von Neumann analysis indicates l_2 norm stability regardless of time step or grid spacing, but this analysis is limited to a periodic, initial value problem on a regular grid.

The fixed point iteration was presented and proved convergent when formulated using a semi-discretization in time. With a monotone spatial discretization, the iteration is convergent in the l_∞ norm by maximum analysis. By von Neumann analysis the iteration is also always convergent in the l_2 norm. The convergence is rapid for typical values of the jump rate parameter λ and time step $\Delta\tau$.

Numerical demonstrations with conventional and rotated grids, for both American and European payoffs, confirmed that the overall method is up to second order convergent with respect to the discretization size in space and time. These tests also demonstrated that the convergence rate of the fixed point iteration method agreed with theory. The demonstrations used market parameters which were synthetic, but had values typical of an equity market.

Chapter 3

Stochastic Volatility with Jumps and Fully Monotone Methods

3.1 Introduction

The numerical methods presented in Chapter 2 are modified in this chapter to value options on a single asset, where the asset price model is a risk-neutral diffusion with finite activity jumps and the variance of the diffusion is stochastic. As noted in the introduction, the variance model studied here is the mean-reverting, diffusion process of Heston [60] with an added jump term. The jumps in the logarithm of the price may have any distribution, but the jumps in variance must be positive and may be correlated with the price jumps. For brevity, the working example is the two-dimensional distribution used by Duffie, Pan and Singleton [41], as well as the one-dimensional Normal and positive exponential distributions which arise when certain of its parameters go to zero.

The particular focus of this chapter will be the calculation of option values when the model parameters are derived from actual financial markets, rather than using synthetic problems as in Chapter 2. In these real-life examples a number of degenerate cases appear, which require the solution method to be adapted. Under these conditions, the three approaches for monotone diffusion discretization listed in Section 2.5.3 for constant coefficient problems (the skewed, rotated and spacing restricted grids) are adapted for stochastic volatility, which has coefficients that change along the independent variable axis of variance. The three approaches are compared for the relative error they incur and their relative computational cost. This comparison informs the approach in Chapter 4, which selects the most efficient of the monotone spatial discretization methods as the basis for the solution method for a non-linear problem.

3.1.1 Previous Work

The semi-analytic, Fourier transform method for European options under affine price processes created by Duffie et al. [41] also applies under stochastic volatility with jumps. Boyarchenko et al. [16] extend this semi-analytic approach for one-touch options. The integral equation approach of Tsavalis et al. [104] or Chiarella et al. [29] can value American options using stochastic volatility models under assumptions about the form of the free boundary.

The two-factor numerical PDE approaches of Zvan et al. [111, 112, 113] were also applied to stochastic volatility alone. These are fully implicit and produce linear to quadratic convergence. In Chapter 2 and in [113], some solutions and results for two-factor monotone methods are reported. Recent work by Ikonen et al. [63, 64] also explores grid spacing restriction to form monotone methods for stochastic volatility and form an operator splitting approach.

3.1.2 Overview of this Chapter

Section 3.2 reviews the governing equations of the price process used in this chapter. Section 3.3 reviews and modifies the localization approach of Section 2.3 for the Heston stochastic volatility model with jumps. Alterations in the stability and monotonicity constraints are addressed in Section 3.4. Section 3.5 demonstrates the approach applied to some fundamental contracts: European and American vanilla options and a one-touch digital option.

3.1.2.1 Synopsis of Contributions in this Chapter

The contributions in this chapter are as follows.

- The extension of two-asset PIDE localization and solution technique of Chapter 2 to Heston stochastic volatility [60] with jumps is detailed in Sections 3.3 and 3.4. In particular, this permits solutions to options with American exercise and also solutions to one-touch digital options without the complexities of the analytic formulation of Boyarchenko et al. [16].

- Numerical demonstrations are provided in Section 3.5.1 of European, American and one-touch digital options demonstrate the efficiency of the approach in the stochastic volatility case. Between 4 and 9 fixed point iterations are required to solve each time step. In the stochastic volatility case, problem parameters are taken from studies of actual markets.

- The three approaches to monotone diffusion discretization of Section 2.5.3 are modified in Section 3.4.3 for Heston stochastic volatility. Numerical evaluations in Section 3.5.2 use only completely monotone discretizations which, although the approach is only linearly convergent, are required for convergence to the viscosity solution. The computational cost and absolute error of each monotone approach is evaluated against the conventional, non-monotone alternative.

3.2 Governing Equations

In this chapter, the objective is to find the theoretical value U of a financial option on an asset with price S. The asset price evolves by a risk-neutral, time-dependent, jump diffusion, while the diffusion variance v evolves by a mean-reverting drift diffusion process also with jumps. The finite activity jump process for the two factors is independent of the diffusion, but jumps in price and variance may be correlated. The option value is given by the solution of a partial integro-differential equation over three independent variables: the time to expiry $\tau = T - t$ and two spatial dimensions, the logarithm of the asset price x and its diffusion variance v where

$$\begin{aligned} \tau &\in [0, T] \\ S &\in [0, \infty] \\ (x, v) = (\log(S), v) &\in [-\infty, \infty] \times [0, \infty] = \Omega_{\overline{\infty}} \ . \end{aligned} \tag{3.2.1}$$

The contract is written at $t = t_0$ when $\tau = T$ remains and the asset price $x_0 = \log(S_0)$ and variance v_0 can be measured from the market. The contract pays $\mathcal{I}(x)$ at expiry time $t = t_0 + T$, when $\tau = 0$. The value of a European option $U(x, v, \tau)$ over $\Omega_{\overline{\infty}} \times [0, T]$ is found by taking expectations under the risk-neutral price process (see Appendix A.2, [18, 33]) and solving

$$\begin{aligned} \frac{\partial U}{\partial \tau} &= \mathcal{L}U + \lambda \mathcal{H}U \\ U(x, v, 0) &= \mathcal{I}(x) \end{aligned} \tag{3.2.2}$$

where, for this chapter,

$$\mathcal{L}U = \left(r - \frac{v}{2}\right)\frac{\partial U}{\partial x} + \kappa_v \left(\theta_v - v\right)\frac{\partial U}{\partial v} + \frac{1}{2}\, v \left[\frac{\partial^2 U}{\partial x^2} + 2\rho_v\sigma_v\frac{\partial^2 U}{\partial x \partial v} + {\sigma_v}^2\frac{\partial^2 U}{\partial v^2}\right] - rU \tag{3.2.3}$$

$$\lambda\mathcal{H}U = \lambda \int_{-\infty}^{\infty}\int_{0}^{\infty} g(J_x, J_v)\left[U\left(x + J_x, v + J_v\right) - U - \left(e^{J_x} - 1\right)\frac{\partial U}{\partial x}\right] dJ_x\, dJ_v \tag{3.2.4}$$

$$g = g(J_x, J_v) \in \mathbb{R}$$

with parameters

$r \geq 0$	the risk free rate of return,
$\sigma_v \geq 0$	the volatility of the asset variance,
$\kappa_v > 0$	the rate of return to
$\theta_v \geq 0$	the reversion variance and
$\lvert\rho_v\rvert \leq 1$	the correlation between the diffusion of x and v.
$\lambda > 0$	is the mean arrival rate of Poisson-distributed jump events and
$J_x, J_v \sim g(J_x, J_v)$	are the jump magnitudes and the PDF by which they are distributed, where
$J_v \geq 0$	is required to ensure that $v \geq 0$ after a jump.

As in Chapter 2, $g(J_x, J_v)$ must satisfy the technical conditions of [54] §II.1.2 Definition 1.6, so that the second term of the integral of Equation 3.2.4 can be written

$$\int_{-\infty}^{\infty}\int_{0}^{\infty} g(J_x, J_v)\, U(x, v, \tau)\; dJ_x\, dJ_v = U(x, v, \tau) \tag{3.2.5}$$

and the third term

$$\int_{-\infty}^{\infty}\int_{0}^{\infty} g(J_x, J_v)\left[\left(e^{J_x} - 1\right)\frac{\partial U}{\partial x}\right] dJ_x\, dJ_v = \kappa_x\frac{\partial U}{\partial x} \tag{3.2.6}$$

where value $|\kappa_x| < \infty$ corrects for the mean drift in price which is due to jumps. Although any $g(x, v)$ satisfying these constraints may be used, this study will use the example from [41] given by

$$g_d(x, v) = \begin{cases} \dfrac{\exp\{-v/\check{\mu}_v\}}{\check{\mu}_v} \cdot \dfrac{1}{\check{\sigma}_x\sqrt{2\pi}} \exp\left\{-\dfrac{1}{2}\left(\dfrac{x - (\check{\mu}_x + v\check{\rho}_j)}{\check{\sigma}_x}\right)^2\right\} & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases} \tag{3.2.7}$$

where $\check{\mu}_x$ and $\check{\sigma}_x \geq 0$ specify the mean and standard deviation of the Normal distribution in x and $\check{\mu}_v \geq 0$ is the mean of the positive exponential distribution in v. Parameter $\check{\rho}_j$ induces correlation, but is not the usual linear correlation so is not bounded in the usual range. If $\check{\mu}_v = 0$ or $\check{\sigma}_x = 0$ then Equation 3.2.7 specifies a one-dimensional jump process. If $\check{\sigma}_x = 0$ and $\check{\rho}_j \neq 0$ then this process is still defined using both x and v dimensions. The PDF is defined for $v < 0$ so that the correlation term in Equation 3.2.4 may be re-arranged to Equation 3.3.7 below.

The value of an American option is, again, the solution to a linear complementarity problem [91, 110] similar to Equations 2.2.10 and 2.2.11

$$\begin{aligned} \frac{\partial U}{\partial \tau} &\geq \mathcal{L}U + \lambda\mathcal{H}U & (3.2.8) \\ U(x, v, \tau) &\geq \mathcal{I}(x) & (3.2.9) \\ U(x, v, 0) &= \mathcal{I}(x) \end{aligned}$$

where at least one of Equations 3.2.8 or 3.2.9 must hold with equality.

The PIDE initial conditions $\mathcal{I}(x)$ are defined by the terminal payoff of the option contract. In this chapter, it is assumed that the payoff has no dependence on v. For an option with strike price K some typical contracts are

$$\mathcal{I}(x) = \begin{cases} \max(\exp(x) - K, 0) = \max(S - K, 0) & \text{vanilla call} \\ \max(K - \exp(x), 0) = \max(K - S, 0) & \text{vanilla put} \\ 0|_{x<\log(K)}\,,\ 1|_{x\geq\log(K)} & \text{digital call and} \\ 0|_{x>\log(K)}\,,\ 1|_{x\leq\log(K)} & \text{digital put.} \end{cases} \tag{3.2.10}$$

Imposing an American condition on a digital put (call) results in a digital "one-touch" option. There is a region below (above) K where the option value is constant as a function of asset price. Since the integral operation of PIDE 3.2.2 is not local this barrier does not result in a domain bounded at $x = \log(K)$, as it would in the pure diffusion case.

3.3 Localization

The solution of PIDE 3.2.2 on the infinite domain is approximated by solving the PIDE on the finite domain

$$\Omega_D = [-X_D, X_D] \times [0, v_D] \tag{3.3.1}$$

as shown in Figure 3.3.1. Again, the integral term in PIDE 3.2.2 must be truncated to Ω_D, which is equivalent to assuming a zero value outside the domain and introduces an error. As in the two-asset case, the full PIDE 3.2.2 is only solved on a smaller domain

$$\Omega_C = [-X_C, X_C] \times [0, v_C] \ ; \quad 0 < X_C < X_D, \ 0 < v_C < v_D \tag{3.3.2}$$

and an approximation computed over the remainder of Ω_D so as to limit the error in the truncated correlation integral term. Note that in this case a portion of the $v = 0$ boundary is considered to be inside Ω_C.

Similarly to the two-asset case, observe that a sufficiently large Ω_D would have, letting $V(x, v, \tau)$ denote the solution to the localized equations,

$$V \simeq a(\tau)S + b(\tau); \quad S = e^x \tag{3.3.3}$$

over large portions of the solution because of the independence of the payoff from v. If we write PIDE 3.2.2 in terms of S then it may be deduced that

- when $S = 0$ ($x \to -\infty$) then $V = b_1(\tau)$,
- as $S \to \infty$ ($x \to \infty$) then $V = a_2(\tau)S + b_2(\tau)$ and
- as $v \to \infty$, which is the coefficient of asset price diffusion in PIDE 3.2.2, then $V \simeq a_3(\tau)S + b_3(\tau)$. This also implies that $\partial V/\partial v \to 0$ as $v \to \infty$.

In these limits, $\mathcal{H}U \simeq 0$ similarly to the two-asset case in Section 2.3.2. Thus over $\Omega_D \setminus \Omega_C$ (the portion of the solution outside Ω_C) the numerical technique which follows will again solve the portion of PIDE 3.2.2 which remains after setting $\lambda = 0$.

Rearranging PIDE 3.2.2 and applying Equations 3.2.5 and 3.2.6, the option value $V(x, v, \tau)$ over the localized domain $\Omega_D \times [0, T]$ is given by solving

$$\begin{aligned} \frac{\partial V}{\partial \tau} &= \mathcal{G}V - (r + \lambda_C)V + \lambda_C \mathcal{J}V \\ V(x, v, 0) &= \mathcal{I}(x) \\ V(x, v, \tau) &= B(x, v, \tau) \quad ; \quad (x, v) \in \partial\Omega_D \end{aligned} \tag{3.3.4}$$

where, for this chapter,

$$\mathcal{G}V = \left(r - \frac{v}{2} - \lambda_C \kappa_x\right)\frac{\partial V}{\partial x} + \kappa_v \left(\theta_v - v\right)\frac{\partial V}{\partial v} + \frac{1}{2}\, v \left[\frac{\partial^2 V}{\partial x^2} + 2\rho_v \sigma_v \frac{\partial^2 V}{\partial x \partial v} + {\sigma_v}^2 \frac{\partial^2 V}{\partial v^2}\right] \tag{3.3.5}$$

$$\lambda_C = \begin{cases} \lambda & \text{for } (x, v) \in \Omega_C \\ 0 & \text{for } (x, v) \in \Omega_D \setminus \Omega_C \quad \text{(the region outside } \Omega_C\text{) and} \end{cases} \tag{3.3.6}$$

$$\mathcal{J}V = \iint_{(J_x, J_v) \in \Omega_D} g(x - J_x, v - J_v)\ V(x, v)\ dJ_x\, dJ_v \quad . \tag{3.3.7}$$

Operator $\mathcal{G}$ contains all of the differential terms of Equations 3.2.2 and 3.2.6. $\mathcal{J}$ is the correlation integral term of Equation 3.2.2 truncated on Ω_D, arranged so that the integration bounds are

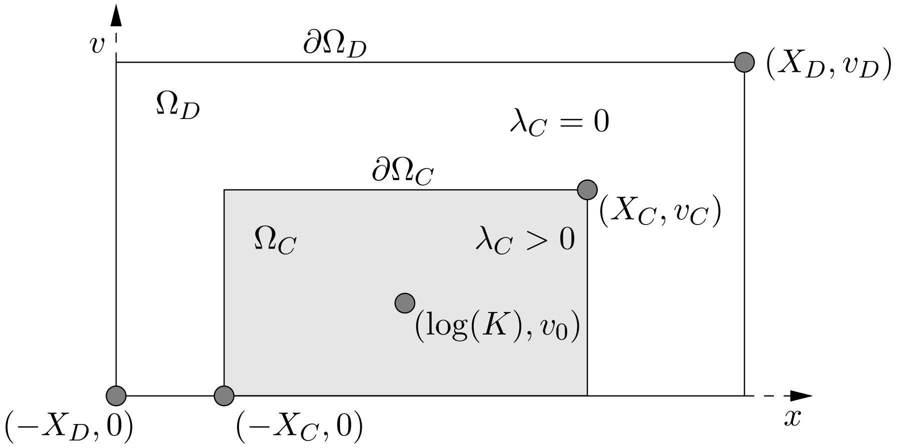

Figure 3.3.1: Domains $\Omega_C \subset \Omega_D$ with boundaries $\partial\Omega_C$ and $\partial\Omega_D$ truncate the infinite domain. Over Ω_C, $\lambda_C > 0$ so that the PIDE 3.2.2 is computed with the jump component in Ω_C only. Ω_D has a lower bound intersecting the x axis at a finite point on the price axis in log-scaling, thus above the zero axis of the unscaled asset price S. Although this figure shows the domains with symmetric x bounds $0 < X_C < X_D$ this is for convenience only: in general, the upper and lower boundaries in x need not be symmetric.

$(J_x, J_v) \in \Omega_D$, the form most like the final numerical method. Note that this form is why $g_d(x, v < 0) = 0$ in Equation 3.2.7.

At $v = 0$ Equation 3.3.4 is solved; no additional boundary condition is required because the characteristics in the v direction are outgoing. The jump term remains on the portion of $v = 0$ inside Ω_C, which is an additional complexity compared to the two-asset problem of Chapter 2. On the $v = v_D$ boundary $\lambda_C = 0$ and the condition $\partial V/\partial v = 0$ is set as in [49, 111]. Thus, on the v boundaries

$$\frac{\partial V}{\partial \tau} = \begin{cases} (r - \lambda_C \kappa_x)\dfrac{\partial V}{\partial x} + \kappa_v \theta_v \dfrac{\partial V}{\partial v} - (r + \lambda_C)V + \lambda_C \mathcal{J}_D V & \text{on } v = 0,\ x \in (-X_D, X_D) \\ \left(r - \dfrac{v}{2}\right)\dfrac{\partial V}{\partial x} + \dfrac{1}{2}\, v\dfrac{\partial^2 V}{\partial x^2} - rV & \text{on } v = v_D,\ x \in (-X_D, X_D)\ . \end{cases} \tag{3.3.8}$$

As noted at the start of this section $V(x \to -\infty, v, \tau) = b_1(\tau)$ so the lower boundary in x is set to

$$\frac{\partial V}{\partial \tau} = -rV \quad \text{on } x = -X_D,\ v = [0, v_D] \tag{3.3.9}$$

and from Equation 3.3.3 the upper boundary in x is set to

$$V(x, v, \tau) = a_0 e^x + b_0 e^{-r\tau} \quad \text{on } x = X_D,\ v = [0, v_D] \tag{3.3.10}$$

where a_0, b_0 are determined by the contract payoff.

Note that in the case of a digital one-touch option, the extended domain is still used to ensure that the integral term is correctly computed into the region beyond the payoff at $x = \log(K)$, even though the option value there is simply constant, as if it were a Dirichlet boundary condition.

Theorem 2.3.7 and 2.3.8 which limit localization error also apply to stochastic volatility problems. Minor modifications to Assumptions 2.3.1 and 2.3.2 are required, reflecting the fact that the payoff is assumed to be bounded in v by a constant value.

Assumption 3.3.1 *The initial condition (the option payoff) $\mathcal{I}(x)$ can be bounded by*

$$\mathcal{I}(x) \leq c_1 + c_2 e^x$$

for some constants c_1 and c_2 and the jump distribution must be such that $|\mathcal{H}\mathcal{I}| < \infty$ for $|x|, v < \infty$.

Assumption 3.3.2 *The solution $U(x, v, \tau)$ to Equation 3.2.2 satisfies the condition*

$$|\mathcal{H}U| \leq c_3 + c_4 e^x \quad \textit{for} \quad (x, v) \in \Omega_{\overline{\infty}} \setminus \Omega_C \tag{3.3.11}$$

for constants c_3 and c_4.

3.3.1 Sizing Ω_D and Ω_C for Error Control

Domain sizing is done nearly identically to Section 2.3.3. Under stochastic volatility, the width of the marginal distributions of an asset path at expiry has a strong dependence on v. Also, no negative jumps in v are permitted in this case. Let (x_p, v_p) be a point from a region of particular interest, typically a small region around the $(\log(K), v_0)$ point. Equation 2.3.12 for the size of Ω_C is replaced by the following computation, where minimum widths Δx_C and Δv_C are computed based on a tolerance parameter ϵ_c and marginal distributions f_x and f_v of the price process such that

$$\begin{aligned} &\int_{-\Delta x_C}^{\Delta x_C} f_x(x_p + z; x_p, v_p, T)\, dz > (1 - \epsilon_c) \\ \text{and} \quad &\int_{0}^{\Delta v_C} f_v(v_p + z; x_p, v_p, T)\, dz > \left(1 - \frac{\epsilon_c}{2}\right) . \end{aligned} \tag{3.3.12}$$

The size of Ω_C is chosen to ensure $(x_p \pm \Delta x_C, v_p + \Delta v_p) \in \Omega_C \;\forall\; (x_p, v_p)$.

Distances Δx_D and Δv_D between $\partial\Omega_C$ and the boundary $\partial\Omega_D$ are computed using tolerance ϵ_d in the same fashion as Equation 2.3.13. Both $\partial\Omega_C$ and $\partial\Omega_D$ run along $v = 0$ and no gap is required at this domain boundary. In the cases where the marginal distribution of the jumps is zero in a given direction for any of the other boundaries, the gap between $\partial\Omega_C$ and $\partial\Omega_D$ is not permitted to go to zero so that the boundary conditions of the problem may remain unchanged from Equation 3.3.8.

3.4 Discretization

The construction of a finite difference grid, a discrete Fourier transform grid, the discrete differential operator G and integral operator J follows the approach of the two-asset case as detailed in Section 2.4. A finite difference grid is formed of N points in space and N_τ points in time

$$(x, v)_{\mathsf{i}} \in (-X_D, X_D) \times [0, v_D] \ , \qquad \mathsf{i} = 1 \dots N \tag{3.4.1}$$

$$\tau_n \in [0, T] \ , \qquad n = 0 \dots N_\tau \ , \quad \tau_n = n\Delta\tau \ . \tag{3.4.2}$$

Once again, the discrete linear form of the operators in Equation 3.3.4 is

$$\mathsf{T} = -\left[\mathsf{G} + \lambda \mathsf{I_c}(\mathsf{J} - \mathsf{I}) - r\,\mathsf{I}\right] \tag{3.4.3}$$

with the operators

$$\mathcal{G}V \simeq \mathsf{G}\mathsf{v}$$
$$\lambda_C \mathcal{J} V \simeq \mathsf{I_c}\mathsf{J}\mathsf{v}$$

and the solution vector

$$V(\tau_n) \simeq \mathsf{v}^n \ .$$

A time step is advanced using the theta method time discretization, written as

$$\left[\mathsf{I} + (1-\theta)\Delta\tau\ \mathsf{T}\right]\mathsf{v}^{n+1} = \left[\mathsf{I} - \theta\Delta\tau\ \mathsf{T}\right]\mathsf{v}^n + \mathsf{b} \tag{3.4.4}$$

where vector b enforces Dirichlet boundary conditions. Again, G is formed by first and second order finite differences and J using second order quadrature, with Jv computed by interpolation and an FFT as in Section 2.4.3.3. Note that the $v = v_D$ upper boundary is an independent, one-dimensional PDE and could be computed and imposed as a Dirichlet boundary condition rather than treated as part of the solution vector. The lower $v = 0$ boundary is a PIDE and must be part of the solution vector.

3.4.1 Time Step Solution

The fixed point iteration described in Section 2.6.1 and the penalty iteration of [37, 48, 111] described in Section 2.6.2, combined in Algorithm 2.6.1, are used to solve each discrete time step.

Compared to the two-asset case, there is a small difference in the approach to solving the linear systems which result. The linear system from iteration Equation 2.6.2 is again solved using the iterative, conjugate-gradient style solver Bi-CGStab with an incomplete LU preconditioner ILU(2), retaining 2 levels of fill. However, where the grid is aligned with the axes (i.e. not rotated or skewed), the ILU preconditioner is ordered following the grid in the v direction first, then the x. This follows a well established rule of thumb for anisotropic, diffusion-dominated systems [32] and slightly improves performance in these cases. For rotated or skewed grids, RCM re-ordering is used [98].

3.4.2 Rotation and Alignment of the Jump PDF

In Section 2.4.3.4 it was noted that the jump PDF could be shifted on the DFT grid to improve the absolute error of the computation. For stochastic volatility problems with a shift (z_x, z_v), so that $g'(x - z_x, v - z_v) = g(x, v)$ where $g'(x, v < 0) = 0$, attention must be paid to the bounds of integration in v so

$$\begin{aligned}\mathcal{J}U(x,v) &= \int_{-\infty}^{\infty}\int_{0}^{\infty} g(J_x, J_v)\, U(x + J_x, v + J_v)\, dJ_x\, dJ_v \\ &= \int_{-\infty}^{\infty}\int_{0}^{\infty} g'(J_x - z_x, J_v - z_v)\, U(x + J_x, v + J_v)\, dJ_x\, dJ_v \\ &= \int_{-\infty}^{\infty}\int_{-z_v}^{\infty} g'(J'_x, J'_v)\, U(x + J'_x + z_x, v + J'_v + z_v)\, dJ'_x\, dJ'_v \\ &= \mathcal{J}'U(x + z_x, v + z_v)\ .\end{aligned} \tag{3.4.5}$$

The results using $\mathcal{J}'U$ with g' can be used as results for $\mathcal{J}U$ by taking the correlation result at $(x+z_x, v+z_v)$. As in the two-asset case, this observation may be used to align discontinuities in a jump PDF to fall between DFT grid nodes to simplify integration. In the stochastic volatility case, it can be used to align a 1D PDF to fall exactly on the nodes of the 2-dimensional DFT grid.

Two degenerate cases of the jump PDF of Equation 3.2.7 are of interest. Where $\breve{\sigma}_x = 0$ and $\breve{\rho}_j \neq 0$ the DFT grid should be rotated by

$$\theta_f = \tan^{-1}\left(-\breve{\rho}_j\right) \tag{3.4.6}$$

and translated to place the PDF exactly on DFT grid nodes. The DFT grid must also be extended to completely overlap Ω_D. Where $\breve{\mu}_v = 0$ the DFT grid must be translated, but not rotated, to accurately represent the PDF. These rotations and translations change the interpolation matrices K and L of Equation 2.4.20, but do not significantly alter the overall approach.

3.4.3 Convergence by using a Stable, Monotone Approach

The preconditions which must hold for Equation 3.4.4 to be a stable method in the l_∞ norm, which also imply convergence to the viscosity solution of Equation 3.3.4, are analagous to those outlined for two-asset problems in Section 2.5. As in the two-asset case, these reduce to the requirement that T of Equation 3.4.3 be an M-matrix. The von Neumann analysis in the l_2 norm does not apply, since the coefficients of the stochastic volatility problem are not constant.

The conditions for the monotonicity of the finite difference approximation, given in Section 2.5.3, must be adapted when they are used for stochastic volatility problems. Note that $\sigma_v = 0$ is also allowed, in which case there are no special conditions required on the diffusion terms.

1. The drift term weighting approach “central weighting as much as possible” is also applied to stochastic volatility problems. Note that as $v \to 0$, as well as on the $v = 0$ boundary, a first order approach must be used.

2. The skewed co-ordinate system of Hull and White [62, 112] becomes

$$(\psi_1, \psi_2) = (\sigma_v x + v, \sigma_v x - v) \tag{3.4.7}$$

which will result in a zero diffusion correlation in the transformed system. See Appendix E.2 for details of the transformed partial differential terms.

3. The grid spacing condition under which G has positive, off-diagonal coefficients is, for stochastic volatility

$$|\rho_v \sigma_v| \le \frac{\Delta v}{\Delta x} \le \left| \frac{1}{\rho_v} \sigma_v \right| \ . \tag{3.4.8}$$

4. The grid rotation approach also works for stochastic volatility, where the rotation is given by

$$\theta_r = \frac{1}{2} \tan^{-1} \left(\frac{2\rho_v \sigma_v}{1 - {\sigma_v}^2} \right) \ . \tag{3.4.9}$$

See Appendix E.3 for details of the transformed partial differential terms.

As in the two-asset case, these restrictions can often be violated when solving linear problems without incurring any practical, numerical problems, particularly for those restrictions involving the discrete diffusion coefficients.

The above schemes 2 to 4, which enforce monotone discrete diffusion, come with drawbacks. Using the grid skew of Hull and White allows the terminal or barrier conditions of simple options to be represented exactly. However, the transformed axes diminish the ability to concentrate points in a specific region of interest in (x, v) co-ordinates.

Although the grid spacing of Equation 3.4.8 is easily satisfied, to do so may not always be computationally efficient, particularly for the differential portion of the problem computed over the extended region $\Omega_D \setminus \Omega_C$, where a high-accuracy solution is not required. Equation 3.4.8 is more restrictive as $\rho_v \to \pm 1$.

The grid rotation approach was examined in [113] and Chapter 2 for two-asset problems. In both cases the rotation was shown to introduce some numerical error, apparently due to no longer having an exact representation of the option payoff or barrier conditions. However, the rate of convergence when grid and time step were refined was still as high as quadratic in numerical tests.

3.4.3.1 Boundary Considerations

At the boundary $v = 0$, Equation 3.3.8 supplies an outflow boundary condition where first order differencing of the first partial differential term in v guarantees an M-matrix. The drift terms in x must be forward or backward weighted at $v = 0$ because there is no diffusion, thus in x and v the discretization is always first order. At the $v = v_D$ boundary, well away from Ω_C, the drift term in v has been set to zero, hence there is no difficulty at this boundary.

Under grid rotation or skew there is no guarantee that the $v = 0$ boundary will be an outflow boundary along the transformed axes. In these cases the boundaries will be represented with lines truncating the transformed grid along $v = 0$ and $v = v_D$. An irregularly spaced, finite difference approximation will be formulated along those lines using the points where the rotated or skew grid intersects the boundary. The interior points required to form the component of the drift term normal to the boundary will be formed by second order, positively weighted, linear interpolation. The accuracy of this approach is first order overall, which matches the order of the forward or backward approximation required for the drift terms on $v = 0$ boundary points. Dirichlet conditions are imposed at $x = \pm X_D$, simplifying the discretization on these boundaries when lines $x = \pm X_D$ are used to truncate the grid.

3.5 Numerical Demonstrations

The solution approach from Chapter 2, adapted for stochastic volatility in Sections 3.3 and 3.4, is applicable to any contract with a payoff bounded linearly in asset price and to any variations in asset price model that satisfy the conditions in Sections 2.3, 3.2 and 3.3. This includes jumps in one dimension and multiple finite activity jump processes.

As a practical demonstration, five market cases were selected which use the stochastic volatility with jumps model defined in PIDE 3.2.2. In all cases the single jump distribution is distributed as Normal in log-price x and positive exponential in v using the PDF $g_d(J_x, J_v)$ given in Equation 3.2.7. The cases were taken from [24, 41, 100] and are summarized in Table 3.5.1. Cases A and B from [24] were derived from the statistical analysis of a large data set of option values. Cases C, D and E were derived from fitting the model parameters to an implied volatility surface observed in a market, which is a relatively small data set. These last three cases contain at least one extreme parameter value each: $\rho_v = -1$ and $\kappa_v = 9.70$ in case C, a degenerate $\breve{\sigma}_x = 0.0$ in Case D (approximating $\breve{\sigma}_x = 0.0001$ in [41]), and $\breve{\mu}_v = 0.0, \breve{\rho}_j = 0$ in Case E. Cases A and B are well behaved and typical. Cases C, D and E will illustrate the robustness of the numerical approach under extreme parameter values.

All market cases were used to value options by employing both Crank–Nicolson and fully implicit time stepping, with central weighting as much as possible and using both monotone and non-monotone spatial discretizations. A selection of results are presented for each contract valued: a European put and call, an American put and a digital one-touch option. Each option is computed with exercise price $K = 100$, a risk-free rate $r = 0.05$ and expiry time $T = 1$.

The error tolerance parameters used for these demonstrations are

- $\epsilon_c = 5 \times 10^{-4}$, which the inner domain Ω_C size must satisfy (Equation 3.3.12),
- $\epsilon_d = 10^{-4}$, which the grid region Ω_D spacing from Ω_C must satisfy (Equation 2.3.13),
- $\epsilon_u = 10^{-7}$ for a relative update to a time-step solution by fixed point or combined fixed point, penalty iteration (Section 2.6.1) and
- $\epsilon_l = 10^{-8}$ for a relative update by Bi-CGStab to a linear system solution (Section 2.6.3).

Monotone discretization methods require an FD grid to be customized for each market case so, in addition, the domain size dictated by tolerances ϵ_c and ϵ_d were computed for each market

Market Case	$\sigma_0 = \sqrt{v_0}$	κ_v	θ_v	σ_v	ρ_v	λ	$\check{\mu}_x$	$\check{\sigma}_x$	$\check{\mu}_v$	$\check{\rho}_j$	See
A	0.1000	5.06	0.060	0.61	-0.10	1.64	-0.03	0.2200	0.0036	-7.87	[24]
B	0.1000	4.08	0.050	0.57	-0.21	1.20	-0.04	0.1600	0.0049	-9.14	[24]
C	0.1254	9.70	0.011	0.38	-1.00	1.16	-0.10	0.1801	0.0696	-0.06	[100]
D	0.0870	3.46	0.008	0.14	-0.82	0.47	-0.10	-	0.0500	-0.38	[41]
E	0.0940	3.99	0.014	0.27	-0.79	0.11	-0.14	0.1500	-	-	[41]

Table 3.5.1: The model parameters for five real market cases from four papers are used to demonstrate the numerical approach. In all cases the coefficients were measured from option values observed in the market and are risk-adjusted. In the models A and B from [24] no σ_0 value was given, so an arbitrary value was selected. Other references [20, 43, 44] report parameters in similar ranges. Risk free rate $r = 0.05$ for all demonstrations, with strike $K = 100$ and expiry time $T = 1$.

case and used to specify the grid sizes. The region of particular interest, around which Ω_C was defined using ϵ_c, was specified from $(S, v) = (80, 0.0)$ to $(125, 0.2)$ for options with a vanilla payoff and from $(S, v) = (70, 0.0)$ to $(100, 0.2)$ for digital one-touch options. Table 3.5.2 lists the resulting Ω_C and Ω_D sizes for each market case, for vanilla payoff options. Where the jump distribution dictated a zero distance between Ω_C and Ω_D on any boundary other than $v = 0$, the domain was extended to accommodate one extra coarse grid FD node so that the boundary condition equations did not need to change. The size of the FD and DFT grid in (x, v) was set to fully cover the Ω_D specific to each market case. For the DFT grid, this could be done only as far as the spacing rules for each grid and the rules for setting node counts N_1 and N_2 permitted (see Section 2.7.1).

The values of ϵ_c and ϵ_d are such that the option prices do not change, at the accuracy reported, when tolerances ϵ_c and ϵ_d are tightened (which is equivalent to enlarging Ω_C, Ω_D). This was checked by re-running a number of cases with grids which had a size to match $\epsilon_c = 1 \times 10^{-4}$ and $\epsilon_d = 5 \times 10^{-5}$. Similarly, tighter tolerances for ϵ_u and ϵ_l do not change the solutions within the accuracy reported.

The numerical demonstrations in this chapter use four sets of finite difference grids. One is conventional, and the remaining three sets each test the merits of each the three monotone discrete diffusion approaches of Section 3.4.3.

- Plain FD grids are formed with lines parallel to the x and v axes, but without attempting to satisfy the monotonicity restriction of Equation 3.4.8. An example is given in Figure 3.5.1 and a detailed description in Section 3.5.1.

- Skew grids apply the grid skew of Hull and White, Equation 3.4.7, to guarantee monotonicity in the FD discretization. Figure 3.5.2 shows the coarsest Skew grid for market case B. Details are given in Section 3.5.2.1.

- PosD grids apply Equation 3.4.8 and restrict grid spacing. Figure 3.5.3 displays the coarsest PosD grid for market case B. Details of the construction are in Section 3.5.2.2

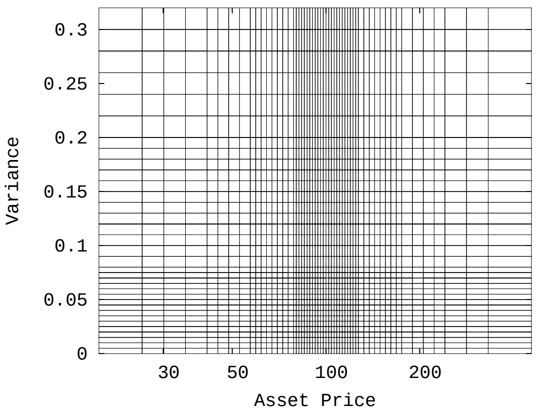

Figure 3.5.1: The coarsest finite difference grids of the Plain set for vanilla payoff options for market case E. The other three grid types attempt to match the spacing which the Plain grids have along x near the $v = v_0$ line. The grid range is given in Table 3.5.2.

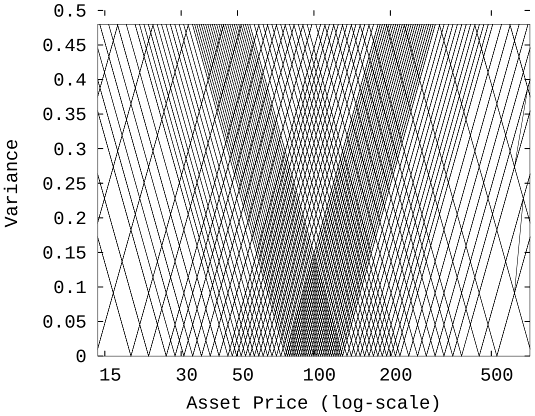

Figure 3.5.2: The coarsest finite difference grid of the Skew set for market case B. The grid range is given in Table 3.5.2.

- Rot grids apply the rotation given by Equation 3.4.9 to ensure monotonicity. Figure 3.5.4 shows the coarsest Rot grid for the digital one-touch contract for market case D. See Section 3.5.2.3 for details and results.

Two to three refinements of the coarsest grid were performed within each set. Each refinement of a grid has twice the resolution in log-price x, variance v and time τ and thus eight times the number of solution points.

3.5.1 Results Using the Non-monotone Approach

The coarsest grid of the Plain set has a spacing of $(\Delta x, \Delta v) = (0.02, 0.005)$ near the $(\log(K), v_0)$ node for vanilla European and American options (as shown in Figure 3.5.1) and $(\Delta x, \Delta v) = (0.01, 0.005)$ for one-touch options. The region of high resolution in v is only near v_0; at $v = 0.08$ the spacing widens to $\Delta v = 0.01$. The grid widens to up to $(\Delta x, \Delta v) = (0.32, 0.04)$ near the outer boundaries. The Plain grid for one-touch options is conventional outside of the payoff, inside the payoff region it is extended by a single node in x to cover the remainder of Ω_D.

The DFT grid spacing for these problems was set, at the coarsest level, to $(\Delta x_f, \Delta v_f) = (0.04, 0.02)$ for options with vanilla payoffs and $(\Delta x_f, \Delta v_f) = (0.02, 0.01)$ for the one-touch options. For market cases A and B, $(\Delta x_f, \Delta v_f) = (0.02, 0.005)$ for one-touch options; the exponential jumps have a small mean $\breve{\mu}_v$ which led to erratic convergence when the PDF was

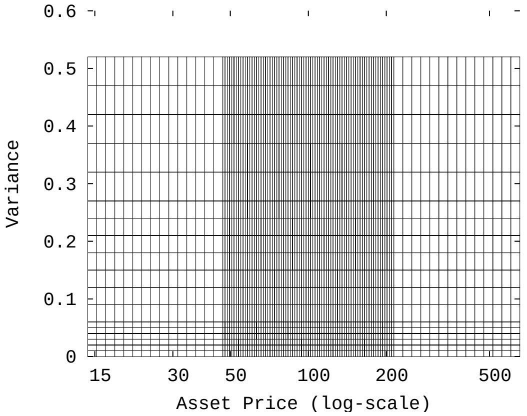

Figure 3.5.3: The PosD finite difference grid set has stringent restrictions on node spacing. This is the coarsest grid for market case B. The grid range is given in Table 3.5.2.

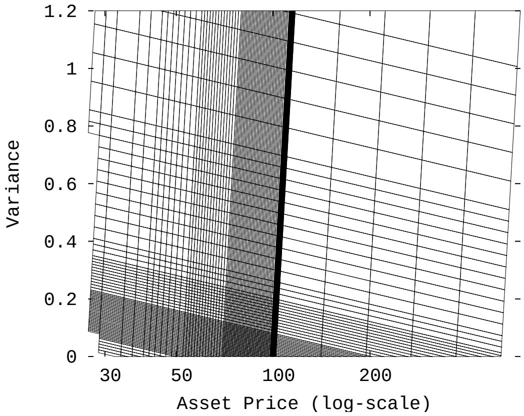

Figure 3.5.4: This is the coarsest grid of the Rot grid set for market case D for the one-touch options. The nodes in the payoff zone are shown to illustrate the structure of the grid. The exact grid range is not given in Table 3.5.2 for one-touch problems.

			Plain	Plain	
Market			FD Grid	DFT Grid	
Case	Ω_C Size (S, v)	Ω_D Size (S, v)	Nodes	Nodes	Rotation
A	(17.7, 0)→(565, 0.43)	(5.4, 0)→(1346, 0.50)	3330	140×28	
B	(27.9, 0)→(358, 0.43)	(14.1, 0)→(710, 0.50)	2553	96×28	
C	(22.2, 0)→(450, 1.04)	(9.8, 0)→(867, 1.80)	4602	112×90	
D	(45.6, 0)→(219, 0.64)	(26.7, 0)→(232, 1.22)	2303	56×90	20.81
E	(42.0, 0)→(238, 0.29)	(18.6, 0)→(457, 0.32)	1836	80×16	

Table 3.5.2: This table lists, for each market case in Table 3.5.1, the Ω_D, Ω_C size required by ϵ_c and ϵ_d. The number of non-Dirichlet FD nodes and DFT nodes are reported for the coarsest Plain grid case. Finally the DFT grid rotation θ_f of Equation 3.4.6 required to accurately capture the one-dimensional jump PDF is given in degrees. This is non-zero only for Market case D.

poorly resolved in the v axis. Table 3.5.2 reports the Plain FD grid node count and DFT grid node counts for the coarsest grid used in each market case for vanilla European and American options.

The time step used with the coarsest grid was $\Delta\tau = 1/12 \simeq 0.0833$ for vanilla European and American options and $\Delta\tau = 1/24 \simeq 0.04167$ for the one-touch options. The time step and grid spacing was set to yield an error of the order of 1% at the coarsest grid level.

The DFT grid was aligned so that its nodes and edges match FD grid nodes near $(\log(K), v_0)$. If this grid alignment is not done, then every evaluation of $\mathsf{K} \cdot \mathsf{F} \cdot \mathsf{L}$ in Equation 2.4.20 incurs significantly more interpolation error. Numerical trials indicated that it is better to reduce the interpolation error in each computation of $\mathsf{K} \cdot \mathsf{F} \cdot \mathsf{L}$ and then interpolate a final value from the grid than to have misaligned grids. Indeed, interpolation error from misaligned grids can grow in absolute value until it dominates the other sources of error.

A good fit and alignment is not always possible: market case D requires the DFT grid to be rotated by $\theta_f = 20.81$ degrees (Table 3.5.2) so as to accurately capture the jump PDF, which is one-dimensional and not aligned with the (x, v) axes. The grid was then enlarged by 1.21 and 1.60 times in x and v respectively to cover Ω_D. Without DFT grid rotation and alignment, the numerical values for market case D on the first three grid resolutions were so poor as to be unusable.

3.5.1.1 Convergence of the Solution for European Options

The first column of Table 3.5.3 reports the option values and the rate of convergence of the numerical solution to a European call for each market case from Table 3.5.1 using the Plain grid set. Central drift term weighting was used as much as possible with Crank–Nicolson time stepping. The exact solutions reported were computed by the semi-analytic approach in [78]. The convergence rate is the exponent α in $\mathrm{O}(\Delta x^\alpha, \Delta v^\alpha, \Delta\tau^\alpha)$ with respect to the grid spacing and time step, computed from grid resolutions 3 and 4 by assuming the error $\epsilon = c\Delta x^{\alpha}$ and $\Delta v, \Delta\tau = \mathrm{O}(\Delta x)$ (see Appendix E.4).

The solution in this case is to a linear problem where the convergence to the classical solution is likely. The convergence is roughly quadratic in all cases. Case D, which requires a rotated DFT grid, had somewhat erratic convergence but a near-zero error at the finest grid, so the convergence rate was computed from grid resolutions 2 and 3. Since there are a number of distinct refinement processes in play (FD and DFT grid, interpolation and the τ grid) and some nodes of the problem which are computed to first order accuracy, completely smooth convergence cannot be expected.

The second column of Table 3.5.3 reports the results of a numerical computation of a European put option. In this case, implicit time stepping is used so that only the discrete diffusion component of G in Equation 3.4.3 does not satisfy the M-matrix compatibility requirements discussed above in Section 3.4.3. With implicit time stepping, the rate of convergence is generally reduced and the error due to the first order convergence in $\Delta\tau$ appears to dominate starting with the second refinement.

For each time step of the European options, an average of between 4 and 9 fixed point iterations were required on the coarsest grids, where $\Delta\tau = 1/12$, and 3 to 6 on the finest grids. A higher number of iterations was required consistently for market cases A and B where λ was higher. Market cases D and E, which have the smallest λ values, required the lowest number of iterations. This is consistent with the prediction of Theorem 2.6.2 for update tolerance $\epsilon_u = 10^{-7}$. An average of 6 to 9 preconditioned Bi-CGStab iterations were required on the coarsest grid to solve the linear system from the fixed-point iteration, which increased to 8 to 39 iterations on the finest grid. This indicates that the fixed-point approach generates a linear system which is inexpensive to solve, but which is less well conditioned than the ones arising from the two-asset case.

3.5.1.2 Convergence of the Solution to American and One-Touch Options

The third column of Table 3.5.3 reports the numerical solution for an American put option over the five market cases using Crank–Nicolson time stepping. The Plain grids and time steps were identical to the European case. This generated a roughly quadratic convergence rate with respect to grid spacing and time step in most cases. The convergence rate α of the error is assumed to converge as for European options however, since no exact solution is available, it is measured using the difference between the solutions on the three finest grids.

The only change in numerical approach for American options is the combination of the fixed point iteration with a penalty iteration. This enforces the free boundary that appears with the early exercise condition Equation 3.2.9. The number of combined iterations required for each time step was comparable to the number of plain fixed point iterations required for a European put solution: on average between 4 to 9 per time step for the coarsest grids and 4 to 7 on the finest. The linear systems were also solved with similar cost, between 6 and 9 iterations of preconditioned Bi-CGStab on the coarsest grid and 6 to 28 iterations on the finest.

Figure 3.5.5 shows the solution to the American option for market case A using central drift term differencing where possible and implicit time stepping. Figure 3.5.6 shows the result

			European Call	European Put	American Put	One-touch "Up-and-in"	
			Plain	Plain	Plain	Plain	Rot
Market Case	σ_0	Grid	C-N $S = 100$	Implicit $S = 100$	C-N $S = 100$	Implicit $S = 90$	Implicit $S = 90$
A	0.1000	1	16.3935	11.2659	11.9219	68.1661	68.2071
		2	16.2659	11.2993	11.8568	68.4474	68.4670
		3	16.2582	11.3290	11.8385	68.5796	68.5979
		4	16.2508	11.3480	11.8332	68.6660	68.6650
		Exact	16.2479	11.3708	-	-	-
		α	1.9	0.9	1.8	0.6	1.0
B	0.1000	1	13.4675	8.4045	9.0892	67.7794	67.7961
		2	13.3410	8.4206	9.0334	68.1154	68.1209
		3	13.3585	8.4421	9.0194	68.2829	68.2901
		4	13.3524	8.4563	9.0156	68.3696	68.3712
		Exact	13.3501	8.4730	-	-	-
		α	1.9	0.9	1.9	1.0	1.0
C	0.1254	1	12.3332	7.2729	8.0808	63.6539	63.4675
		2	12.2743	7.3213	8.0670	64.0555	64.0536
		3	12.2619	7.3500	8.0646	64.2907	64.2773
		4	12.2591	7.3654	8.0642	64.3954	64.3899
		Exact	12.2584	7.3813	-	-	-
		α	2.2	1.0	2.8	1.2	1.0
D	0.0870	1	8.2104	3.2317	3.7674	51.3434	51.1531
		2	8.2085	3.3024	3.8188	52.0396	52.0448
		3	8.2120	3.3208	3.8234	52.3652	52.3746
		4	8.2121	3.3279	3.8234	52.5326	52.5309
		Exact	8.2121	3.3350	-	-	-
		α	6.0	1.0	6.7	1.0	1.0
E	0.0940	1	7.9331	2.9903	3.5449	49.1279	49.0291
		2	7.9312	3.0244	3.5555	49.9408	49.8774
		3	7.9317	3.0403	3.5589	50.2934	50.2582
		4	7.9320	3.0479	3.5598	50.4656	50.4558
		Exact	7.9321	3.0551	-	-	-
		α	1.6	1.0	1.9	1.0	1.0

Table 3.5.3: Option values at $(S, v) = (100, \sigma_0^2)$ for the European call and put, American put, and $(S, v) = (90, \sigma_0^2)$ for the one-touch contract are listed. The first four columns are results using the Plain grid, the final uses a Rot grid. Central drift term weighting was used where possible, Crank–Nicholson or implicit time stepping was used as noted. Each refinement halves the values of Δx, Δv and $\Delta\tau$. An exact solution [41] is given where it exists. The convergence rate given is the exponent α in $\mathrm{O}(\Delta x^\alpha, \Delta v^\alpha, \Delta\tau^\alpha)$.

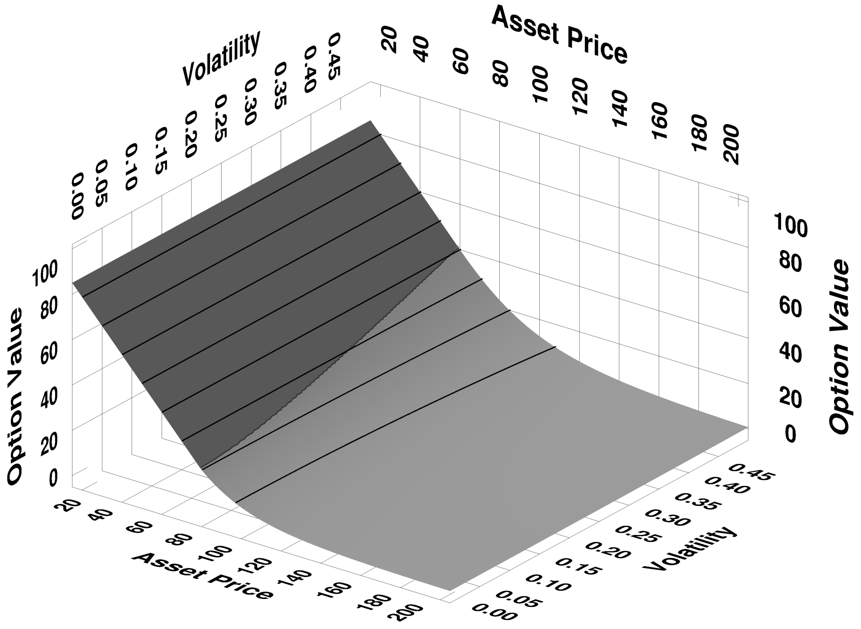

Figure 3.5.5: This is the solution on the finest Plain grid to the American put option for market case B using implicit time stepping. The v axis is scaled as $\sigma = \sqrt{v}$ volatility. The early exercise region is denoted by the darker shading and its boundary is nearly a straight line in volatility scaling.

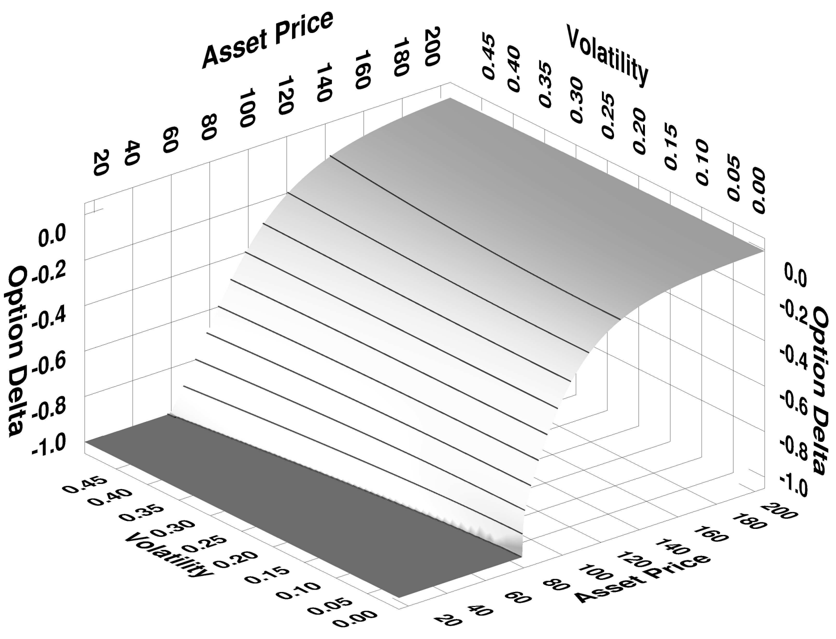

Figure 3.5.6: The option delta with respect to S is given here for the medium resolution Plain grid solution to the American put option for market case B using implicit time stepping. The v axis is scaled as $\sigma = \sqrt{v}$ volatility.

seen, qualitatively, in all test cases: the option delta generated by the solution is smooth, in spite of the lack of a theoretical demonstration to guarantee that the result will be smooth and convergent.

The fourth column of Table 3.5.3 reports the "up-and-in" one-touch contract values for each market case. These are computed with implicit time stepping and central drift term weighting where possible and converge at a linear rate. For comparison, the results using the alternate, rotated grids (described below in Section 3.5.2.3) are also listed. Market case A on the Plain grids shows some erratic convergence between grids 2, 3 and 4, but between grids 1, 2 and 3 the convergence rate is $\alpha = 1.1$.

3.5.2 Results Using the Monotone Approach

The FD grid conditions in Section 3.4.3 result in discrete diffusion terms that are compatible with an M-matrix. Results under these conditions are presented in this section for implicit time stepping only, since this is the monotone case which guarantees the desired stability and convergence properties. The computational cost and solution error of each of the Skew, PosD and Rot grids are evaluated and compared against the Plain grid results. The solution used for comparison was that of the finest grid computed for all four cases, which corresponds to Plain grid resolution 3.

Each grid for a monotone case was created to match some spatial aspect of the Plain grids, with roughly the same number of nodes and attempting to produce the same error. The same time step sizes were used. Each monotone computation of a European put or call required the same number of fixed point iterations per time step as for the Plain grid for all market cases and grid resolutions. The American put and one-touch digital options in the monotone cases required, on average, no more than 1.6 additional iterations per time step than the Plain grid solutions.

Three factors are used to compare the results with those from the Plain grids.

- The **relative FD node count** is given, based on the number of nodes that are not Dirichlet boundary condition points.
- The **relative cost** of each computation is approximated by multiplying the number of non-Dirichlet nodes on the FD grid by the total number of Bi-CGStab iterations used in all fixed-point iterations for all time steps. This estimate represents the bulk of the work done during the solution process. The number of preconditioned Bi-CGStab iterations required increases with condition number of the grid, requiring approximately $\mathrm{O}(N^{5/4})$ iterations when N is the number of FD grid nodes.
- The **relative error** of each computation is computed using the average magnitude of the error at 36 points near $(\log(K), v_0)$. Each error is computed against the semi-analytic solution to the European put and call problem.

Thus the FD node count, effort and error values are all normalized so that the values for the Plain grids would be 1.00.

The cost and error comparison can also be made for the American put and the one-touch options. In these cases there is no semi-analytic solution, so in its place the solution value from the best Plain grid was used instead. The cost and error comparisons for these contracts turned out to be similar to those for the European contracts and are reported for the Rot grids only.

Also, an error comparison may be made using the “delta” $(\partial V/\partial x)$ of the option, which is an important quantity when hedging. The experiments that follow were repeated using error in the delta, rather than error in the price, as the measure for comparison. This did not alter the conclusions.

3.5.2.1 Results on Skew Grids

The Skew grids apply the transformation of Equation 3.4.7. The resolution in x is held identical to that of the Plain grids for nodes placed along $v = 0$. This means that the region near v_0 is also well resolved. With this approach, no control could be imposed over grid spacing in the v direction. One could match the Skew grids to the Plain grids in the v axis direction instead, but this turned out to be a poor approach since it is the x resolution that dominates the error. The grid upper and lower v boundaries were represented by a line as described in Section 3.4.3.1. The DFT grid used was identical to that of the Plain grid.

Skew Grids	FD Grid Relative Node Count	European Call Relative		European Put Relative	
		Cost	Error	Cost	Error
A	0.97	1.08	0.94	1.20	0.93
B	0.97	1.01	0.90	1.01	0.94
C	0.99	1.79	1.31	1.51	0.43
D	0.81	1.28	2.68	1.12	1.59
E	1.02	3.15	0.86	2.53	0.97

Table 3.5.4: Cost and error comparison for Skew grids. The FD grid node count, computational cost and error are given relative to Plain grids. Note the reduced node count for market case D, due to low value in the σ_v parameter, corresponds to a solution with high error. Section 3.5.2 explains the cost and error calculation.

From Table 3.5.4, note that the node count of the grid is nearly exactly that of the Plain grids except for market case D, which had 0.81 the number of Plain grid nodes. The low value of σ_v in market case D led to quite oblique grid lines and the grid became coarse at low values of v. Overall computational effort is usually higher for Skew grids than Plain grids even when the number of FD nodes was nearly identical. This indicates a less well conditioned problem, resulting in a more costly linear system solution. The error is significantly worse in three cases, significantly better only in one.

3.5.2.2 Results on Positive Diffusion Grids

The PosD grids have lines which are aligned with the (x, v) axes, but these adhere to the positive diffusion coefficient grid spacing restrictions of Equation 3.4.8. Near $(\log(K), v_0)$ the PosD grids match the Plain grids along x only. As with Skew grids, this strategy produced better accuracy than attempting to match the spacing along v. The grid spacing restrictions are given in the first columns of Table 3.5.5.

For market cases A and B, $(\Delta x, \Delta v) = (0.02, 0.01)$ on the coarsest grids for Vanilla payoff options near $(\log(K), v_0)$. There was enough flexibility in the spacing to allow Δv to be coarsened away from the region of interest, and to permit the doubling of Δx starting midway between the region of interest from $x = [80, 125]$ and the edge of Ω_C. Market case C restricts the grid spacing to be $(\Delta x, \Delta v) = (0.02, 0.0076)$ everywhere. In cases D and E, Δv was held constant and the spacing of $\Delta x = 0.02$ near the strike was coarsened as with cases A and B, but only by the largest allowable factor, which was less than two. The DFT grids were set to have a spacing which was twice that of the PosD FD grids at $(\log(K), v_0)$ and were aligned to match the nodes of the FD grids.

From Table 3.5.5, for cases A and B, note that the node count of the grid is lower and the computational effort is only ~ 0.35 that of the Plain grids, but the relative error is only $1.13 \rightarrow 1.51$. These are the cases where $|\rho_v|$ is low, σ_v is high and the restrictions on PosD grid

PosD Grids	Minimum $\Delta x/\Delta v$	Maximum $\Delta x/\Delta v$	FD Grid Relative Node Count	European Call Relative Cost	European Call Relative Error	European Put Relative Cost	European Put Relative Error
A	0.16	16.39	0.77	0.34	1.51	0.37	1.32
B	0.37	8.35	0.79	0.36	1.22	0.34	1.13
C	2.63	2.63	2.81	5.62	0.63	5.89	0.85
D	5.86	8.71	3.10	14.44	1.01	14.17	0.97
E	2.93	4.69	2.96	3.94	0.99	3.78	0.99

Table 3.5.5: The PosD grid spacing ratios (Equation 3.4.8) listed in this table generate a positive coefficient discretization on an axis-aligned FD grid. The FD node count relative to Plain grids is given, along with the relative cost and error for the European Put and call options. Note that case C permits no variation in spacing and cases D and E only minor variation in spacing. The leads to higher node counts with no corresponding large reduction in error. See Section 3.5.2 for details of the calculation of cost and error.

generation are minimal.

For market cases C, D and E, the PosD grids tended to have finer overall Δv spacing than the Plain grids, leading to a higher node count and total effort. In these cases $|\rho_v| \to 1$ which forces constant, or nearly constant, spacing and σ_v is low which forces a high $\Delta x/\Delta v$ (low $\Delta v/\Delta x$) ratio. At grid refinement 3, the PosD grid for market case C had 197989 nodes, versus 70560 on the Plain grid. For cases C, D and E the PosD approach incurs 3.78 to 14.44 times the computational cost of the Plain grid and reduces error significantly (0.63 the Plain value) in one case only.

3.5.2.3 Results on Rotated Grids

The Rot grids for the Vanilla payoff options are formed by mirroring the Plain grids on the line $v = 0$ and rotating them around $(\log(K), v_0)$ by θ_r of Equation 3.4.9 as given in the first column of Table 3.5.6. The grid is then trimmed with a line at $v = 0$ and $v = v_D$ as described in Section 3.4.3.1. These grids have the undesirable feature that they cannot accurately capture the option payoff near $x = \log(K)$. Without adjustments, this caused a significant error for the digital one-touch options. To compensate, eight nodes with $\Delta x = 0.0025$ were placed on both sides of $x = \log(K)$ before rotation (which was always in the negative direction). This ensured good resolution of the barrier near $(\log(K), v_0)$. The nodes inside the payoff regions are also shown in Figure 3.5.4 to illustrate how the grid was constructed. The DFT grid was left identical to that of the Plain grid set.

For computation, if a node i with $x_i < \log(K)$ had a neighbor j with $x_j > \log(K)$, then if $(\log(K) - x_i) < (x_j - \log(K))$ the node i was treated as if it was inside the payoff region. Figure 3.5.7 shows the solution for the coarsest grid one-touch option for market case C, where

Rot	FD Grid		European Call		European Put		One-Touch		
Grids	Rotation	Relative					Relative		
	θ_r	Node	Relative		Relative		Node	Relative	
	(Deg)	Count	Effort	Error	Effort	Error	Count	Effort	Error
A	-5.50	1.04	1.11	1.05	1.17	1.01	1.17	1.52	0.86
B	-9.76	1.04	0.98	1.00	1.01	1.00	1.15	1.45	0.95
C	-20.81	1.04	0.70	0.87	0.91	0.94	1.06	1.14	1.18
D	-6.59	1.05	1.05	1.01	1.17	0.98	0.87	0.88	1.06
E	-12.35	1.01	1.76	0.94	1.74	0.96	1.04	3.08	1.09

Table 3.5.6: The FD grid rotation required to generate a rotated diffusion tensor with zero correlation (Equation 3.4.9) is listed along with the FD node count for the Rot grids relative to the Plain grids. The relative cost and error comparisons are given for the European put and call options and for the one-touch options. The best available solution on the Plain grid is used as a reference value against which the error is computed. See Section 3.5.2 for details of the calculation for cost and error.

the rotation was most severe. Note that although the barrier is irregular, the solution becomes smooth rapidly at lower $S = e^x$ points, particularly near $\sigma_0 = 0.1254$.

The fifth column of Table 3.5.3 lists Rot grid option price results and shows the linear rate of convergence for the one-touch digital option. From Table 3.5.6 note that the number of nodes was nearly exactly the same as with the Plain grids for the European option cases. The computational effort was significantly lower for market case C and significantly higher (~ 1.75) only for market case E. The error over the Rot grids was, essentially, no worse than the Plain grids. For interest, the cost and error measurements for the one-touch digital option is also reported. The best result on the Plain grids was used as the reference value against which the error was measured. It required between 0.88 and 3.08 times the numerical effort for results which were within ± 0.18 of the error in Plain grid value.

3.6 Chapter Summary

The governing PIDE for valuing European options under stochastic volatility with jumps is, as in the two-asset case, a linear, infinite-domain, initial-value problem. This chapter adapts the approach of Chapter 2 to create a localized, initial-boundary value problem.

The discretization approach is essentially identical to that of Chapter 2. The diffusion coefficients under stochastic volatility are not constant. The three approaches for generating a monotone diffusion discretization for the constant coefficient case, as presented in Chapter 2, were adapted for this non-constant coefficient case. Boundary conditions at the $v = 0, v_D$ edges of the domain require special treatment for rotated or skewed grids. DFT rotation had to be applied for the case where the degenerate jump PDF was defined obliquely, along a line not aligned with either the x or v axes.

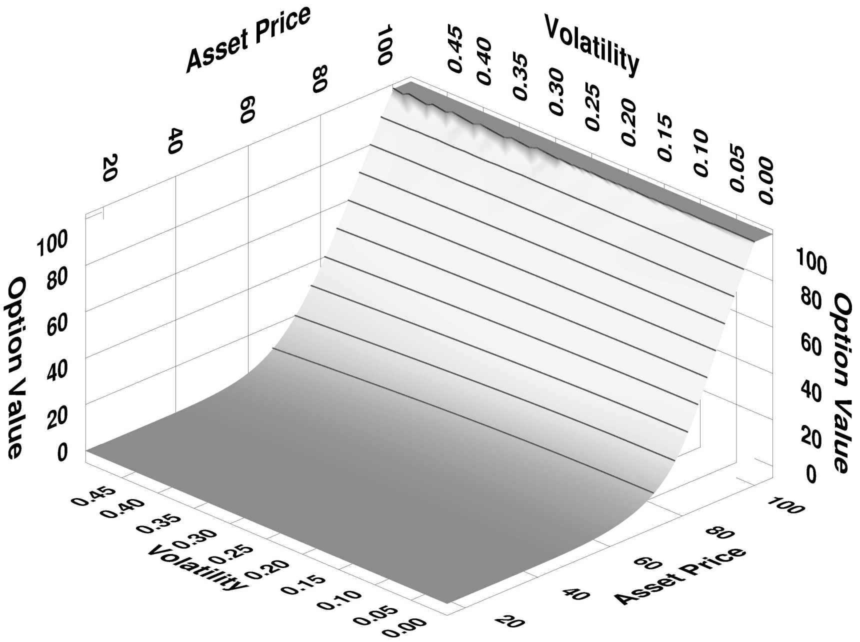

Figure 3.5.7: The option value with respect to price and volatility is given here for the lowest resolution Rot grid solution to the one-touch option for market case C using implicit time stepping. Case C has the largest FD grid rotation. The v axis is scaled as $\sigma = \sqrt{v}$ volatility. Note that the irregularity near $K = 100$ is smoothed inside the solution domain and is not important near the initial volatility value $\sigma_0 = 0.1254$.

The numerical results in this chapter were based on parametrization from five actual market cases. Three of these presented parameters with degenerate or extreme values and, where necessary, the adaptations made in the numerical solution technique were applied. Convergence under grid and time step refinement was shown to be as expected for smooth problems. The main thrust of the numerical demonstrations was to determine which of the three monotone diffusion approaches was the most cost-effective and reliable.

Of the three variant grids which resulted from those three strategies, the rotated grids produced the most favourable results. These had the least variation in cost and error. Thus, in Chapter 4, grid rotation is selected as the approach for ensuring a monotone spatial discretization.

Chapter 4

Option Values where Jump Parameters are Uncertain

4.1 Introduction

Any model of a specific financial market is typically determined from a sample of data by using a calibration procedure. Even after a model structure has been chosen, the parameter values have a degree of uncertainty. The choice and quality of the calibration data [8], data point weighting [33], the measure of goodness of fit and even the calibration procedure [59] can vary to some degree and such factors introduce uncertainty into the final parameter values. The problem of actually fitting models to markets has been studied extensively and will not be examined further in this thesis. Rather, this chapter addresses the challenge of determining a best- or worst-case value for an option contract given a range of parameter values. In this chapter, the solution method for the linear two-factor problems studied in Chapters 2 and 3 is adapted for this non-linear problem.

The "best-case" and "worst-case" option values in this context have precise financial definitions. The best-case value is that which should be charged by the seller, so that a hedging portfolio with a short position in the option is guaranteed to have a non-negative balance at the contract expiry. Conversely, the "worst-case" value is that which should be paid by the buyer so that a hedging portfolio with a long position in the option is guaranteed to have a non-negative balance at contract expiry [6]. Of course, this definition requires an idealized hedge using an infinite number of options trading at infinitesimal intervals. In practice, however, jump-diffusion can be hedged using a small number of options with a small degree of error [70].

Note that parameter uncertainty contains no information other than the allowed range. Any further information, such as how parameters hold over time or in a particular market scenario, would require additional modeling assumptions. In this chapter, an efficient algorithm to study this problem is developed for two-factor models with diffusion and jumps, where the jump parameters are uncertain. The synthetic market used for the numerical demonstrations in this chapter is based on models calibrated to actual markets. The demonstrations not only show that the numerical solution approach is tractable and converges as expected from the theory, but also reveal a wide difference between best- and worst-case prices for a typical variation in parameters. The contribution from each uncertain parameter is evaluated to compare their effect on the option values computed and the spread between the best- and worst-case values.

4.1.1 Previous Work

Early work on the problem by Avellaneda et al. [6] was carried out for one-asset options under the classical Black–Scholes model. They dynamically select the maximum or minimum asset volatility locally to maximize or minimize an option value. It can be shown in this case that the locally optimal choice is also globally optimal. Oztukel and Wilmott [90] extend this to value options using an uncertain risk-free rate as well, treating the uncertain parameters themselves with a stochastic model. Work by Pooley et al. [94, 95] investigated this problem for the case of uncertain parameters within classical one- or two-asset Black–Scholes models. Fedotov and Panayides [45] address the problem using stochastic dynamic programming. Da Fonseca et al. [46] focus on a multi-asset problem with a particular structure of uncertain diffusion correlations which permits a semi-analytic solution.

In this chapter, the problem is posed as one of optimal control of a process governed by Hamilton-Jacobi-Bellman (HJB) partial integro-differential equations with finite activity jumps [89]. Camilli [23] showed that piecewise deterministic approximations of control problems under jump-diffusion converge to the viscosity solution of the continuous problem. Ishikawa [65] further studied the problem of optimal control for jump processes. Forsyth and Labahn [47] provide a general theoretical framework which, when applied to the problem at hand, allows it to be treated using the HJB framework and a finite-difference based approach. Jakobsen et al. [68] provide error bounds and discuss the convergence of such schemes to the viscosity solution in the Bellman PIDE case. They assert that the results also hold for the time-dependent case in combination with the results in [66]. Recent applications of optimal control with PIDE's have been published by Mnif and Sulem [87] and Framstad et al. [50]. Recent work by Wang and Forsyth [105] addresses the issue of optimization in the case where the discretization must change when local control parameter values change.

This work in this chapter begins with, as a basis, the results of [47] and [105] and adapts the approach of Chapters 2 and 3. A two-asset problem is selected since a vanilla contract payoff and option value surface are such that the optimization of the controls is clearly not trivial. Provided the preconditions for convergence to the viscosity solution are satisfied, this approach could also be applied to one-asset problems under stochastic volatility with jumps.

4.1.2 Overview of this Chapter

Section 4.2 presents the governing equations for the problem, which must be altered from Chapter 2 into non-linear, HJB-type PIDE's. Discretization is discussed in Section 4.3. The fixed point policy iteration is described in Section 4.4 and proved to be convergent. Finally, in Section 4.5, numerical demonstrations are provided. These evaluate the rate of convergence of the approach with respect to grid spacing and examine the dependence of the best-case, worst-case spread on the individual control parameters.

4.1.2.1 Synopsis of the Contributions in this Chapter

The following results are presented.

- The discretization method of [30, 31] is extended to controlled HJB equations with jumps. This forms a fully implicit, monotone method. The implicit controls are also discretized over their continuous range. This is of the form used in [18, 23, 68] for the study of the viscosity solutions of related general problems.

- The fixed point method of Chapter 2 is reformulated into the new fixed point policy iteration for the solution of the nonlinear, discretized, algebraic equations at each time step. This allows a significant reduction in the computational cost from that which would be required using simple piecewise constant policy time stepping. The convergence of the fixed point policy iteration is proved under a mild time step restriction. The new iteration is combined with a penalty iteration to compute American option values.

- Numerical demonstrations are provided for both the best-case and worst-case value of European and American options. The overall convergence of the algorithm is shown to be linear or better in the parameter controlling the spacing of the finite difference grid and policy space discretization. The fixed point policy iteration is demonstrated numerically to be rapidly convergent, typically in 4 to 7 iterations per time step. The sensitivity of the gap between best- and worst-case prices to the uncertainty in individual parameters is assessed.

4.2 Governing Equations

The theory and examples in this chapter are developed using the concrete example of an option written on two assets. Thus, the objective is to find the best- or worst-case theoretical value U of a financial option of duration T on a pair of assets S_1 and S_2 at time $t = t_0$. As in Chapter 2, the independent variables are the logarithm of the asset prices (x_1, x_2) and the time remaining to expiry τ as defined in Equations 2.2.1 and 2.2.2. As before, the log-scaled asset prices evolve by a correlated Brownian motion and correlated finite activity jumps which are independent of the Brownian motion. However, the parameters of the jumps are known only to fall within a deterministic range of possible values. For the best-case value this can be written [33, 47, 68] as an optimal control problem on a PIDE over $\Omega_\infty \times [0, T]$

$$\begin{aligned} \frac{\partial U}{\partial \tau} &= \mathcal{G}U - rU + \sup_{Q \in \widehat{Q}} \left\{ \mathcal{H}(Q)U \right\} \\ U(x_1, x_2, 0) &= \mathcal{I}(x_1, x_2) \ . \end{aligned} \tag{4.2.1}$$

In this chapter, operator $\mathcal{G}$ represents the partial differential terms due to Brownian motion and drift, $r \geq 0$ is the risk free rate of return and $\mathcal{H}(Q)$ represents the integral operator due to the arrival of finite activity jumps controlled by a vector $Q \in \widehat{Q}$ of parameters, optimal for each point in $\Omega_\infty \times [0, T]$. For simplicity, the option value is restricted to $U \geq 0$. To compute the worst-case option value, replace sup with inf; only the sup form will be considered further, since the inf form requires only simple changes to the formulation and only obvious changes to the numerical method. The differential operator $\mathcal{G}U$ has constant diffusion parameters $\sigma_1, \sigma_2 \geq 0$ and $|\rho| \leq 1$ over $\Omega_\infty \times [0, T]$ where

$$\mathcal{G}U = \left(r - \frac{{\sigma_1}^2}{2}\right) \frac{\partial U}{\partial x_1} + \left(r - \frac{{\sigma_2}^2}{2}\right) \frac{\partial U}{\partial x_2} + \frac{{\sigma_1}^2}{2} \frac{\partial^2 U}{\partial {x_1}^2} + \rho \sigma_1 \sigma_2 \frac{\partial^2 U}{\partial x_1 \partial x_2} + \frac{{\sigma_2}^2}{2} \frac{\partial^2 U}{\partial {x_2}^2} \ . \tag{4.2.2}$$

The integral operator is defined as

$$\begin{aligned} \mathcal{H}(Q)U = \iint_{-\infty}^{\infty} g(J_1, J_2; Q) \Big[U\left(x_1 + J_1, x_2 + J_2, \tau\right) - U \\ - \left(e^{J_1} - 1\right) \frac{\partial U}{\partial x_1} - \left(e^{J_2} - 1\right) \frac{\partial U}{\partial x_2} \Big] \, dJ_1 \, dJ_2 \end{aligned} \tag{4.2.3}$$

$$\begin{aligned} g &= g(J_1, J_2; Q) \in \mathbb{R} \\ Q &= Q(x_1, x_2, \tau) \in \mathbb{R}^m \end{aligned}$$

where J_1, J_2 are the jump magnitudes and g is the two-dimensional Lévy measure by which they are distributed, defined by a vector Q of m parameters.

This chapter assumes that for all $Q \in \widehat{Q}$, $g(J_1, J_2; Q)$ again defines a finite activity process which satisfies the bounds given in [54] §II.1.2 Definition 1.6 (see also [33] Proposition 3.18) so that the second, third and fourth terms of $\mathcal{H}(Q)$ of Equation 4.2.3 can be written, respectively, as

$$\begin{aligned} \iint_{-\infty}^{\infty} g(J_1, J_2; Q) \, U \, dJ_1 \, dJ_2 &= \lambda(Q) \, U & (4.2.4) \\ \iint_{-\infty}^{\infty} g(J_1, J_2; Q) \left[\left(e^{J_1} - 1\right) \frac{\partial U}{\partial x_1} \right] dJ_1 \, dJ_2 &= \lambda(Q) \, \kappa_1(Q) \, \frac{\partial U}{\partial x_1} & (4.2.5) \\ \iint_{-\infty}^{\infty} g(J_1, J_2; Q) \left[\left(e^{J_2} - 1\right) \frac{\partial U}{\partial x_2} \right] dJ_1 \, dJ_2 &= \lambda(Q) \, \kappa_2(Q) \, \frac{\partial U}{\partial x_2} & (4.2.6) \\ \kappa_1(Q), \kappa_2(Q), \lambda(Q) &\in \mathbb{R} \\ \lambda(Q) &> 0 \\ \left|\kappa_1(Q)\right|, \left|\kappa_2(Q)\right| &< \infty \end{aligned}$$

where $\lambda(Q)$ is the arrival rate of Poisson-distributed jumps and values $\kappa_1(Q), \kappa_2(Q)$ correct for the mean drift in each asset due to jumps. The integral operator differs from Chapters 2 and 3. Here, $g(J_1, J_2, Q)$ varies with time and asset price and includes the $\lambda(Q)$ multiplier.

The PIDE initial conditions $\mathcal{I}(x_1, x_2)$ are defined by the terminal payoff equation of the option contract and a strike price K as in Section 2.2.3.

The linear complementarity problem for American options, modified from Equations 2.2.10

and 2.2.11 for uncertain coefficients, is

$$\frac{\partial U}{\partial \tau} \geq \mathcal{G}U - rU + \sup_{Q \in \widehat{Q}} \{\mathcal{H}(Q)U\} \tag{4.2.7}$$

$$U \geq \mathcal{I} \tag{4.2.8}$$

$$U(x_1, x_2, 0) = \mathcal{I}(x_1, x_2)$$

where at least one of Equations 4.2.7 or 4.2.8 must hold with equality.

4.2.1 Jump Distribution

Although any jump measure satisfying the above noted constraints may be used, this chapter will define $g(J_1, J_2; Q)$ as a bi-variate Normal distribution scaled by the jump arrival rate parameter λ. Thus, the control vector $Q \in \widehat{Q}$ has a length of $m = 6$ elements

$$Q = (\lambda, \check{\mu}_1, \check{\mu}_2, \check{\sigma}_1, \check{\sigma}_2, \check{\rho}) \in \mathbb{R}^6 \ .$$

The finite activity Lévy measure is defined as

$$\begin{aligned} g\big(J_1, J_2; Q\big) &= g\big(J_1, J_2; (\lambda, \check{\mu}_1, \check{\mu}_2, \check{\sigma}_1, \check{\sigma}_2, \check{\rho})\big) \\ &= \lambda \, g_n(J_1, J_2; \check{\mu}_1, \check{\mu}_2, \check{\sigma}_1, \check{\sigma}_2, \check{\rho}) \end{aligned} \tag{4.2.9}$$

where g_n is the conventional bi-variate Normal distribution given in Equation B.1.1. The range of the parameters is defined as a compact, convex region

$$\begin{aligned} Q \in \widehat{Q} = \quad & [\lambda_{\min}, \lambda_{\max}] \\ & \times \big[\check{\mu}_{(1,\min)}, \check{\mu}_{(1,\max)}\big] \times \big[\check{\mu}_{(2,\min)}, \check{\mu}_{(2,\max)}\big] \\ & \times \big[\check{\sigma}_{(1,\min)}, \check{\sigma}_{(1,\max)}\big] \times \big[\check{\sigma}_{(2,\min)}, \check{\sigma}_{(2,\max)}\big] \times [\check{\rho}_{\min}, \check{\rho}_{\max}] \ . \end{aligned} \tag{4.2.10}$$

4.3 Localization and Discretization

Equation 4.2.1 must be localized to a finite domain with appropriate boundary conditions and discretized before it is solved numerically. This can be done by applying the approach of Section 2.3.1, since there are no uncertain parameters in $\Omega_D \setminus \Omega_C$ where only the diffusion component is computed.

The solution of PIDE 4.2.1 on the infinite domain is approximated by solving the PIDE on the finite domains shown in Figure 2.3.1. Let V denote the option value computed on finite domain $\Omega_D \times [0, T]$ by the equations

$$\begin{aligned} \frac{\partial V}{\partial \tau} &= \mathcal{G}V - rV + \begin{cases} \max\limits_{Q \in \widehat{Q}} \big\{\mathcal{H}_D(Q)V\big\} & \text{for } (x_1, x_2) \in \Omega_C \\ 0 & \text{for } (x_1, x_2) \in \Omega_D \setminus \Omega_C \end{cases} \\ V(x_1, x_2, 0) &= \mathcal{I}(x_1, x_2) \\ V(x_1, x_2, \tau) &= B(x_1, x_2, \tau) \ ; \quad (x_1, x_2) \in \partial\Omega_D \end{aligned} \tag{4.3.1}$$

where B represents the boundary conditions at $\partial\Omega_D$, which are Equations 2.3.4 and 2.3.5 on the upper and lower boundaries respectively. The sup operator is replaced by max since $\widehat{Q}$ is a compact region and V is assumed finite. The integral operator $\mathcal{H}_D(Q)V$ must be truncated to Ω_D. Equations 4.2.4 to 4.2.6 are applied to write

$$\mathcal{H}_D(Q)V = \mathcal{J}_D(Q)V - \lambda(Q)\ V - \lambda(Q)\ \left(\kappa_1(Q)\,\frac{\partial V}{\partial x_1} + \kappa_2(Q)\,\frac{\partial V}{\partial x_2}\right) \tag{4.3.2}$$

$$\mathcal{J}_D(Q)V = \iint_{(J_1,J_2)\in\Omega_D} g(J_1 - x_1, J_2 - x_2; Q)\ V\,(J_1, J_2, \tau)\ dJ_1\ dJ_2\ . \tag{4.3.3}$$

This chapter shall proceed under the assumption that the localization approach used in the linear case, proved convergent using Greens functions, is reasonable for the non-linear case and check the assumption numerically. There are some further assumptions on the regularity of the coefficients discussed in Section 4.3.3.

The appropriate size for Ω_D and Ω_C is again determined by the error control approach described in Section 2.3.3 and tolerances $0 < \epsilon_c, \epsilon_d < 1$. The largest distribution at time T of the two assets is computed, starting from a point of interest $(x_1, x_2)_p$, under the permitted policies, which contains $(1 - \epsilon_c)$ of the price paths. This determines the size of Δx_C and Ω_C. Similarly, the largest value in each direction of Δx_D, computed using Equation 2.3.13 from the allowed PDF's $g(J_1, J_2; Q)$, is the one used to determine the size of Ω_D.

4.3.1 Discretization

Discretization is performed as in Section 2.4.1 over a finite difference (FD) grid of N points in space and N_τ points in time

$$(x_1, x_2)_{\mathsf{i}} \in [-X_D, X_D) \times [-X_D, X_D)\ , \qquad \mathsf{i} = 1 \ldots N \tag{4.3.4}$$

$$\tau_n \in [0, T]\ , \qquad n = 0 \ldots N_\tau\ , \quad \tau_n = n\Delta\tau \tag{4.3.5}$$

over which a discrete solution v and control vector q are determined which approximate their continuous counterparts

$$V((x_1, x_2)_{\mathsf{i}}, \tau_n) \simeq \mathsf{v}_{\mathsf{i}}^n \in \mathbb{R}\ , \qquad \mathsf{v}^n \in \mathbb{R}^{N\times 1}$$
$$Q((x_1, x_2)_{\mathsf{i}}, \tau_n) \simeq \mathsf{q}_{\mathsf{i}}^n \in \mathbb{R}^m\ , \quad \mathsf{q}^n \in \mathbb{R}^{N\times m}\ .$$

As in Section 2.4.1, each of the terms of the continuous Equation 4.3.1 has a discrete counterpart. The linear differential operator $\mathcal{G}$ is discretized with second order finite differences for the diffusion term. Second order, central finite differences are used for the drift terms, except where a first order forward or backward difference is required to create a monotone method (see Section 2.5.3 and 4.3.3 below). This choice is made by considering all drift terms together, composed from the first partial derivatives in $\mathcal{G}$ and $\mathcal{H}_D(Q)$ (Equations 4.2.2 and 4.3.2)

$$\left(r - \frac{{\sigma_1}^2}{2} - \lambda(Q)\kappa_1(Q)\right)\frac{\partial V}{\partial x_1} + \left(r - \frac{{\sigma_2}^2}{2} - \lambda(Q)\kappa_2(Q)\right)\frac{\partial V}{\partial x_2}\ . \tag{4.3.6}$$

The discrete terms due to Brownian motion and drift are given at time step $n+1$ with $\tau = \tau_{n+1}$ under policy q^{n+1} by

$$\mathcal{G}V(\tau_{n+1}) \simeq \mathsf{G}(\mathsf{q}^{n+1})\,\mathsf{v}^{n+1} \quad ; \; \mathsf{G}(\mathsf{q}^{n+1}) \in \mathbb{R}^{N\times N} \; .$$

Even though $\mathcal{G}$ does not contain coefficients which depend on Q, the discrete form $\mathsf{G}(\mathsf{q}^{n+1})$ has a dependence on the discrete control. Different values of q^{n+1} may select a different discretization for different point-wise controls when considering the full set of first partial differential terms in Equation 4.3.6.

As in the linear case, the integral operator $\mathcal{H}_D(Q)$ of Equation 4.3.2 is discretized term-by-term, using a second order numerical quadrature for $\mathcal{J}_D(Q)$. This first term is discretized by integrating $g(x_1, x_2; \mathsf{q}^{n+1})$ over the cells surrounding each of the grid nodes in turn to form a dense matrix $\mathsf{J}(\mathsf{q}^{n+1})$. A diagonal matrix $\mathsf{D}(\mathsf{q}^{n+1})$ is formed for the second term. A sparse matrix $\mathsf{P}(\mathsf{q}^{n+1})$ is formed for the differential term by finite differences, using the same discretization (central, forward or backward) used for the first partial terms of $\mathcal{G}$. Thus at time step $n+1$ with $\tau = \tau_{n+1}$ under policy q^{n+1}

$$\mathcal{J}_D(Q)\,V(\tau_{n+1}) \simeq \mathsf{J}(\mathsf{q}^{n+1})\mathsf{v}^{n+1} \tag{4.3.7}$$

$$\lambda(Q)\,V(\tau_{n+1}) \simeq \mathsf{D}(\mathsf{q}^{n+1})\mathsf{v}^{n+1} \tag{4.3.8}$$

$$\lambda(Q)\left(\kappa_1(Q)\,\frac{\partial V(\tau_{n+1})}{\partial x_1} + \kappa_2(Q)\,\frac{\partial V(\tau_{n+1})}{\partial x_2}\right) \simeq \mathsf{P}(\mathsf{q}^{n+1})\mathsf{v}^{n+1} \tag{4.3.9}$$

$$\mathsf{J}(\mathsf{q}^{n+1}), \mathsf{D}(\mathsf{q}^{n+1}), \mathsf{P}(\mathsf{q}^{n+1}) \in \mathbb{R}^{N\times N} \; . \tag{4.3.10}$$

The integral $\mathcal{J}_D(Q)V(\tau_{n+1})$ and its discrete form $\mathsf{J}(\mathsf{q}^{n+1})\mathsf{v}^{n+1}$ in Equation 4.3.7 are correlations. Section 4.4.3 discusses the efficient computation of this term and the difference in approach from the known-parameter, linear case.

The equation for one time step from τ_n to τ_{n+1} in the discrete form of the PIDE is

$$\begin{aligned} \frac{\mathsf{v}^{n+1} - \mathsf{v}^n}{\Delta\tau} &= \max_{\mathsf{q}\in\widehat{Y}} \left\{ \Big[\mathsf{G}(\mathsf{q}) - r\,\mathsf{I} + \mathsf{I_c}\left[\mathsf{J}(\mathsf{q}) - \mathsf{D}(\mathsf{q}) - \mathsf{P}(\mathsf{q})\right]\Big]\mathsf{v}^{n+1} \right\} + \mathsf{b} \\ \mathsf{v}^0_{\mathsf{i}} &= \mathcal{I}((x_1, x_2)_{\mathsf{i}}) \\ \mathsf{b} &\in \mathbb{R}^N \; . \end{aligned} \tag{4.3.11}$$

As in the linear case, vector b encodes the Dirichlet boundary conditions. Only an implicit time step is used so that, provided $\mathsf{G}(\mathsf{q})$ is discretized so as to be monotone spatially, the method is unconditionally monotone.[1] Continuous policy space $\widehat{Q}$ is approximated by discrete policy space $\widehat{Y}$ as detailed in Section 4.3.2 below. The controls must be imposed implicitly (see Section 4.3.3 below) so that the discrete operators are determined by a control $\mathsf{q}^{n+1} \in \widehat{Y}$ which

[1] Higher order time stepping methods would have to satisfy the monotonicity and stability requirements explained in Section 4.3.3. Crank-Nicolson time stepping does so only under a severe time step restriction.

maximizes v^{n+1}. The values of $\mathcal{H}_D(Q)$ are only applied on Ω_C, thus the matrix $\mathsf{I_c}$ of Equation 2.4.7 selects only those points.

It is again convenient to denote the composition of all of the discrete, spatial operators as a single matrix

$$\mathsf{T}(\mathsf{q}^{n+1}) = -\left[\mathsf{G}(\mathsf{q}^{n+1}) - r\,\mathsf{I} + \mathsf{I_c}\left(\mathsf{J}(\mathsf{q}^{n+1}) - \mathsf{D}(\mathsf{q}^{n+1}) - \mathsf{P}(\mathsf{q}^{n+1})\right)\right] \tag{4.3.12}$$

for the purposes of the analysis which follows.

Equation 4.3.11 may be written in a conventional matrix form as

$$\begin{aligned}\left[\mathsf{I} + \Delta\tau\mathsf{T}(\mathsf{q}^{n+1})\right]\mathsf{v}^{n+1} &= \mathsf{v}^n + \mathsf{b} \\ \mathsf{q}^{n+1} &= \arg\max_{\mathsf{q}\in\widehat{Y}}\left\{-\mathsf{T}(\mathsf{q})\mathsf{v}^{n+1}\right\} \ .\end{aligned} \tag{4.3.13}$$

This may also be written in a compact form

$$\begin{aligned}&\mathsf{M}_\mathsf{i}^{n+1}(h, \mathsf{v}_\mathsf{i}^{n+1}, \mathsf{v}_\mathsf{j}^{n+1}, \mathsf{v}_\mathsf{i}^{n}, \mathsf{v}_\mathsf{j}^{n}) = \\ &\left[\mathsf{I} + \Delta\tau\mathsf{T}(\mathsf{q}^{n+1})\right]\mathsf{v}^{n+1} - \mathsf{v}^n - \mathsf{b} = 0\ ; \quad \mathsf{i},\mathsf{j} \in 1\ldots N\ , \quad \mathsf{i}\neq\mathsf{j} \\ &\mathsf{q}^{n+1} = \arg\max_{\mathsf{q}\in\widehat{Y}}\Big\{-\mathsf{T}(\mathsf{q})\,\mathsf{v}^{n+1}\Big\}\end{aligned} \tag{4.3.14}$$

as in Equation 2.4.11. In this form the h parameter also refers to h_y, which defines the grid spacing for $\widehat{Y}$. The optimization of the control is implied.

In Equations 4.3.11, 4.3.13 and 4.3.14, as well as in all future equations for the optimization of vector valued functions, the minimum or maximum is understood to be taken line by line. Thus the value $\mathsf{v}_\mathsf{i}^{n+1}$ on row i of Equation 4.3.13 is maximized by selecting a control $\mathsf{q}_\mathsf{i}^{n+1} \in \widehat{Y}$.

Note that, under a grid skew or rotation, all of the partial differential terms are transformed. This includes those that are discretized to form matrix $\mathsf{P}(\mathsf{q}^{n+1})$ in the jump term in Equation 4.3.9. Hence, the maximization in Equation 4.3.13 is also solved in the transformed co-ordinates. For simplicity, the discussion will continue in the original (x_1, x_2) co-ordinates.

4.3.2 Policy Discretization

Jump distributions used in this method must have a continuous dependence on their parametrization; preferably, that dependence should be Lipschitz continuous. Equation 4.2.9 satisfies the Lipschitz continuity criterion, which permits Equation 4.3.1 to be maximized in a consistent fashion by Equation 4.3.13 by discretizing the controls $\widehat{Q}$ into a finite set $\widehat{Y}$.

Remark 4.3.1 *For a discretization $\widehat{Y}$ of continuous controls $\widehat{Q}$, with spacing h_y in each control dimension, recall that the definition of consistency relevant to viscosity solutions is defined in terms of a smooth test function $\phi(x_1, x_2)$ [105]. The discretization of the controls is consistent if*

$$\left|\left(\frac{\partial\phi}{\partial\tau} - \mathcal{G}\phi + r\phi - \max_{Q\in\widehat{Q}}\{\mathcal{H}_D(Q)\phi\}\right) - \left(\frac{\partial\phi}{\partial\tau} - \mathcal{G}\phi + r\phi - \max_{Q\in\widehat{Y}}\{\mathcal{H}_D(Q)\phi\}\right)\right| \to 0$$

$$\textit{as} \quad h_y \to 0\ .$$

Theorem 4.3.2 *Let $\mathcal{H}_D(Q)$ be a continuous function of the control vector Q. If the control space $\widehat{Q}$ is discretized by a finite set of controls $\widehat{Y}$ with a spacing of h_y then, for a continuous test function $\phi(x_1, x_2)$, the solution to Equation 4.3.1 over $Q \in \widehat{Y}$ is consistent with the solution over $Q \in \widehat{Q}$. If $\mathcal{H}_D(Q)$ is a Lipschitz continuous function of the control vector Q then the solution to Equation 4.3.1 over $Q \in \widehat{Y}$ is consistent with the solution over $Q \in \widehat{Q}$ to $\mathrm{O}(h_y)$.*

Proof. This result follows from [23, 68] by applying the approach of [105]. □

To discretize $\widehat{Q}$, first note that the integral term may be written from Equations 4.3.2, 4.3.3 and jump measure Equation 4.2.9 as

$$\begin{aligned} \max_{Q \in \widehat{Q}} \Big\{ \mathcal{H}_D(Q) V \Big\} &= \max_{Q \in \widehat{Q}} \Big\{ \lambda(Q) \left(\mathcal{B}^R V \right) \Big\} \\ \mathcal{B}^R V &= \Bigg[\iint_{(J_1,J_2) \in \Omega_D} g_n(J_1 - x_1, J_2 - x_2; R) \, V(J_1, J_2, \tau) \; dJ_1 \, dJ_2 \\ &\qquad\qquad - V - \kappa_1(R) \, \frac{\partial V}{\partial x_1} - \kappa_2(R) \, \frac{\partial V}{\partial x_2} \Bigg] \\ R &= (\breve{\mu}_1, \breve{\mu}_2, \breve{\sigma}_1, \breve{\sigma}_2, \breve{\rho}) \\ (\lambda \times R) &= Q \in \widehat{Q} \end{aligned} \tag{4.3.15}$$

where $g_n(J_1, J_2; R)$ is the PDF of the bi-variate Normal distribution and g_n and $\mathcal{B}^R V$ have no dependence on $\lambda(Q)$ in their continuous form. Noting that $\lambda(Q)$ is a multiple of $\mathcal{B}^R V$ and

$$\lambda(Q) = \begin{cases} \lambda_{\min} & \text{if } \ \mathcal{B}^R V < 0 \\ \lambda_{\max} & \text{if } \ \mathcal{B}^R V > 0 \end{cases} \quad \text{to select } \max_{Q \in \widehat{Q}} \Big\{ \lambda(Q) \left(\mathcal{B}^R V \right) \Big\} ,$$

it seems reasonable that $\lambda(Q)$ is a so-called “bang-bang” constraint, since the maximum is attained at extreme values of $\lambda(Q)$.

At least the minimum and maximum points of each element of Q must be present in the discrete form of $\widehat{Q}$. At each refinement an evenly spaced grid will be created with two nodes in the λ dimension and N_q nodes in each remaining dimension

$$\begin{aligned} Q \in \widehat{Y} = \Big\{ \quad & (\lambda_{\min}, \lambda_{\max}) \\ & \times \left(\breve{\mu}_{(1,\min)}, \breve{\mu}_{(1,\min)} + h_1, \ldots, \breve{\mu}_{(1,\max)} \right) \times \left(\breve{\mu}_{(2,\min)}, \breve{\mu}_{(2,\min)} + h_2, \ldots, \breve{\mu}_{(2,\max)} \right) \\ & \times \left(\breve{\sigma}_{(1,\min)}, \breve{\sigma}_{(1,\min)} + h_3, \ldots, \breve{\sigma}_{(1,\max)} \right) \times \left(\breve{\sigma}_{(2,\min)}, \breve{\sigma}_{(2,\min)} + h_4, \ldots, \breve{\sigma}_{(2,\max)} \right) \\ & \times \left(\breve{\rho}_{\min}, \breve{\rho}_{\min} + h_5, \ldots, \breve{\rho}_{\max} \right) \Big\} \end{aligned} \tag{4.3.16}$$

where $h_1, \ldots, h_5 = \mathrm{O}(h_y)$. The total number of discrete policies is denoted $N_Y = 2{N_q}^{m-1}$ where in this case $m = 6$ policy dimensions including $\lambda(Q)$. The coarsest $\widehat{Y}$ grid used in this chapter will have $N_q = 3$ nodes in the $(\breve{\mu}_1, \breve{\mu}_2, \breve{\sigma}_1, \breve{\sigma}_2, \breve{\rho})$ dimensions, which will then be refined by halving the spacing.

Although $\lambda(Q)$ appears to be "bang-bang" in the continuous form of the problem, it is not likely to be so once the problem is discretized. The question of whether $\lambda(Q)$ converges to a "bang-bang" constraint as $h_y \to 0$ will be checked numerically. This is done by comparing the results using $Q \in \widehat{Y}$ with results using a "no bang-bang λ" discretization of the controls

$$Q \in \widehat{Y}_\lambda = \big\{(\lambda_{\min}, \lambda_{\min} + h_\lambda, \ldots, \lambda_{\max}) \times \ldots\big\} \tag{4.3.17}$$

where $\lambda(Q)$ is discretized along with the other parameters. By Theorem 4.3.2, if $\lambda(Q)$ is a "bang-bang" constraint then the solution using a "no-bang-bang" discretization of the $\lambda(Q)$ control must be convergent to the "bang-bang" solution as $h_\lambda \to 0$.

4.3.3 Stability, Convergence and a Monotone Method

The reader is referred to [23, 47, 65, 66, 68, 89, 105] for the general theory behind the solution of optimal control problems with discrete, implicitly imposed controls and [50, 87] for some recent applications under jump processes. The following definitions and constraints for the HJB-type PIDE of this chapter are similar to those of Section 2.5, but repeated here for clarity.

The finite difference discretization of $\mathsf{G}(\mathsf{q}^{n+1})$ and $\mathsf{P}(\mathsf{q}^{n+1})$ as well as the quadrature scheme used for $\mathsf{J}(\mathsf{q}^{n+1})$ match that of [68] on grids with constant spacing. The region $\Omega_D \setminus \Omega_C$ is analogous to the extended domains of [18, 34] in the linear case and all boundary conditions may be treated as Dirichlet.

Theorem 4.3.3 *If $\mathsf{T}(\mathsf{q}^{n+1})$ of Equation 4.3.12 is an M-matrix then the time step method of Equation 4.3.13 is unconditionally strictly stable in the l_∞ norm, i.e.*

$$\left\|\mathsf{v}^{n+1}\right\|_\infty \le c \cdot \max\Big[\left\|\mathcal{I}(x_1, x_2)\right\|_\infty, \left\|B(x_1, x_2, \tau)\right\|_\infty\Big]$$

where the constant c is independent of grid spacing Δx_1, Δx_2, $\Delta\tau$ or Δh_y. This is stability in the sense of Definition 2.5.2.

Proof. As in the linear case in Section 2.5, this result is proved by a simple maximum analysis which includes the boundary conditions. □

Theorem 4.3.4 *If $\mathsf{T}(\mathsf{q}^{n+1})$ of Equation 4.3.12 is an M-matrix then the discretized Equation 4.3.13, written as $\mathsf{M}_\mathsf{i}^{n+1}$ in Equation 4.3.14, is a monotone scheme as in Definition 2.5.3.*

Proof. If $\mathsf{T}(\mathsf{q}^{n+1})$ is an M-matrix then the scheme of Equation 4.3.13 is a positive coefficient discretization (see Section 2.5.3). Note also that

$$\big[\max f(\cdot) - \max g(\cdot)\big] \le \max\big[f(\cdot) - g(\cdot)\big]\ .$$

Then the result follows from the same steps as [47] Lemma 5.3, where $f(\cdot)$ and $g(\cdot)$ are replaced by the perturbed and unperturbed values of $\mathsf{M}_\mathsf{i}^{n+1}(\cdot)$ in the Definition 2.5.3 of monotonicity, Equation 2.5.1. □

Definition 4.3.5 *Equation 4.3.13, written as $\mathsf{M}_{\mathsf{i}}^{n+1}$ in Equation 4.3.14, is a consistent scheme if, for any smooth function $\phi((x_1, x_2), \tau)$ with a vector $\bar{\phi}$ where $\bar{\phi}_{\mathsf{i}}^{n} = \phi((x_1, x_2)_{\mathsf{i}}, \tau_n)$ over the $\mathsf{i} = 1 \ldots N$ finite difference points,*

$$\left| \left(\frac{\partial \phi}{\partial \tau} - \mathcal{G}\phi + r\phi - \max_{Q \in \widehat{Y}} \{ \mathcal{H}_D(Q)\phi \} \right)_{\mathsf{i}}^{n+1} - \mathsf{M}_{\mathsf{i}}^{n+1}(h, \bar{\phi}_{\mathsf{i}}^{n+1}, \bar{\phi}_{\mathsf{j}}^{n+1}, \bar{\phi}_{\mathsf{i}}^{n}, \bar{\phi}_{\mathsf{j}}^{n}) \right| \to 0$$

$$as \quad h \to 0 \ .$$

Remark 4.3.6 *The form of consistency in Definition 4.3.5 is all that is required, rather than the more involved form of, for example [9, 10, 18]. This is so when either [9, 47]*

- *the boundary conditions are the limit of the PIDE from the interior, or*
- *the PIDE is non-degenerate normal to the boundary.*

Theorem 4.3.7 *The discrete Equation 4.3.13 is consistent with PIDE 4.3.1.*

Proof. This follows from Theorem 4.3.2 and the Taylor series expansion used to create the discretization. □

For convergence to a viscosity solution, it is generally necessary [67, 68] that regularity conditions hold for the coefficients of the continuous PIDE 4.3.1 (typically that they are Lipschitz continuous in (x_1, x_2)). This is not the case in Equation 4.3.1 at $\partial\Omega_C$. However, this difficulty can be avoided by using a smooth mollifier function which makes $\mathcal{H}_D(Q)V \to 0$ smoothly as $(x_1, x_2) \in \Omega_C \to \Omega_D$. Under this regularization, the localized PIDE 4.3.1 satisfies the requirements of [67] for a maximum principle over a finite spatial domain, notably also of [67] Remark 3.4 that the integral term not permit jumps outside the finite domain.

In [18, 23, 68], the requirements for monotonicity, l_∞-norm stability and consistency are required to prove the convergence of the numerical scheme to the viscosity solution of a nonlinear PIDE. PIDE 4.3.1 does not match the problems in [18, 23, 68] in every respect. Hence, this work does not claim a rigorous proof of convergence to the viscosity solution. This study proceeds under the assumption that because the scheme used here is monotone, stable and consistent that it does converge to the viscosity solution.

Remark 4.3.8 *Matrix $\mathsf{T}(\mathsf{q}^{n+1})$ of Equation 4.3.12 will be an M-matrix if $-\left(\mathsf{G}(\mathsf{q}^{n+1}) - \mathsf{I_c}\mathsf{P}(\mathsf{q}^{n+1})\right)$ and $\left[-\mathsf{I_c}\left(\mathsf{J}(\mathsf{q}^{n+1}) - \mathsf{D}(\mathsf{q}^{n+1})\right)\right]$, which are components of $\mathsf{T}(\mathsf{q}^{n+1})$, are M-compatible (see Definition 2.5.8).*

Section 3.5.2 indicates that of the strategies listed in Section 2.5.3, in addition to central weighting as much as possible, grid rotation is to be preferred to ensure a monotone spatial discretization. The entries of $\mathsf{J}(\mathsf{q}^{n+1})$ in Equation 4.3.7 are defined by integrating the values of $g(J_1, J_2; \mathsf{q}^{n+1})$ over the FD grid cells. Since the integral of the entire function g around $(x_1, x_2)_{\mathsf{i}}$

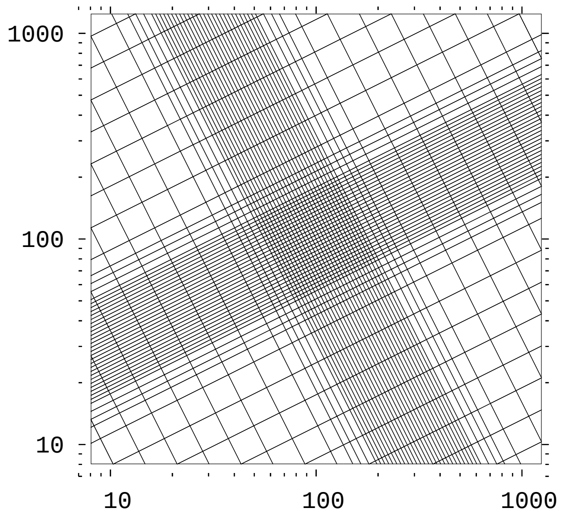

Figure 4.3.1: The coarsest finite difference grid used in the numerical demonstrations is rotated, but has a boundary aligned with the axes. This grid has 2399 nodes and a grid spacing at the strike $(\log(K), \log(K))$ of $(\Delta x_1, \Delta x_2) = (0.04, 0.04)$.

is, by definition, the $\lambda(\mathsf{q}^{n+1})$ value at node i then

$$\begin{aligned} & \left[\mathsf{J}(\mathsf{q}^{n+1})\right]_{\mathsf{ij}} \geq 0 \\ \text{and} \quad & \sum_{\mathsf{j}} \left[\mathsf{J}(\mathsf{q}^{n+1})\right]_{\mathsf{ij}} = \lambda(\mathsf{q}^{n+1})_{\mathsf{i}} \\ \text{with} \quad & \left[\mathsf{D}(\mathsf{q}^{n+1})\right]_{\mathsf{ii}} = \lambda(\mathsf{q}^{n+1})_{\mathsf{i}} \ . \end{aligned} \tag{4.3.18}$$

As in the linear case, $\left[\mathsf{I}_{\mathsf{c}} \left(\mathsf{J}(\mathsf{q}^{n+1}) - \mathsf{D}(\mathsf{q}^{n+1})\right)\right]$ is M-compatible.

4.3.3.1 Boundary Considerations

Grid rotation produces an irregular boundary along $(x_1, x_2) = (\pm X_D, \pm X_D)$. This, in turn, complicates the otherwise simple boundary conditions specified in Equation 2.3.5 on the lower edges. The approach taken in Chapter 3 is the one used here: the boundaries are represented by a line which truncates the rotated grid. Since there are no terms in boundary condition Equation 2.3.5 which are normal to the boundary, this simply puts an irregularly spaced, one-dimensional grid along the lower edges. An example of the type of rotated grid used for numerical demonstrations is given in Figure 4.3.1.

4.4 Time Step Solution Techniques

Equation 4.3.13 is unlike the time step for the jump-diffusion problems in Chapters 2 and 3 or of [37, 38, 48, 111]: $\mathsf{T}(\mathsf{q}^{n+1})$ is an implicit function of the time step solution v^{n+1}, making $\mathsf{T}(\mathsf{q}^{n+1})$ non-linear even for European options.

In this section, the fixed point iteration is modified to introduce a policy selection step, which is performed together with the time step solution. This time step solution method implicitly imposes the policy constraints for Equation 4.3.13. By using the computational savings techniques detailed in Section 4.4.3 below, the cost of the time step solution becomes considerably lower than is possible with the existing approach: piecewise constant time stepping. This is analyzed in Section 4.4.4. Penalty iteration for American options is readily combined with the new approach, as described in Section 4.4.5. The resulting time step solution algorithm is given below as Algorithm 4.4.1 for both European and American options.

4.4.1 Alternate Approach: Piecewise Constant Time Stepping

At first glance, piecewise constant policy time stepping [47, 74] could be applied to solve for the next time step value v^{n+1} of Equation 4.3.13. This is a well-studied approach which starts by solving for $\mathsf{v}^{n+1}(\mathsf{e})$ for each time step matrix $\mathsf{T}(\mathsf{e})$ that corresponds to a constant policy $\mathsf{e} \in \widehat{Y}$. For each constant policy evaluation, the control vector is set to the same policy $\mathsf{q}_{\mathsf{i}}^{n+1} = \mathsf{e}\ \forall\ \mathsf{i} = 1 \ldots N$ for each solution node in the spatial grid.

Thus, this time step approach first solves

$$\Big[\mathsf{I} + \Delta\tau\, \mathsf{T}(\mathsf{e})\Big] \mathsf{v}^{n+1}(\mathsf{e}) = \mathsf{v}^n + \mathsf{b} \quad \forall\ \mathsf{e} \in \widehat{Y} \tag{4.4.1}$$

then, as the second step, computes

$$\big[\mathsf{v}^{n+1}\big]_{\mathsf{i}} = \left[\max_{\mathsf{e}\in\widehat{Y}} \big(\mathsf{v}^{n+1}(\mathsf{e})\big)\right]_{\mathsf{i}} \qquad \forall\ \mathsf{i} = 1 \ldots N\ . \tag{4.4.2}$$

Thus, the solution v^{n+1} and the policy q is selected, point-wise at each grid node, from the maximum of the implicit time step solutions over each of the N_Y constant policies.

Using the fixed point iteration method would require M_f iterations, consisting of one sparse linear system solution and two FFT's, for each of the N_Y constant policy solutions in Equation 4.4.1. From Chapter 2, $M_f = 3$ to 5 iterations would typically be required to converge to the solution of Equation 4.4.1 for European, two-asset option problems with a single, constant policy.

4.4.2 Fixed Point Policy Iteration

Fixed point policy iteration computes a series of intermediate solutions z^k, starting with an initial guess $\mathsf{z}^0 = \mathsf{v}^n$, which converge to the solution v^{n+1} of Equation 4.3.13. At the start of each iteration, the optimal policy q^k is determined for iterate z^k. The corresponding matrices

$\mathsf{G}(\mathsf{q}^k)$, $\mathsf{J}(\mathsf{q}^k)$, $\mathsf{D}(\mathsf{q}^k)$ and $\mathsf{P}(\mathsf{q}^k)$ are constructed so that

$$\Big[\mathsf{G}(\mathsf{q}^k) + \mathsf{I_c}\left(\mathsf{J}(\mathsf{q}^k) - \mathsf{D}(\mathsf{q}^k) - \mathsf{P}(\mathsf{q}^k)\right)\Big]\,\mathsf{z}^k = \max_{\mathsf{q}\in\widehat{Y}}\Big\{\big[\mathsf{G}(\mathsf{q}) + \mathsf{I_c}\;(\mathsf{J}(\mathsf{q}) - \mathsf{D}(\mathsf{q}) - \mathsf{P}(\mathsf{q}))\big]\mathsf{z}^k\Big\}\ . \tag{4.4.3}$$

The solution will be unique, as proved below, but the policy may not be unique. In this case the choice between policies that produce the same value in Equation 4.4.3 will be made arbitrarily. In order to avoid solving a system with the $\mathsf{J}(\mathsf{q})$ term in the LHS, as in Equation 4.3.13 (noting Equation 4.3.12), the fixed point policy iteration splits the linear system into

$$\Big\{\mathsf{I} - \Delta\tau\left[\mathsf{G}(\mathsf{q}^k) - r\,\mathsf{I} - \mathsf{I_c}\left(\mathsf{D}(\mathsf{q}^k) + \mathsf{P}(\mathsf{q}^k)\right)\right]\Big\}\,\mathsf{z}^{k+1} = \Delta\tau\,\mathsf{I_c}\mathsf{J}(\mathsf{q}^k)\,\mathsf{z}^k + \mathsf{v}^n + \mathsf{b} \tag{4.4.4}$$

$$\mathsf{q}^k = \arg\max_{\mathsf{q}\in\widehat{Y}}\Big\{\Big[\mathsf{G}(\mathsf{q}) + \mathsf{I_c}\left[\mathsf{J}(\mathsf{q}) - \mathsf{D}(\mathsf{q}) - \mathsf{P}(\mathsf{q})\right]\Big]\mathsf{z}^k\Big\} \tag{4.4.5}$$

and solves for z^{k+1}. The solution of Equation 4.4.5 is discussed in Section 4.4.3 below. Convergence is reached when the relative update is less than a tolerance ϵ_u. Linear system Equation 4.4.4 is solved using preconditioned Bi-CGStab, as previously discussed in Section 2.6.3.

In order to prove the convergence of iteration Equations 4.4.4 and 4.4.5, a number of intermediate results are required.

Remark 4.4.1 *For brevity in the following development, denote the matrices in the LHS and RHS of Equation 4.4.4 and 4.4.5 as*

$$\begin{aligned}\mathsf{N}^k &= \mathsf{I} - \Delta\tau\;\left(\mathsf{G}(\mathsf{q}^k) - r\,\mathsf{I} - \mathsf{I_c}\left[\mathsf{D}(\mathsf{q}^k) + \mathsf{P}(\mathsf{q}^k)\right]\right)\\ \mathsf{B}^k &= \Delta\tau\mathsf{I_c}\mathsf{J}(\mathsf{q}^k)\end{aligned} \tag{4.4.6}$$

to re-write Equation 4.4.4 as

$$\mathsf{N}^k\mathsf{z}^{k+1} = \mathsf{B}^k\mathsf{z}^k + \mathsf{v}^n + \mathsf{b}\ . \tag{4.4.7}$$

Lemma 4.4.2 ***Norm of the Iteration Matrices.*** *Assume that* $(\mathsf{G}(\mathsf{q}^k) - \mathsf{I_c}\,\mathsf{P}(\mathsf{q}^k))$ *are M-compatible and satisfy Remark 4.3.8 so that* N^k *is an M-matrix (Definition 2.5.6). Then the norm*

$$\left\|\left(\mathsf{N}^k\right)^{-1}\,\mathsf{B}^k\right\|_\infty \le c_1 = \frac{\Delta\tau\,\lambda_{max}}{1 + \Delta\tau\,(\lambda_{max} + r)} < 1 \tag{4.4.8}$$

and similarly

$$\left\|\left(\mathsf{N}^k\right)^{-1}\,\mathsf{B}^{k-1}\right\|_\infty \le c_2 = \frac{\Delta\tau\,\lambda_{max}}{1 + \Delta\tau\,(\lambda_{min} + r)} \tag{4.4.9}$$

where λ_{min} *and* λ_{max} *are defined in Equation 4.2.10.*

Proof. Note that the elements of $\big(\mathsf{G}(\mathsf{q}^k) - \mathsf{I_c}\mathsf{P}(\mathsf{q}^k)\big)$ satisfy

$$-\big(\mathsf{G}(\mathsf{q}^k) - \mathsf{I_c}\mathsf{P}(\mathsf{q}^k)\big)_{\mathsf{ii}} \ge \sum_{\mathsf{j}\neq\mathsf{i}}\big(\mathsf{G}(\mathsf{q}^k) - \mathsf{I_c}\mathsf{P}(\mathsf{q}^k)\big)_{\mathsf{ij}} \ge 0\ .$$

Consider two vectors $\mathsf{y}, \mathsf{z} \in \mathbb{R}^N$ such that, for an arbitrary z, y satisfies

$$\mathsf{N}^k \mathsf{y} = \mathsf{B}^k \mathsf{z} \ . \tag{4.4.10}$$

Split Equation 4.4.10 as follows for row i using the definitions in Equation 4.4.6 and apply Equation 4.3.18 for the row sum of B^k

$$\begin{aligned} &[1 + \Delta\tau \ \left(r + (\mathsf{I_c})_{\mathsf{ii}} \, \lambda(\mathsf{q}^k)_\mathsf{i}\right) - \Delta\tau\big(\mathsf{G}(\mathsf{q}^k) - \mathsf{I_c}\mathsf{P}(\mathsf{q}^k)\big)_{\mathsf{ii}}\Big] \, \mathsf{y}_\mathsf{i} \\ &\qquad = \sum_{\mathsf{j} \neq \mathsf{i}} \left[\Delta\tau \left(\mathsf{G}(\mathsf{q}^k) - \mathsf{I_c}\mathsf{P}(\mathsf{q}^k)\right)_{\mathsf{ij}} \mathsf{y}_\mathsf{j}\right] + \left(\mathsf{B}^k \mathsf{z}\right)_\mathsf{i} \\ &\qquad \leq \sum_{\mathsf{j} \neq \mathsf{i}} \left[\Delta\tau \left(\mathsf{G}(\mathsf{q}^k) - \mathsf{I_c}\mathsf{P}(\mathsf{q}^k)\right)_{\mathsf{ij}}\right] \|\mathsf{y}\|_\infty + \Delta\tau \, (\mathsf{I_c})_{\mathsf{ii}} \, \lambda(\mathsf{q}^k)_\mathsf{i} \, \|\mathsf{z}\|_\infty \ . \end{aligned} \tag{4.4.11}$$

Let $1 \leq \mathsf{i} \leq N$ be an index of y such that $\mathsf{y}_\mathsf{i} = \|\mathsf{y}\|_\infty$. Then, from Equation 4.4.11

$$\begin{aligned} \left[1 + \Delta\tau \left(r + (\mathsf{I_c})_{\mathsf{ii}} \, \lambda(\mathsf{q}^k)_\mathsf{i}\right)\right] \|\mathsf{y}\|_\infty &\leq \Delta\tau \, (\mathsf{I_c})_{\mathsf{ii}} \, \lambda(\mathsf{q}^k)_\mathsf{i} \, \|\mathsf{z}\|_\infty \\ \|\mathsf{y}\|_\infty &\leq \frac{\Delta\tau \ (\mathsf{I_c})_{\mathsf{ii}} \, \lambda(\mathsf{q}^k)_\mathsf{i}}{1 + \Delta\tau \ [r + (\mathsf{I_c})_{\mathsf{ii}} \, \lambda(\mathsf{q}^k)_\mathsf{i}]} \, \|\mathsf{z}\|_\infty \\ \|\mathsf{y}\|_\infty &\leq \frac{\Delta\tau \, \lambda_{\max}}{1 + \Delta\tau \, (\lambda_{\max} + r)} \, \|\mathsf{z}\|_\infty \end{aligned} \tag{4.4.12}$$

which implies Equation 4.4.8 as required.

If $\mathsf{N}^k \mathsf{z} = \mathsf{B}^{k-1} \mathsf{y}$ then Equation 4.4.12 must be changed to allow for $\lambda(\mathsf{q}^k)$ on the left and $\lambda(\mathsf{q}^{k-1})$ on the right hand side. Matrix $\mathsf{I_c}$ does not change. Thus

$$\begin{aligned} \left[1 + \Delta\tau \left(r + (\mathsf{I_c})_{\mathsf{ii}} \, \lambda(\mathsf{q}^k)_\mathsf{i}\right)\right] \|\mathsf{y}\|_\infty &\leq \Delta\tau \, (\mathsf{I_c})_{\mathsf{ii}} \, \lambda(\mathsf{q}^{k-1})_\mathsf{i} \, \|\mathsf{z}\|_\infty \\ \|\mathsf{y}\|_\infty &\leq \frac{\Delta\tau \, \lambda_{\max}}{1 + \Delta\tau \, (\lambda_{\min} + r)} \, \|\mathsf{z}\|_\infty \end{aligned}$$

which implies Equation 4.4.9 as required. □

Lemma 4.4.3 ***Bounded Iterates.*** *Under the preconditions of Lemma 4.4.2, the norm* $\|\mathsf{z}^{k+1}\|_\infty$ *of the solution to Equation 4.4.4 with* $\mathsf{z}^0 = \mathsf{v}^n$ *is bounded by*

$$\|\mathsf{z}^{k+1}\|_\infty < c_1 \, \|\mathsf{v}^n\|_\infty + \frac{c_3}{1 - c_1} \, \|\mathsf{v}^n + \mathsf{b}\|_\infty \tag{4.4.13}$$

where $0 < c_1 < 1$ *as given by Equation 4.4.8 and* $c_3 < \infty$ *are both constants independent of iteration number* k.

Proof. Iteration Equation 4.4.4, written using Equation 4.4.6 to form Equation 4.4.7, can be rearranged as

$$\mathsf{z}^{k+1} = \left(\mathsf{N}^k\right)^{-1} \cdot \left[\mathsf{B}^k \mathsf{z}^k + \mathsf{v}^n + \mathsf{b}\right] \ . \tag{4.4.14}$$

Taking the l_∞ norm of both sides and applying the triangle inequality and Equation 4.4.8 of Lemma 4.4.2

$$\begin{aligned}
\left\|\mathsf{z}^{k+1}\right\|_\infty &\le \left\|\left(\mathsf{N}^k\right)^{-1}\mathsf{B}^k\right\|_\infty \cdot \left\|\mathsf{z}^k\right\|_\infty + \left\|\left(\mathsf{N}^k\right)^{-1}\right\|_\infty \|\mathsf{v}^n + \mathsf{b}\|_\infty \\
&\le \frac{\Delta\tau\ \lambda_{\max}}{1+\Delta\tau\,(\lambda_{\max}+r)}\left\|\mathsf{z}^k\right\|_\infty + c_3\left\|\mathsf{v}^n+\mathsf{b}\right\|_\infty \qquad (4.4.15)\\
&\le c_1\left\|\mathsf{z}^k\right\|_\infty + c_3\left\|\mathsf{v}^n+\mathsf{b}\right\|_\infty
\end{aligned}$$

where c_1 is defined in Equation 4.4.8 and $c_3 = \left\|\left(\mathsf{N}^k\right)^{-1}\right\|_\infty < \infty$ because N^k is an M-matrix. Assuming $\mathsf{z}^0 = \mathsf{v}^n$ (as in Equation 4.4.4), then the expanded recursion of Equation 4.4.15 is bounded by a geometric series in $0 < c_1 < 1$

$$\begin{aligned}
\left\|\mathsf{z}^{k+1}\right\|_\infty &\le {c_1}^k\,\|\mathsf{v}^n\| + c_3\left(\sum_{i=0}^{k-1}{c_1}^i\right)\|\mathsf{v}^n+\mathsf{b}\|_\infty \\
&< c_1\,\|\mathsf{v}^n\| + \frac{c_3}{1-c_1}\,\|\mathsf{v}^n+\mathsf{b}\|_\infty \ .
\end{aligned}$$

□

Remark 4.4.4 *Define*

$$\begin{aligned}
\mathsf{Y}^k\mathsf{z}^k = &\quad \left[\left(\mathsf{N}^{k-1}-\mathsf{B}^{k-1}\right)-\left(\mathsf{N}^k-\mathsf{B}^k\right)\right]\mathsf{z}^k \\
= \Delta\tau\Big\{ &\left[\mathsf{G}(\mathsf{q}^k)+\mathsf{I_c}\left(\mathsf{J}(\mathsf{q}^k)-\mathsf{D}(\mathsf{q}^k)-\mathsf{P}(\mathsf{q}^k)\right)\right] \qquad (4.4.16)\\
&-\left[\mathsf{G}(\mathsf{q}^{k-1})+\mathsf{I_c}\left(\mathsf{J}(\mathsf{q}^{k-1})-\mathsf{D}(\mathsf{q}^{k-1})-\mathsf{P}(\mathsf{q}^{k-1})\right)\right]\Big\}\,\mathsf{z}^k
\end{aligned}$$

where q^k and q^{k-1} solve Equation 4.4.5 for z^k and z^{k-1} respectively. Then, by construction (see Equation 4.4.5),

$$\mathsf{Y}^k\,\mathsf{z}^k \ge \mathit{0}\ .$$

Equality is reached if the policy applied at iterations k and $k-1$ both maximize the result, or if they are the same policy.

Theorem 4.4.5 ***Convergence of the Fixed Point Policy Iteration.*** *Provided the preconditions of Lemma 4.4.2 hold and the time step $\Delta\tau$ is restricted to ensure that Equation 4.4.9 satisfies $0 < c_2 < 1$, then the fixed point policy iteration of Equations 4.4.4 and 4.4.5, with $\mathsf{z}^0 = \mathsf{v}^n$, is convergent to the unique solution of time step Equation 4.3.13.*

Proof. Iteration Equation 4.4.7 must also hold for previous iteration $k-1$ when $k>0$, so

$$\mathsf{v}^n = \mathsf{N}^{k-1}\mathsf{z}^k - \mathsf{B}^{k-1}\mathsf{z}^{k-1} - \mathsf{b} \ . \tag{4.4.17}$$

Equations 4.4.7 and 4.4.17 may be combined to form the equation for the update to z^{k+1}, applying Equation 4.4.16 to introduce Y^k

$$\begin{aligned}\mathsf{N}^k\left(\mathsf{z}^{k+1}-\mathsf{z}^k\right) &= \mathsf{N}^{k-1}\mathsf{z}^k - \mathsf{N}^k\mathsf{z}^k + \mathsf{B}^k\mathsf{z}^k - \mathsf{B}^{k-1}\mathsf{z}^{k-1}\\ &= \left[\left(\mathsf{N}^{k-1}-\mathsf{B}^{k-1}\right)-\left(\mathsf{N}^k-\mathsf{B}^k\right)\right]\mathsf{z}^k + \mathsf{B}^{k-1}\left(\mathsf{z}^k-\mathsf{z}^{k-1}\right)\\ &= \mathsf{Y}^k\mathsf{z}^k + \mathsf{B}^{k-1}\left(\mathsf{z}^k-\mathsf{z}^{k-1}\right) \ .\end{aligned}$$

This may be rewritten to obtain, for $k>0$

$$\begin{aligned}\mathsf{z}^{k+1}-\mathsf{z}^k &= \left(\mathsf{N}^k\right)^{-1}\mathsf{Y}^k\mathsf{z}^k + \left(\mathsf{N}^k\right)^{-1}\mathsf{B}^{k-1}\left(\mathsf{z}^k-\mathsf{z}^{k-1}\right)\\ &= \mathsf{u}^k + \mathsf{W}^k\left(\mathsf{z}^1-\mathsf{z}^0\right)\end{aligned} \tag{4.4.18}$$

where

$$\begin{aligned}\mathsf{u}^k = &\quad \left(\mathsf{N}^k\right)^{-1}\mathsf{Y}^k\mathsf{z}^k\\ &+ \left(\mathsf{N}^k\right)^{-1}\mathsf{B}^{k-1}\left(\left(\mathsf{N}^{k-1}\right)^{-1}\mathsf{Y}^{k-1}\mathsf{z}^{k-1}\right)\\ &\quad \ldots\\ &+ \left(\mathsf{N}^k\right)^{-1}\mathsf{B}^{k-1}\left(\mathsf{N}^{k-1}\right)^{-1}\mathsf{B}^{k-2}\ldots\left(\mathsf{N}^2\right)^{-1}\mathsf{B}^1\left(\left(\mathsf{N}^1\right)^{-1}\mathsf{Y}^1\mathsf{z}^1\right)\end{aligned} \tag{4.4.19}$$

$$\mathsf{W}^k\left(\mathsf{z}^1-\mathsf{z}^0\right) = \left(\mathsf{N}^k\right)^{-1}\mathsf{B}^{k-1}\left(\mathsf{N}^{k-1}\right)^{-1}\mathsf{B}^{k-2}\ldots\left(\mathsf{N}^1\right)^{-1}\mathsf{B}^0\left(\mathsf{z}^1-\mathsf{z}^0\right) \ . \tag{4.4.20}$$

When Equation 4.4.18 is applied recursively back to the first iteration it becomes

$$\mathsf{z}^{k+1} = \mathsf{z}^1 + \sum_{i=1}^{k}\mathsf{u}^i + \sum_{i=1}^{k}\mathsf{W}^i\left(\mathsf{z}^1-\mathsf{z}^0\right) \ . \tag{4.4.21}$$

At this point, the condition restricting $\Delta\tau$ so that constant $0<c_2<1$ in Lemma 4.4.2 Equation 4.4.9 is required. By the M-matrix condition for N^k of Lemma 4.4.2, $\left(\mathsf{N}^k\right)^{-1}\geq 0$. Also $\mathsf{B}^k\geq 0$ so $\mathsf{W}^k\geq 0$. By Lemma 4.4.2

$$\left\|\mathsf{W}^k\right\|_\infty = \left\|\prod_{i=1}^{k}\left(\mathsf{N}^i\right)^{-1}\mathsf{B}^{i-1}\right\|_\infty \leq (c_2)^k \ . \tag{4.4.22}$$

Then, from Equation 4.4.22

$$\left\|\sum_{i=1}^{k}\mathsf{W}^i\right\|_\infty \leq \sum_{i=1}^{k}(c_2)^i < \frac{c_2}{1-c_2} \ .$$

Thus $\left\{\sum_{i=1}^{k} \mathsf{W}^i\right\}_k$ is a non-decreasing series bounded from above with $\left\|\mathsf{W}^k\right\|_\infty \to 0$ as $k \to \infty$. By Equation 4.4.22 and Lemma 4.4.3 that portion of Equation 4.4.21 converges to a finite value

$$\left\|\sum_{i=1}^{k} \mathsf{W}^i \left(\mathsf{z}^1 - \mathsf{z}^0\right)\right\|_\infty \to c_4 < \infty$$

for any z^1 and z^0. By Remark 4.4.4, $\mathsf{Y}^k\mathsf{z}^k \geq 0$. With $\left(\mathsf{N}^k\right)^{-1} \geq 0$ and $\mathsf{B}^k \geq 0$, $\mathsf{u}^k \geq 0$ then the series $\sum_{i=0}^{k} \mathsf{u}^i$ is non-decreasing. By Lemma 4.4.3 the iterates z^k are bounded from above. With $\sum_{i=1}^{k} \mathsf{W}^i \left(\mathsf{z}^1 - \mathsf{z}^0\right)$ convergent and finite, the conclusion is that $\mathsf{u}^k \to 0$ as $k \to \infty$.

Note that if $\mathsf{z}^{k+1} = \mathsf{z}^k$ then z^{k+1} is the solution to Equation 4.3.13. Thus the iteration converges to a limit which, by Theorem 5.1 of [47], is the unique solution to Equation 4.3.13.

□

The condition for Theorem 4.4.5 that $c_2 < 1$ in Equations 4.4.9 and 4.4.22 is relatively mild. It is always satisfied, for example, if $\Delta\tau\, \lambda_{\max} \leq 1$. Convergence can be proved with no time step condition by forcing $\left(\mathsf{z}^1 - \mathsf{z}^0\right) \geq 0$ since, in this case, $\sum_{i=1}^{k} \mathsf{W}^i \left(\mathsf{z}^1 - \mathsf{z}^0\right)$ is non-decreasing. Provided $\mathsf{v}^n \geq 0$ and the boundary conditions force $\mathsf{v}^{n+1} \geq 0$ at the boundary, setting $\mathsf{z}^0 = 0$ forces $\left(\mathsf{z}^1 - \mathsf{z}^0\right) \geq 0$. However, as observed in numerical testing, this approach always incurs the expense of an extra fixed point policy iteration to converge the solution to the required tolerance. Note that the condition $c_2 < 1$ is sufficient and does not appear to be necessary. In many numerical experiments conducted with $c_2 \geq 1$ and $\mathsf{z}^0 = \mathsf{v}^n$, the iteration was always convergent.

4.4.3 Reducing Computational Cost

When a correlation $\mathsf{J}(\mathsf{e})\mathsf{v}^k$ is performed over a rectangular grid with constant spacing for a single, constant policy $\mathsf{e} \in \widehat{Y}$, then the computation can be performed by applying the approach of Section 2.4.3. The FFT approach can be exploited to form the product $\mathsf{J}(\mathsf{q}^k)\mathsf{v}^k$ in Equation 4.4.3 by evaluating each of the N_Y constant policies individually. An FFT solution generates all N points together for a given constant policy e. Once q^k is determined, then the $\mathsf{J}(\mathsf{q}^k)\mathsf{v}^k$ components which are selected can be taken from the set of N_Y constant policy solutions.

To use the FFT approach, note that the solution to Equation 4.4.3 may be written for fixed point policy iterate z^k as finding $\mathsf{R}^k(\mathsf{e})\ \forall\ \mathsf{e} \in \widehat{Y}$ such that for q^k in Equation 4.4.5

$$\left[\mathsf{G}(\mathsf{q}^k) + \mathsf{I_c}\left(\mathsf{J}(\mathsf{q}^k) - \mathsf{D}(\mathsf{q}^k) - \mathsf{P}(\mathsf{q}^k)\right)\right]\mathsf{z}^k = \sum_{\mathsf{e}\in\widehat{Y}}\left\{\mathsf{R}^k(\mathsf{e})\left[\mathsf{G}(\mathsf{e}) + \mathsf{I_c}\left(\mathsf{J}(\mathsf{e}) - \mathsf{D}(\mathsf{e}) - \mathsf{P}(\mathsf{e})\right)\right]\mathsf{z}^k\right\}. \tag{4.4.23}$$

Matrices $\mathsf{G}(\mathsf{e})$ and $[\mathsf{J}(\mathsf{e}) - \mathsf{D}(\mathsf{e}) - \mathsf{P}(\mathsf{e})]$ represent the discrete operators when a single constant policy $\mathsf{e} \in \widehat{Y}$ is applied over the entire grid. Matrix $\mathsf{R}^k(\mathsf{e}) \in \mathbb{R}^{N\times N}$ selects the nodes at iteration k for which a particular policy is applied to maximize each row so that

$$\left[\mathsf{R}^k(\mathsf{e})\right]_{\mathsf{ij}} = \begin{cases} 0 \text{ or } 1 & \mathsf{i} = \mathsf{j} \\ 0 & \mathsf{i} \neq \mathsf{j} \end{cases} \quad \text{and} \quad \sum_{\mathsf{e}\in\widehat{Y}} \mathsf{R}^k(\mathsf{e}) = \mathsf{I}\ . \tag{4.4.24}$$

Note that the solution of Equation 4.4.23 must be done with the same drift term discretization (i.e. central, forward or backward weighting) used in the iteration Equation 4.4.4 or convergence is not guaranteed [47] §4.3,[105].

To use the FFT approach, an interpolation of the original FD grid of values must again be made onto the rectangular, constant spacing, DFT grid which overlaps Ω_D. With the interpolation onto the DFT grid, the actual computation for a single, constant policy e becomes

$$\mathsf{J}(\mathsf{e})\,\mathsf{v} \simeq (\mathsf{K}\cdot\mathsf{F}(\mathsf{e})\cdot\mathsf{L})\,\mathsf{v} \tag{4.4.25}$$

where $\mathsf{F}(\mathsf{e})$ computes the correlation operation corresponding to $\mathsf{J}(\mathsf{e})$ using the FFT approach. Matrices $\mathsf{K} \geq 0$ and $\mathsf{L} \geq 0$ are the same second order accurate, monotone, linear interpolations between the DFT grid and the FD grid as were used in previous chapters

Actually, fewer than $\mathrm{O}(N_Y \cdot N\log(N))$ operations are required to compute $\mathsf{J}(\mathsf{q}^k)\mathsf{v}^k$. Observe that some $\lambda(\mathsf{e})$ multiplies each line of the matrix

$$\Big(\mathsf{J}(\mathsf{q}^k) - \mathsf{D}(\mathsf{q}^k) - \mathsf{P}(\mathsf{q}^k)\Big) = \sum_{\mathsf{e}\in\widehat{Y}} \mathsf{R}^k(\mathsf{e})\,[\mathsf{K}\cdot\mathsf{F}(\mathsf{e})\cdot\mathsf{L} - \mathsf{D}(\mathsf{e}) - \mathsf{P}(\mathsf{e})]$$

and hence that factor may be taken outside the discrete integral operator when forming $\mathsf{R}^k(\mathsf{e})$.

Also, two constant policies $E, F \in \widehat{Y}$ which differ by a shift in the mean of the distribution may be evaluated with only one application of the FFT approach. If the jump measure for policy F is identical to that of E to within a shift (z_1, z_2) in the mean where $g(J_1, J_2; F) = g(J_1 - z_1, J_2 - z_2; E)$ then for

$$\begin{aligned}\mathcal{J}(F)\,U(x_1, x_2, \tau) &= \iint_{-\infty}^{\infty} g(J_1, J_2; F)\,U(x_1 + J_1, x_2 + J_2, \tau)\;dJ_1\;dJ_2 \\ &= \iint_{-\infty}^{\infty} g(J_1 - z_1, J_2 - z_2; E)\,U(x_1 + J_1, x_2 + J_2, \tau)\;dJ_1\;dJ_2 \\ &= \iint_{-\infty}^{\infty} g(J_1', J_2'; E)\,U(x_1 + J_1' + z_1, x_2 + J_2' + z_2, \tau)\;dJ_1'\;dJ_2' \\ &= \mathcal{J}(E)\,U(x_1 + z_1, x_2 + z_2, \tau)\ .\end{aligned} \tag{4.4.26}$$

The correlation result for policy F can be computed for point (x_1, x_2) by re-using the correlation result for E and taking the value at the point (x_1+z_1, x_2+z_2). This result holds for the periodic, discrete case as well to within $\mathrm{O}(h^2)$ where h controls the DFT grid spacing, provided that Ω_D has been constructed so that (z_1, z_2) is small relative to its size. This will be the case if the domain sizing approach of Sections 2.3.3 and 4.3 has been employed.

Thus, for the particular case in question where $g(J_1, J_2; \mathsf{e})$ is defined by Equation 4.2.9 for a constant policy e, $\lambda(\mathsf{e})$ may be factored out of g to make it a conventional bi-variate Normal distribution with FFT evaluation operation $\mathsf{F}^*(\mathsf{e})$. Only a new value of the triplet $(\check{\sigma}_1, \check{\sigma}_2, \check{\rho})$ for a policy e triggers a need to re-evaluate $\mathsf{F}^*(\mathsf{e})\cdot\mathsf{L}\,\mathsf{z}^k$ with the $\mathrm{O}(N\log(N))$ FFT procedure. If constant policy e and constant policy f differ only in $(\lambda, \check{\mu}_1, \check{\mu}_2)$ then only an $\mathrm{O}(N)$ operation is required to re-evaluate

$$\Big[\mathsf{G}(\mathsf{f}) + \lambda(\mathsf{f})\mathsf{I_c}\big[\mathsf{K}(\mathsf{f})\cdot\mathsf{F}^*(\mathsf{e})\cdot\mathsf{L} - \mathsf{I_c}\mathsf{D}(\mathsf{f}) - \mathsf{I_c}\mathsf{P}(\mathsf{f})\big]\Big]\mathsf{z}^k$$

by re-using $\mathsf{F}^*(\mathsf{e}) \cdot \mathsf{L}\,\mathsf{z}^k$ with the matrices $\mathsf{G}(\mathsf{f})$, $\mathsf{K}(\mathsf{f})$, $\mathsf{D}(\mathsf{f})$ and $\mathsf{P}(\mathsf{f})$. Considerable computational effort can be saved by grouping the evaluation of all policies in Equation 4.4.23 which have common $(\breve{\sigma}_1, \breve{\sigma}_2, \breve{\rho})$ values.

Applying this to the first step of the piecewise constant policy time step approach could save work, but on the first iteration only. In that case it could be used to evaluate the N_Y linear systems where $\mathsf{z}^0 = \mathsf{v}^n$ for all cases.

4.4.4 Algorithm Cost Comparison

Recall from Section 4.3.2 that $N_Y = 2{N_q}^5$ for the problem under investigation where N_q is the number of grid points, spaced $\mathrm{O}(h_y)$ apart, in each of the discretized control parameter dimensions except that of $\lambda(Q)$. Although $N_q < N_Y \ll N$ in the numerical tests which follow, it is also useful to express the cost of the algorithms in terms of a general discretization parameter h where $\Delta x_1, \Delta x_2, \Delta\tau, h_y = \mathrm{O}(h)$ such that

$$N = \mathrm{O}\left(\left[\frac{1}{h}\right]^2\right) \quad \text{and} \quad N_Y = \mathrm{O}\left(\left[\frac{1}{h}\right]^5\right) \ .$$

A single piecewise constant policy time step (Equations 4.4.1 and 4.4.2) requires N_Y linear system solutions, each of which may be done using M_f fixed point iterations. Thus, over N finite difference grid nodes, for each of $N_Y = 2{N_q}^5$ policies, for M_f iterations, an $\mathrm{O}(N^{5/4})$ sparse linear solution and two $\mathrm{O}(N \log(N))$ FFT's are required. The cost per piecewise constant policy time step is

$$C_{\text{pwcp}} = \mathrm{O}\Big(2{N_q}^5 \cdot M_f \cdot \Big[N^{5/4} + N\log(N)\Big]\Big) = \mathrm{O}\left(M_f \cdot \left[\frac{1}{h}\right]^{15/2}\right) \ . \tag{4.4.27}$$

The cost saving measures of Section 4.4.3 split the cost of evaluating N_Y policies into ${N_q}^3$ FFT evaluations for the unique triplets of $(\breve{\sigma}_1, \breve{\sigma}_2, \breve{\rho})$ parameters and $2{N_q}^5$ linear system evaluations for all unique elements of $\widehat{Y}$. In the fixed point policy iteration, this means the effort is reduced, for one time step, to M_p iterations of the policy evaluation and M_p sparse linear system solutions. Thus the cost per time step using a fixed point policy iteration is

$$C_{\text{fppi}} = \mathrm{O}\Big(M_p \cdot \Big[2{N_q}^5 \cdot N + {N_q}^3 \cdot N\log(N) + N^{5/4}\Big]\Big) = \mathrm{O}\left(M_p \cdot \left[\frac{1}{h}\right]^{7}\right) \ . \tag{4.4.28}$$

Assuming $M_f \simeq M_p$, then the complexity of the fixed point policy iteration computed by Equation 4.4.28 is lower than that of the piecewise constant policy method given by Equation 4.4.27. The fact that $N_Y \ll N$ under the demonstration scenarios which follow makes this a pronounced savings.

4.4.5 American Options by Penalty Iteration

The penalty method of Section 2.6.2, when used with the fixed point policy iteration, does not need to be separate. The penalty vector for iterate z^k is evaluated before iteration update

Algorithm 4.4.1 Solve one time step using a combined fixed point policy and penalty iteration.

```
FixedPointPolicyIteration( $\mathsf{v}^n$, $\mathsf{v}^*$, $\Delta\tau$, $\mathsf{G(e)}$, $\mathsf{J(e)}$, $\mathsf{D(e)}$, $\mathsf{P(e)}$, $\mathsf{b}$, $\epsilon_u$ )
where
        $\mathsf{v}^n$                      the price at time step $n$
        $\mathsf{v}^*$                      the minimum option value (usually the payoff)
        $\Delta\tau$                        time step size
        $\mathsf{G(e)}$                     discrete Brownian motion differential terms and
        $\mathsf{J(e), D(e), P(e)}$         discrete integral operator components, policies $\mathsf{e} \in \widehat{Y}$.
        $\mathsf{b}$                        boundary condition imposition vector
        $\epsilon_u$                        required solution update tolerance

1.  Set $\mathsf{z}^{k=0} = \mathsf{v}^n$.

2.  For $k = 0, 1, 2, 3, \ldots$ until convergence (tested in Step 7)

3.      For American options: set $\mathsf{c}^k$ using Equation 2.6.3.
        For European options: set $\mathsf{c}^k = 0$.

4.      Determine $\mathsf{q}^k$ and matrices $\mathsf{G(q}^k)$,$\mathsf{J(q}^k)$,$\mathsf{D(q}^k)$ and $\mathsf{P(q}^k)$ such that
            $\Big[\mathsf{G(q}^k) + \mathsf{I_c}\left(\mathsf{J(q}^k) - \mathsf{D(q}^k) - \mathsf{P(q}^k)\right)\Big]\mathsf{z}^k$
            $= \max_{\mathsf{q}\in\widehat{Y}}\Big\{\Big[\mathsf{G(q)} + \mathsf{I_c}\left[\mathsf{J(q)} - \mathsf{D(q)} - \mathsf{P(q)}\right]\Big]\mathsf{z}^k\Big\}$
        using the approach of Section 4.4.3 and $\mathsf{G(e)}$, $\mathsf{J(e)}$, $\mathsf{D(e)}$, $\mathsf{P(e)}$, $\forall\ \mathsf{e} \in \widehat{Y}$.

5.      Set  $\mathsf{N}^k = \Big[\mathsf{I} - \Delta\tau\ \left(\mathsf{G(q}^k) - r\,\mathsf{I} - \mathsf{I_c}\left[\mathsf{D(q}^k) + \mathsf{P(q}^k)\right]\right)\Big] + (\mathsf{c}^k)'\mathsf{I}$
             $\mathsf{y}^k = \left[\Delta\tau\ \mathsf{I_c}\mathsf{J(q}^k)\right]\mathsf{z}^k + \mathsf{v}^n + \mathsf{b} + (\mathsf{c}^k)'\mathsf{I}\,\mathsf{v}^*$

6.      Solve  $\mathsf{N}^k\,\mathsf{z}^{k+1} = \mathsf{y}^k$  using ILU(2) preconditioned Bi-CGStab.

7.      If $\max_{\mathsf{i}}\left[\dfrac{\left|\mathsf{z}^{k+1} - \mathsf{z}^k\right|_{\mathsf{i}}}{\max\left(1, |\mathsf{z}^{k+1}|_{\mathsf{i}}, |\mathsf{z}^k|_{\mathsf{i}}\right)}\right] < \epsilon_u$ then the iteration is finished.

    End For

Return the solution vector $\mathsf{v}^{n+1} = \mathsf{z}^{k+1}$.
```

Equation 4.4.4. Equation 4.4.4 is modified by adding the term $\left(\mathsf{c}^k\right)' \,\mathsf{I}\left(\mathsf{v}^* - \mathsf{z}^{k+1}\right)$ to the RHS to form

$$\begin{aligned}\Big\{\mathsf{I} - \Delta\tau \left[\mathsf{G}(\mathsf{q}^k) - r\,\mathsf{I} - \mathsf{I_c}\left(\mathsf{D}(\mathsf{q}^k) + \mathsf{P}(\mathsf{q}^k)\right)\right] &+ \left(\mathsf{c}^k\right)'\mathsf{I}\Big\}\,\mathsf{z}^{k+1} \\ &= \Delta\tau\,\mathsf{I_c}\mathsf{J}(\mathsf{q}^k)\,\mathsf{z}^k + \left(\mathsf{c}^k\right)'\mathsf{I}\,\mathsf{v}^* + \mathsf{v}^n + \mathsf{b}\ . \qquad (4.4.29)\end{aligned}$$

Algorithm 4.4.1 is the combination of both iterations.

Theorem 4.4.6 ***Convergence of the Combined Penalty, Fixed Point Policy Iteration.*** *Let the preconditions of Lemma 4.4.2 hold and let Equation 4.4.9 satisfy* $0 < c_2 < 1$. *Then the combined penalty, fixed point policy iteration Equation 4.4.29 is convergent to the unique solution of the following time step equation for American options (equivalent to Equation 4.3.13 for European options)*

$$\left[\mathsf{I} + \Delta\tau\ \mathsf{T}(\mathsf{q}^{n+1}) + \left(\mathsf{c}^{n+1}\right)'\mathsf{I}\right]\mathsf{v}^{n+1} = \mathsf{v}^n + \left(\mathsf{c}^{n+1}\right)'\mathsf{I}\,\mathsf{v}^* + \mathsf{b}$$

where c^{n+1} *is the penalty vector at time* τ_{n+1}.

Proof. The proof follows by creating an additional control $\beta(x_1, x_2, \tau) \in \{0, 1\}$ to form a penalty term added to Equation 4.2.1. This leads to an HJB-style PIDE for American options [47] which is equivalent to the linear complementarity form of Equations 4.2.7 and 4.2.8

$$\begin{aligned}\frac{\partial U}{\partial \tau} &= \mathcal{G}U - rU + \max_{Q \in \widehat{Q}}\big\{\mathcal{H}(Q)U\big\} + \max_{\beta \in \{0,1\}}\left\{\beta\,\frac{1}{\epsilon}\cdot(\mathcal{I} - U)\right\}\ , \quad \epsilon \ll 1 \qquad (4.4.30)\\ U(x_1, x_2, 0) &= \mathcal{I}(x_1, x_2)\ .\end{aligned}$$

Localization and discretization of Equation 4.4.30 leads to the terms of iteration Equation 4.4.29. The proof of convergence of the penalty iteration, combined with fixed point policy iteration, proceeds using the same steps as in Section 4.4.2 above. Remark 4.4.4 changes so that matrix Y^k is formed to include penalty term

$$\left(\mathsf{c}^k\right)'\,\mathsf{I}\left(\mathsf{v}^* - \mathsf{z}^{k+1}\right)$$

and $\mathsf{Y}^k\mathsf{z}^k \geq 0$ still holds [37] §4. Then the remaining steps of the proof proceed in nearly identical fashion to those of Theorem 4.4.5. □

Amadori (see [3] § 4.2) investigates the conditions for convergence of the penalty method to the viscosity solution of an American option under constant coefficients. The combined method presented here is consistent, monotone and stable, thus is assumed to converge to the viscosity solution of the localized problem for uncertain coefficients [18, 19, 91]. However, this thesis does not undertake to prove this assumption.

	λ		$\breve{\mu}$		**Data**	**Source of**
Study	Min	Max	Min	Max	**Source**	**Variation**
Bakshi & al. 1997 [8]	0.57	0.70	-0.090	-0.040	S&P 500	Fit error, data weights.
Eraker & al. 2003 [44]	0.98	2.04	-0.038	-0.014	S&P 500	Fit error, 1 std. dev.
Eraker & al. 2003 [44]	1.89	6.78	-0.035	-0.014	Nasdaq	Fit error, 1 std. dev.
Eraker 2004 [43]	0.25	0.75	-0.037	0.031	S&P 500	Fit error, 1 std. dev.
Bakshi & al. 2003 [24]	2.56	3.27	-0.050	-0.010	S&P 500	Choice of data set.

Table 4.5.1: Jump parameters λ and $\breve{\mu}$ ranges from stochastic volatility with jump studies, where the jumps are in price only and are Normally distributed with mean $\breve{\mu}$. These ranges are taken as indicative, but would likely be larger in a constant volatility study since the jump process would have to account for more of the variation in the non-constant implied volatility. The data source and the main source of variation is identified for each case.

4.5 Numerical Demonstrations

The first step of the numerical demonstrations in this chapter is to construct a reasonable synthetic market which describes the evolution of two correlated assets. This is done in Section 4.5.1. Over this market, a European call option on the maximum of the two assets and an American put option on the minimum of the two assets are valued. The convergence and cost of the approach is demonstrated in Section 4.5.2. Finally, the sensitivity of the best- and worst-case prices to the underlying parameters is reviewed in Section 4.5.3.

4.5.1 Contracts and Synthetic Market

To derive a reasonable synthetic market, the results of He et al. [59] serve as a starting point. In that study, a Merton one-asset model [86] with Normally distributed jumps and diffusion was fit to several option price sets. These sets were computed using values from a known process to which 4.7% to 6% noise was added. In that case, the calibrated jump rate was $\lambda = 0.06$ to 0.15 around the known value of $\lambda = 0.1$ and the mean jump ranged from $\breve{\mu} = -0.0178$ to -0.0061 around the known value of $\breve{\mu} = -0.0092$. Interestingly, in each case the jump volatility estimates were nearly exactly the known value $\breve{\sigma} = 0.20$. This chapter will follow this indication and fix the values of the jump standard deviations $(\breve{\sigma}_1, \breve{\sigma}_2)$ to constant values.

Available, recent, calibration studies from actual markets use the stochastic volatility model with jumps. The calibration ranges for λ and $\breve{\mu}$ from several studies are listed in Table 4.5.1 for stochastic volatility with Normal price jumps. These will be taken as indicative, although not all of the values are in the risk-neutral measure. In a constant Brownian motion setting the jump parameter values could be expected to take on higher values to reflect the absence of stochastic volatility as a source of variation in the implied volatility.

Based approximately on these ranges, Table 4.5.2 lists the parameters of the Brownian motion and jump components (Equation 4.2.3) of the synthetic price process used in the following

		Fixed	Uncertain	
	Parameter	Value	Min	Max
Brownian Motion	σ_1	0.15		
	σ_2	0.12		
	ρ	0.30		
Finite Activity Jumps	λ		1.50	2.00
	$\breve{\mu}_1$		-0.04	0.00
	$\breve{\mu}_2$		-0.02	0.02
	$\breve{\sigma}_1$	0.12		
	$\breve{\sigma}_2$	0.16		
	$\breve{\rho}$		-0.80	0.80

Table 4.5.2: Price process parameters for the numerical demonstration. Four jump parameters are uncertain. All contracts were solved with a risk-free rate $r = 0.05$, strike $K = 100$ and expiry time $T = 1$.

numerical examples. Jump parameter $\breve{\rho}$ is selected to allow both a strong positive and a strong negative correlation. This corresponds to a case where two assets are related and an important news event may affect either both together or one in favour of the other (e.g. a duopoly where an industry wide shock may occur, or an event favouring one firm over the other). The jump arrival rate λ, the mean of the jumps on both axes $(\breve{\mu}_1, \breve{\mu}_2)$ and the jump correlation $\breve{\rho}$ are uncertain, which supplies four parameter dimensions to discretize. The total asset price variance due to Brownian motion in this case is somewhat lower than the variance due to jumps.

4.5.2 Numerical Convergence

All contracts were solved with a risk-free rate $r = 0.05$, strike $K = 100$ and expiry time $T = 1$. For all tests, $\Omega_C = [35, 300] \times [35, 300]$ and $\Omega_D = [8, 1240] \times [8, 1240]$ in price scaling (S_1, S_2). These satisfy the Ω_C and Ω_D sizes around a region of interest $[80, 125] \times [80, 125]$ using domain size tolerances $\epsilon_c = 10^{-3}$ and $\epsilon_d = 10^{-4}$, which were the values used in Chapter 2.

To verify that these finite domains were large enough, the numerical tests which follow were repeated with a much larger $\Omega_D = [4, 2357] \times [4, 2357]$ using the second of the three grid resolutions. This produced no difference in the values to the accuracy reported below. With a larger $\Omega_C = [25, 400] \times [25, 400]$ as well as larger $\Omega_D = [4, 2357] \times [4, 2357]$, the best-case European option value differed by an relative value of 4×10^{-5}. In all other cases the relative difference was less than 1×10^{-5}. For both tests on larger domains, the error due to domain truncation appears to be much less than the discretization error on the finest grid used for tests.

To demonstrate convergence, solutions on three levels of refinement were performed. For each refinement, the spacing of the grid of uncertain parameters, the FD and DFT grid spacing and the time step were divided by two. The coarsest FD grid was formed with a spacing near the strike point $(\log(K), \log(K))$ of $(\Delta x_1, \Delta x_2) = (0.04, 0.04)$ and was then rotated using

Equation 2.5.4 around the strike by $\theta_r = 26.5651$ degrees before truncation at the upper and lower (x_1, x_2) boundaries. This grid is shown in Figure 4.3.1. The DFT grid remained aligned with the (x_1, x_2) axes and for the coarse grid had a spacing of $(\Delta x_{f,1}, \Delta x_{f,2}) = (0.08, 0.08)$. The time step used with the coarsest grid was $\Delta\tau = 0.04$.

The parameter grid $\widehat{Y}$ Equation 4.3.16 has, for the parameters of Table 4.5.2, four dimensions. Each dimension other than λ was discretized by $N_q = 3$ points in each direction for the first, coarse grid computation, which generated $N_Y = 2{N_q}^3 = 54$ discrete policies. On the finest grid, N_Y increased to 1458. Since $\check{\sigma}_1, \check{\sigma}_2$ were constant, Equation 4.4.28 for the cost of the fixed point policy iteration becomes

$$C_{\text{fppi}} = \mathrm{O}\Big(M \cdot \Big[2{N_q}^3 \cdot N + N_q \cdot N \log(N) + N^{5/4}\Big]\Big) = \mathrm{O}\left(M \cdot \left[\frac{1}{h}\right]^5\right) \tag{4.5.1}$$

for each time step.

A time step was considered converged when a fixed point policy iteration or combined penalty, fixed point policy iteration produced a relative update smaller than $\epsilon_u = 10^{-7}$ (Algorithm 4.4.1 Step 7). A linear system was considered solved when the preconditioned Bi-CGStab algorithm produced a relative update smaller than $\epsilon_l = 10^{-8}$. Tighter tolerances produced no change in the option value within the accuracy reported.

The values for the option contracts are listed in Table 4.5.3 for each grid refinement 1 through 3. The convergence rate α is given assuming that the error is $\mathrm{O}((\Delta x_1)^\alpha, (\Delta x_2)^\alpha, (\Delta\tau)^\alpha, (h_y)^\alpha)$. By the second grid, the solutions appear to be accurate to within 1% of the option value. The error at all resolutions is much less than the difference between the best- and worst-case for each contract.

Figure 4.5.1 shows the best-case option value surface for the American option at $\tau = T$ for the finest grid. The surface has dark shading where $\lambda(Q) = \lambda_{\min}$ and light where $\lambda(Q) = \lambda_{\max}$. Figure 4.5.2 shows the difference between the best- and worst-case European option on the finest grid at $\tau = T$. The surface is shaded from dark to light by the value of $\check{\mu}_1$, from minimum to maximum, which is in force at each grid point for the best-case policy.

Numerical runs were also repeated using the "no bang-bang λ" policy set $\widehat{Y}_\lambda$ of Equation 4.3.17, where the λ values were also refined between $\lambda_{\min}$ and $\lambda_{\max}$. Even if the exact solution has bang-bang type controls, there is no guarantee that the controls will be bang-bang on a finite grid. In [27], a no-bang-bang discretization converged more smoothly than a bang-bang discretization, even though the converged solution had a bang-bang control. The solutions using $\widehat{Y}_\lambda$ are listed in the lower half of Table 4.5.3. The policies with intermediate values of λ occurred infrequently, only between larger patches where $\lambda(Q) = \lambda_{\min}$ or $\lambda(Q) = \lambda_{\max}$ and only away from the core of the solution at points, where the grid spacing changed. In the interior of the solution $(S_1, S_2) \in [80, 125] \times [80, 125]$, the solution under $Q \in \widehat{Y}$ versus $Q \in \widehat{Y}_\lambda$ differed by less than an absolute value of 1×10^{-4} on the coarsest grid, less than 1×10^{-6} on the first refinement and less 1×10^{-8} on the finest grid. This agrees with the assertion in Section 4.3.2 that $\lambda(Q)$ is a "bang-bang" constraint and thus only $\lambda_{\min}, \lambda_{\max}$ are required in the discrete policy grid.

$Q \in \widehat{Y}$**: "bang-bang λ"**

	$V(S_1 = 100, S_2 = 100)$			
	European		**American**	
Grid	Call on max()		Put on min()	
Refinement	Worst	Best	Worst	Best
1	16.2688	22.6877	9.3775	13.8786
2	15.8978	22.2701	9.2088	13.7263
3	15.8097	22.1835	9.1748	13.7020
α	2.1	2.3	2.3	2.7
$Q \in \widehat{Y}_\lambda$**: "no bang-bang λ"**				
1	16.2688	22.6877	9.3775	13.8786
2	15.8978	22.2701	9.2088	13.7263
3	15.8097	22.1835	9.1748	13.7020

Table 4.5.3: At each refinement, the best- and worst-case option values for a European call on the maximum and American put on the minimum of two assets are given for $(S_1, S_2) = (100.0, 100.0)$ at $\tau = 1$. The approximate convergence rate α is given, assuming the error is $\mathrm{O}((\Delta x_1)^\alpha, (\Delta x_2)^\alpha, (\Delta\tau)^\alpha, (h_y)^\alpha)$. The top section of the table lists results for $Q \in \widehat{Y}$ Equation 4.3.16 where $\lambda(Q)$ is treated as a "bang-bang" constraint. The bottom section lists the results where $Q \in \widehat{Y}_\lambda$ Equation 4.3.17 where $\lambda(Q)$ is discretized, which resulted in no difference in the results within the accuracy reported.

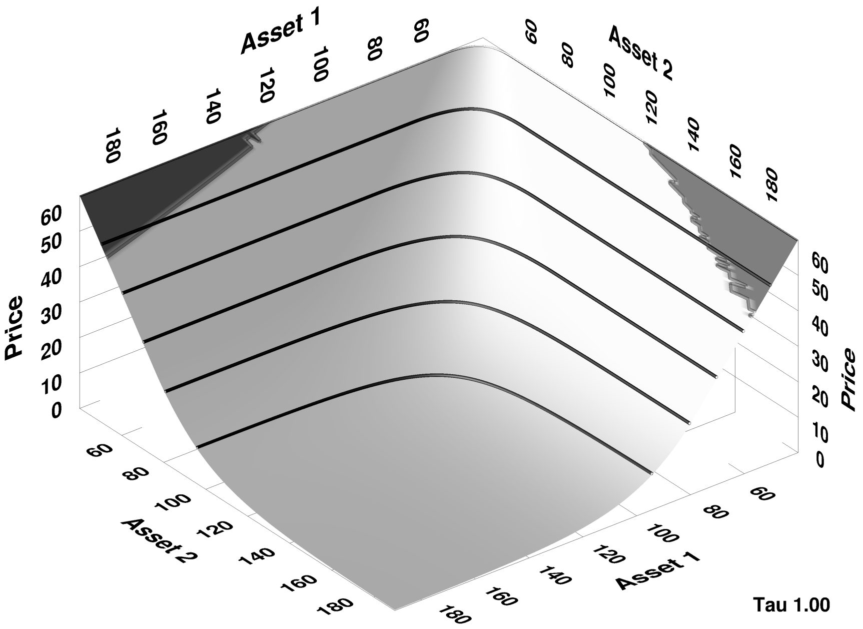

Figure 4.5.1: This figure shows the best-case option value for the American put on the minimum of two assets under the uncertain parameters given in Table 4.5.2. The result is from the finest grid resolution. The surface is shaded by the value of $\lambda(Q)$ in force at $\tau = 1$: $\lambda(Q) = \lambda_{\text{min}}$ where the surface is dark and $\lambda(Q) = \lambda_{\text{max}}$ where the surface is light.

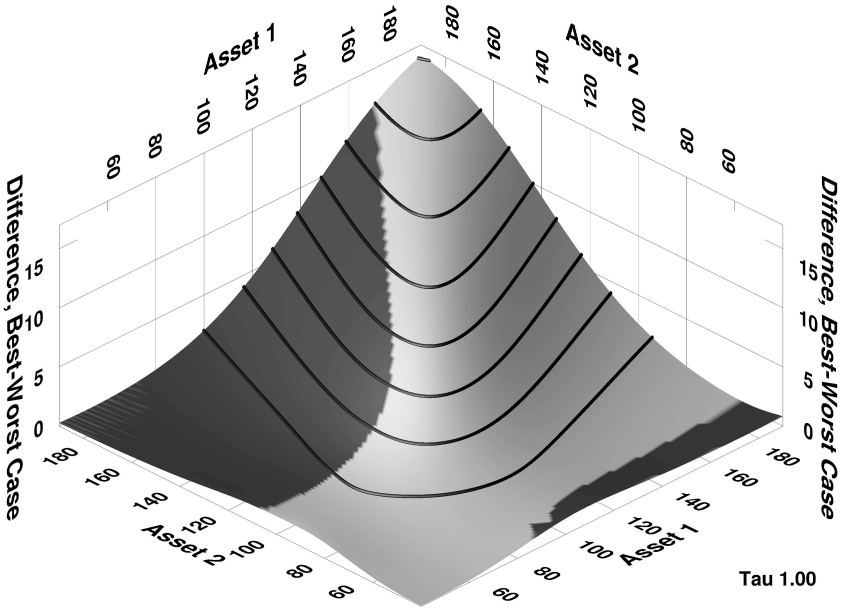

Figure 4.5.2: This shows the difference between the best-case and the worst-case option value for the European call on the maximum of two assets. The result is from the finest grid resolution. The surface is shaded from dark to light by the value of $\check{\mu}_1$, minimum to maximum, which is in force at $\tau = 1$ from the best-case policy.

European Call on max()**, Best-Case**

	Time	FD		Iterations				
Grid	Steps	Nodes	Policies	FPP		Bi-CGStab		CPU
Refinement	N_τ	N	N_Y	Total	/TS	Total	/FPP	Time
1	25	2399	54	161	6.4	483	3.0	1
2	50	8885	250	269	5.4	877	3.3	29
3	100	34091	1458	450	4.5	1800	4.0	1107

American Put on min()**, Best-Case**

1	25	2399	54	159	6.4	477	3.0	1
2	50	8885	250	267	5.3	801	3.0	29
3	100	34091	1458	449	4.5	1796	4.0	1110

Table 4.5.4: The number of time steps N_τ, finite difference nodes N and the number of policies N_Y, is given for each grid refinement. For the best-case option value, for the European call on the maximum and American put on the minimum of two assets, the number of outer iterations is given, total and per time step. The outer iterations are either fixed point policy (FPP) iterations in the European case or combined penalty iterations for the American case. The number of Bi-CGStab solver steps is given along with the average per outer, time step iteration. The CPU time is given with cost of the coarse grid solution for the European case normalized to 1.0.

Table 4.5.4 lists the number of time steps N_τ, FD grid nodes N and the size of the set of discrete policies at each grid refinement N_Y for two contracts. It shows that, for the best-case price of the European option, an average of 6.4 fixed point policy iterations were required to converge each time step on the coarsest grid. On the finest grid this was reduced to 4.5, which implies that the $\lambda_{\max}\Delta\tau$ factor in the proof of Theorem 4.4.5 does indeed influence the rate of convergence. The worst-case European price required only 6 fixed point policy iterations for each time step on the coarsest grid and 4.5 on the finest. The American options required almost exactly the same number of combined penalty, fixed point policy iterations. Each linear system (Step 6 of Algorithm 4.4.1) required, on average, 3 Bi-CGStab iterations on the coarsest grid and 4 on the finest grid in all cases.

The normalized CPU time in Table 4.5.4 increases, approximately, by the expected value of 32 with each refinement. The bulk of the computational time for the all grids was taken up by Step 4 of Algorithm 4.4.1: the determination of the policies over (x_1, x_2) which select the best- or worst-case option value given iterate z^k. The policy selection step could readily exploit parallel computation techniques.

	$V(S_1 = 100, S_2 = 100)$			
	European		**American**	
Uncertain	Call on max()		Put on min()	
Parameters	Worst	Best	Worst	Best
λ	18.38	20.00	10.68	12.08
$(\breve{\mu}_1, \breve{\mu}_2)$	18.57	19.68	11.07	11.76
$\breve{\rho}$	16.97	20.83	9.98	12.46
$\lambda, (\breve{\mu}_1, \breve{\mu}_2)$	17.80	20.51	10.41	12.47
$\lambda, \breve{\rho}$	16.20	21.50	9.36	13.19
$(\breve{\mu}_1, \breve{\mu}_2), \breve{\rho}$	16.42	21.29	9.71	12.90
$\lambda, (\breve{\mu}_1, \breve{\mu}_2), \breve{\rho}$	15.90	22.27	9.21	13.73

Table 4.5.5: Best- and worst-case option values are supplied where different sets of parameters are allowed to be uncertain over the range given in Table 4.5.2. The parameters not listed in the first column above are fixed to the mid-point of their range. The final line repeats the values for all parameters uncertain. Values are from grid resolution 2 and are approximately 1% accurate.

4.5.3 Sensitivity to Parameters

Six tests were performed to discover which of λ, $(\breve{\mu}_1, \breve{\mu}_2)$ or $\breve{\rho}$ had the greatest effect on option value uncertainty. Each was computed at the second grid resolution, thus with approximately 1% accuracy in price. The results for the two contracts are summarized in Table 4.5.5; each row lists the uncertain parameters in each test and, for ease of comparison, the last row repeats the results for the full set of uncertain parameters. The two jump mean parameters were always either both uncertain or both fixed. Fixed parameters were set to the mid-point of their range in Table 4.5.2.

Of the cases where one of λ, $(\breve{\mu}_1, \breve{\mu}_2)$ or $\breve{\rho}$ was uncertain, listed on the first three lines of Table 4.5.5, the correlation $\breve{\rho}$ produced the widest gap between worst- and best-case values as well as the highest best-case values and lowest worst-case values. When only one parameter was fixed, the widest ranges were also produced when $\breve{\rho}$ was allowed to be uncertain. These results suggest that, for the synthetic market used in this chapter, the correlation $\breve{\rho}$ was the parameter which most influenced the range of results, followed by the jump rate λ.

4.6 Chapter Summary

The localization and discretization of the linear, two-asset PIDE of Chapter 2 has been adapted to the case of a non-linear, HJB-type PIDE for the solution of the best-case and worst-case option value under a deterministic range of parameters. The control variable space was discretized using a standard approach. The spatial discretization was performed using a rotated grid. This was shown in Chapter 3 to be the best from three choices of approach which generate a monotone discretization. The boundary approximation of Chapter 3 was also used, where a axis-aligned line of nodes are place along each boundary.

A new algorithm was presented, the fixed point policy iteration, for the solution of the algebraic equations required for a single time step. It combines implicit imposition of policy with the fixed point algorithm of Chapter 2. It was proved to be globally convergent provided a very mild time step restriction is imposed. An approach was presented that significantly reduces the cost of the most computationally expensive part of the algorithm: determining the policy which minimizes or maximizes the result for each iteration.

Numerical demonstrations, based on a synthetic, but realistic, two-asset market, show the cost and efficiency of the approach. Convergence of the numerical solution with respect to a grid spacing parameter was demonstrated to match the linear rate expected for smooth problems. The localization approach of Chapter 2 is shown to be effective for this case as well. The uncertain parameters generate a best-case to worst-case spread which is comparable to, or even exceeds, the bid and ask spread seen in a typical option market written on equities. For the synthetic market example presented, the correlation was the parameter whose uncertainty generated the widest best-case, worst-case difference.

Chapter 5

Conclusions

The PIDE's at the focus of this thesis arise from problems in the financial option markets. This work has focused on creating accurate and efficient pricing algorithms for three closely related, fundamental valuation problems. The algorithms could, in turn, lead to practical improvements in the ability of financial institutions to offer more sophisticated contracts.

5.1 Two Asset Results

In Chapter 2 an implicit, finite-difference based method was developed to compute the solution of the PIDE which arises from a two-asset option valuation problem under jump diffusion. An approach was presented for the localization of the infinite-domain, initial-value problem to a finite-domain, initial- and boundary-value problem. The convergence of this localization was proved using a Green's function approach. Localizing the integral term of the PIDE generates an error, which is reduced using an approximation in an outer, buffer region. This approximation is, in the final numerical method, inexpensive and simple to compute.

A naïve implicit computation of the discrete integral term in the PIDE would involve the solution of a dense linear system. However, the use of a fixed point iteration reduces this to the problem of carrying out a dense matrix-vector multiply. The integration has the form of a discrete correlation and can be carried out by using two FFT's. The method is straightforward to implement, easily extended to American options through use of a penalty method and jump diffusion can be added to an existing two asset Brownian motion pricing model at the expense of a few FFTs per time step.

When the diffusion coefficients are constant, standard schemes for monotone spatial discretization apply (such as grid rotation) which have the result that the discrete system of equations are monotone for fully implicit time stepping. This property can be used to guarantee convergence of the fixed point iteration, and is a requirement for convergence to the viscosity solution of the whole discrete, linear problem. In the case where the discrete equations are non-monotone, a von Neumann analysis indicates that the fixed point iteration is still globally convergent.

The analysis indicates that, for typical market parameters, the fixed point iteration will reduce the initial residual by seven orders of magnitude in 3 to 5 iterations for European options. Numerical experiments on a synthetic, but typical set of parameters confirm this. Numerical experiments also show quadratic convergence as the grid and time step are refined when Crank–Nicolson time stepping was used. Quadratic convergence was obtained for the solutions on the rotated grid, however, for a given mesh size these had somewhat higher absolute error than a comparable conventional grid.

5.2 Stochastic Volatility and Monotone Approaches

In Chapter 3 the numerical method was extended to two-factor, stochastic volatility with jumps for option valuation on one asset. In this case, sets of market model parameters were available from actual markets. The approach was robust for all of the market cases tested, including after the minor adaptions required to compute using the three market cases which had extreme parameter values.

The stochastic volatility case has partial differential coefficients which vary along one spatial dimension. The approaches used in Chapter 2 that result in the three methods for generating monotone, discrete diffusion terms were adapted to apply to stochastic volatility. The numerical tests in Chapter 3 quantified the cost of each monotone diffusion approach, each of which requires an adapted finite difference grid, compared to using the a non-monotone discretization on a conventional finite difference grid. The grid rotation approach is, by the measures used, to be preferred.

5.3 Monotonicity and Non-linearity

The monotone approaches guarantee the convergence of the penalty and fixed point iterations. These also guarantee the unconditional stability of the method in the l_∞ norm which is a theoretical precondition for convergence to the viscosity solution of the localized PIDE. This method is therefore useful for valuation problems with non-linearities, such as those formulated as solutions to optimal control problems under HJB PIDE's with jumps. An HJB-type PIDE was studied which arises from cases where the jump parameters of the process are uncertain. In Chapter 4 the fixed point iteration is adapted to form a new fixed point policy iteration by adding a control optimization step. American options can be valued by combining a penalty iteration with this technique.

The fixed point policy iteration is proved to be globally convergent, starting from the theory in [47], under a mild time step restriction. Numerical tests confirm that the approach is convergent, in the discretization parameter which specifies the time step, finite difference and discrete policy grid, by at least the linear rate which would be expected for smooth problems.

From a practical perspective, these results show that even fairly small, quite typical uncertainty in the parameters specifying the jump size density can result in large differences between the best- and worst-case option values. The bid/ask spread in a typical option market is of the magnitude seen between the best- and worst-case in the numerical tests of Chapter 4. In this case, the spread would reflect the buyer and seller both valuing the option under their own worst-case scenario. For the synthetic market used to generate the numerical results, the jump correlation was the parameter which generated the widest gap between best- and worst-case values, followed by jump arrival rate then the jump mean.

5.4 Future Work

The FFT-based approximation to the correlation integral requires the use of either an inefficient, constant-spacing finite difference grid, too highly resolved in areas of low interest, or the interpolation scheme developed here. One avenue for improvement might be through Fast Multipole Methods [12] which include the Fast Gauss Transform [56, 109]. These methods were originally developed to solve problems in multi-body dynamics, which have less structure than a problem on a finite difference grid. Current implementations appear to spend time locating poles around which functions are expanded. This is likely not required in the case of an FD grid and would explain the failure of these approaches to perform well in tests reported in [36]. To be useful in a non-linear problem, the monotonicity of the resulting approximations would need to be established. Similarly, $\mathcal{H}$–matrices [57, 58] hold some promise for being able to compute the correlation integral, but that technique remains unexplored for option valuation under jump diffusion.

The problem of maintaining monotonicity in the finite difference approximation is one that is not fully resolved. The stochastic volatility problem explored in this thesis has a very simple, structured form of variability in the diffusion coefficients. Less well structured cases remain an open problem. It may indeed by the case that non-local, finite difference approximation used in Generalized Finite Differences [14] is well suited to the problems considered in this work. This would be especially so if the technique can be modified to respect option payoff features.

The solution of the non-linear problem addressed in this work, which uses discrete controls, is an improvement on existing methods but is still relatively expensive. By making some assumptions about the structure and smoothness of both the solution and the jump PDF, it may be possible to create a method which exploits more conventional optimization techniques. Optimal control problems in finance are an area of active research [92], where practical improvements in algorithms are of interest to the finance community.

Appendix A

Pricing PIDE

The stochastic differential equations (SDE), from which the PIDE's 2.2.3 and 3.2.2 are derived, are listed in this appendix. Note the assumption that there is a single Poisson process which drives correlated jumps in both factors. This corresponds to a single market shock process which affects either both asset prices or both the asset price and its variance [33].

A.1 Two-Asset Problems

The underlying assets for two-asset options are assumed to follow the risk neutral processes

$$\begin{aligned} dS_1 &= (r - \lambda\kappa_1)S_1 \ dt + \sigma_1 S_1 \ dZ_1 + \left(e^{J_1} - 1\right) S_1 \ dq \ , \\ dS_2 &= (r - \lambda\kappa_2)S_2 \ dt + \sigma_2 S_2 \ dZ_2 + \left(e^{J_2} - 1\right) S_2 \ dq \ , \\ &\quad \text{or, in log-price scaling } \ (x_1, x_2) = (\log(S_1), \log(S_2)) \\ dx_1 &= \left(r - \lambda\kappa_1 - \frac{{\sigma_1}^2}{2}\right) \ dt + \sigma_1 \ dZ_1 + J_1 \ dq \ , \\ dx_2 &= \left(r - \lambda\kappa_2 - \frac{{\sigma_2}^2}{2}\right) \ dt + \sigma_2 \ dZ_2 + J_2 \ dq \ , \end{aligned} \tag{A.1.1}$$

where

$$\begin{aligned} \sigma_1, \sigma_2 &= \text{asset volatilities,} \\ r &= \text{risk free rate,} \\ dZ_1, dZ_2 &= \text{increments of Wiener processes,} \\ dZ_1 \ dZ_2 &= \rho \ dt \ , \\ dq &= \begin{cases} 0 & \text{with probability } 1 - \lambda dt \\ 1 & \text{with probability } \lambda dt, \end{cases} \\ \lambda &= \text{mean arrival rate of Poisson jumps } (J_1, J_2) \ , \\ & \quad (S_1, S_2) \rightarrow (e^{J_1} S_1, e^{J_2} S_2) \ , \text{ or} \\ & \quad (x_1, x_2) \rightarrow (x_1 + J_1, x_2 + J_2) \ , \end{aligned}$$

$$
\begin{aligned}
\kappa_1 &= E[e^{J_1} - 1]; \kappa_2 = E[e^{J_2} - 1] \\
E[f(J_1, J_2)] &= \iint_{-\infty}^{\infty} f(J_1, J_2) g(J_1, J_2) \; dJ_1 \; dJ_2 \ , \\
g(J_1, J_2) &= \text{density function of the jump magnitudes in log-price scaling.}
\end{aligned}
$$

From this SDE, by using Ito's formula for finite activity jump processes, one can easily derive the pricing PIDE 2.2.3 by taking expectations under the risk neutral process ([33], [54] §1.4.5).

A.2 Stochastic Volatility Problems

For stochastic volatility with jumps problems, the underlying asset is assumed to follow the risk neutral process

$$
\begin{aligned}
dx &= (r - \frac{v}{2} - \lambda \kappa_x) \; dt + \sqrt{v} \; dZ_x + J_x \; dq \ , \\
dv &= \kappa_v \left(\theta_v - v\right) \; dt + \sigma_v \; dZ_v + J_v \; dq \ ,
\end{aligned}
\tag{A.2.1}
$$

where

$$
\begin{aligned}
\sqrt{v}, \sigma_v &= \text{asset volatility and volatility of volatility,} \\
r &= \text{risk free rate,} \\
dZ_x, dZ_v &= \text{increments of Wiener processes,} \\
dZ_x \; dZ_v &= \rho_v \; dt \ , \\
dq &= \begin{cases} 0 & \text{with probability } 1 - \lambda dt \\ 1 & \text{with probability } \lambda dt, \end{cases} \\
\lambda &= \text{mean arrival rate of Poisson jumps } (J_x, J_v \geq 0) \ , \\
(x, v) &\rightarrow (x + J_x, v + J_v) \ , \\
\kappa_x &= E[e^{J_x} - 1] \\
E[f(J_x, J_v)] &= \int_{-\infty}^{\infty} \int_0^{\infty} f(J_x, J_v) g(J_x, J_v) \; dJ_x \; dJ_v \ , \text{ and} \\
g(J_x, J_v) &= \text{density function of the jump magnitudes.}
\end{aligned}
$$

From this SDE, by using Ito's formula for finite activity jump processes, one can easily derive the pricing PIDE 3.2.2 by taking expectations under the risk neutral process ([33], [54] §1.4.5).

Appendix B

Probability Distribution Functions for Jump Magnitudes

The two jump distributions used for the two-asset numerical examples of Chapter 2 are described in this appendix.

B.1 Bi-Variate Normal in Log-Price

The bi-variate Normal distribution is a straightforward and well defined extension of the univariate case. The probability density function for this distribution is, with parameters for the mean $\breve{\mu}_1$, $\breve{\mu}_2$, standard deviations $\breve{\sigma}_1$, $\breve{\sigma}_2$ and correlation $\breve{\rho}$:

$$g_n(x_1, x_2; \breve{\mu}_1, \breve{\mu}_2, \breve{\sigma}_1, \breve{\sigma}_2, \breve{\rho}) = \frac{1}{2\pi\breve{\sigma}_1\breve{\sigma}_2\sqrt{1-\breve{\rho}^2}} \exp\left\{-\frac{z}{2(1-\breve{\rho}^2)}\right\} \tag{B.1.1}$$

with

$$z = \left(\frac{x_1-\breve{\mu}_1}{\breve{\sigma}_1}\right)^2 - \frac{2\breve{\rho}\,(x_1-\breve{\mu}_1)\,(x_2-\breve{\mu}_2)}{\breve{\sigma}_1\breve{\sigma}_2} + \left(\frac{x_2-\breve{\mu}_2}{\breve{\sigma}_2}\right)^2 .$$

B.2 Marshall-Olkin Bi-Variate Exponential

The two-asset extension of the one-dimensional, double-sided, exponential distribution of Kou [72] is not as straightforward as with the Normal case. Numerous bi-variate exponential distributions have been studied (see [71], Chapter 47). Of these, the Marshall-Olkin bi-variate exponential distribution (MOBED) [80] retains the lack-of-memory property of the one-dimensional distribution. The PDF of this distribution is discontinuous

$$g_x(x_1, x_2; \check{\eta}_1, \check{\eta}_2, \check{\eta}_{12}) = \exp\{-\check{\eta}_1 x_1 - \check{\eta}_2 x_2 - \check{\eta}_{12} \max(x_1, x_2)\} \times \begin{cases} \check{\eta}_1(\check{\eta}_2 + \check{\eta}_{12}) & 0 < x_1 < x_2 \\ \check{\eta}_{12} & 0 < x_1 = x_2 \\ \check{\eta}_2(\check{\eta}_1 + \check{\eta}_{12}) & 0 < x_2 < x_1 \end{cases} \tag{B.2.1}$$

where $1/\check{\eta}_1$ and $1/\check{\eta}_2$ are the mean jumps in direction of x_1 and x_2 by themselves and $1/\check{\eta}_{12}$ is the mean jump of both assets together. Although the PDF is discontinuous and the line $x_1 = x_2$ has a two-dimensional Lebesgue measure of zero, there is a positive probability associated with this line. For the computations in Section 2.4.3.1 the integrated PDF must be expressed in terms of the cumulative distribution function given in [80] §3.1.

To define the PDF in the entire real plane it is written in four quadrants and, following the approach of [72], a probability is assigned to each quadrant. Take $\check{p}_1, \check{p}_2 \in [0, 1]$ as the probability of a positive jump in x_1 and x_2 respectively with $\check{q}_1 = (1 - \check{p}_1)$ and $\check{q}_2 = (1 - \check{p}_2)$ the probability of a negative jump. The PDF is then

$$\begin{aligned} g_m(x_1, x_2; &\check{\eta}_{p,1}, \check{\eta}_{q,1}, \check{\eta}_{p,2}, \check{\eta}_{q,2}, \\ &\check{\eta}_{pp}, \check{\eta}_{qp}, \check{\eta}_{pq}, \check{\eta}_{qq}, \check{p}_1, \check{p}_2) = \quad \check{p}_1\check{p}_2 \cdot g_x(+x_1, +x_2; \check{\eta}_{p,1}, \check{\eta}_{p,2}, \check{\eta}_{pp}) \cdot 1_{x_1, x_2 \geq 0} \\ &+\check{q}_1\check{p}_2 \cdot g_x(-x_1, +x_2; \check{\eta}_{q,1}, \check{\eta}_{p,2}, \check{\eta}_{qp}) \cdot 1_{x_1 < 0, x_2 \geq 0} \\ &+\check{p}_1\check{q}_2 \cdot g_x(+x_1, -x_2; \check{\eta}_{p,1}, \check{\eta}_{q,2}, \check{\eta}_{pq}) \cdot 1_{x_1 \geq 0, x_2 < 0} \\ &+\check{q}_1\check{q}_2 \cdot g_x(-x_1, -x_2; \check{\eta}_{q,1}, \check{\eta}_{q,2}, \check{\eta}_{qq}) \cdot 1_{x_1, x_2 < 0} \end{aligned} \tag{B.2.2}$$

where the $\check{\eta}$ parameters are defined for each of the four quadrants. To avoid further complication, this does not encode the case where the peak of the distribution is offset by $(\check{\mu}_1, \check{\mu}_2)$. Note that to satisfy Assumptions 2.3.1 and 2.3.2 the positive jumps must have a mean $1/\check{\eta} < 1$, or $\check{\eta} > 1$, as in [72].

Appendix C

Localization Error Proofs

Section 2.3.2 left the proofs of Theorems 2.3.7 and 2.3.8 to the following two sections. This appendix employs proofs using Green's functions, which require two spatial coordinate pairs and two time parameters. For brevity, the spatial parameters and related terms are written

$$\begin{aligned} x &= (x_1, x_2) \\ x' &= (x_1', x_2') \\ dx' &= dx_1'\, dx_2' \\ J &= (J_1, J_2) \\ dJ &= dJ_1\, dJ_2 \end{aligned}$$

with the usual element-wise vector sum.

C.1 Proof of Theorem 2.3.7 in Log-Price Scaling: Cutoff Error Due to Localization is Bounded

Theorem 2.3.7 (Restated). *Let U be the solution to PIDE 2.2.3. Let V be the solution an initial-value PIDE*

$$\begin{aligned} \frac{\partial V}{\partial \tau} &= \mathcal{L}V + \lambda_C \mathcal{H}_D V \ , \\ V(x, 0) &= \mathcal{I}(x) \ , \quad x \in \Omega_\infty \end{aligned} \tag{2.3.8}$$

where, for this equation

$$\lambda_C = \begin{cases} \lambda & \textit{for } \ x \in \Omega_C \\ 0 & \textit{for } \ x \in \Omega_\infty \setminus \Omega_C \ \ \textit{(outside of } \Omega_C\textit{)} \end{cases}$$

and $\mathcal{H}_D V$ is defined in Equation 2.3.3. This is similar to localized PIDE 2.3.2, but embedded in Ω_∞ and without a boundary condition.

*Define the **cutoff error** $E_c = U - V$. The value of $E_c(x,\tau)$ for a fixed $x \in \Omega_\infty$ due to the approximation of λ by λ_C and $\mathcal{H}$ by $\mathcal{H}_D$ obeys*

$$\lim_{\Omega_C,\Omega_D \to \Omega_\infty} |E_c(x,\tau)| = 0 \ . \tag{2.3.9}$$

Proof. The objective is to prove bounds on the effect of the approximations to the infinite operators of Equation 2.2.3 by the localized operators in Equation 2.3.2. These approximations are

$$\begin{aligned} \lambda &\simeq \lambda_C \quad \text{and} \\ \mathcal{H} &\simeq \mathcal{H}_D \ . \end{aligned} \tag{C.1.1}$$

The difference between $\mathcal{H}_D$ in Equation 2.3.3 and $\mathcal{H}$ given in Equation 2.2.5 is in the range of the correlation integral, which is performed over Ω_D instead of Ω_∞ respectively. In the following the problem is treated as embedded in the infinite domain Ω_∞ and hence the solution is defined outside of Ω_D.

Define the cutoff error $E = U - V$. Manipulating Equations 2.2.3 and 2.3.8 supplies

$$\begin{aligned} \frac{\partial E}{\partial \tau} &= \mathcal{L}E + \lambda_C \mathcal{H}_D E + (\lambda - \lambda_C)\mathcal{H}U + \lambda(\mathcal{H} - \mathcal{H}_D)U \\ E(x,0) &= 0 \ . \end{aligned} \tag{C.1.2}$$

PIDE C.1.2 satisfies the conditions of Assumption 2.3.4 so that a classical solution can be expressed as a convolution of a Green's function and a source function. With the conditions thus satisfied, the solution to $E(x,\tau)$ can be written as [54]

$$\begin{aligned} E(x,\tau) &= \int_0^\tau \int_{\Omega_\infty} G(x,\tau,x',\tau') \left[(\lambda - \lambda_C)\mathcal{H}U(x',\tau') + \lambda(\mathcal{H} - \mathcal{H}_D)U(x',\tau') \right] \, dx' \, d\tau' \\ E(x,0) &= 0 \end{aligned} \tag{C.1.3}$$

where $G(x,\tau,x',\tau') \geq 0$ is the Green's function of Equation 2.3.8, which is the formal solution to

$$\frac{\partial G}{\partial \tau} = \mathcal{L}G + \lambda_C \mathcal{H}_D G + \delta(x - x', \tau - \tau') \ . \tag{C.1.4}$$

Equation C.1.3 can be rewritten in two components: the first of which expresses error due to a finite $\Omega_C \subset \Omega_\infty$ and the second expresses error due to a finite $\Omega_D \subset \Omega_\infty$. This provides a sum of two errors, each of which is to be bounded separately

$$\begin{aligned} E(x,\tau) &= E_1(x,\tau) + E_2(x,\tau) \\ E_1 &= \int_0^\tau \int_{\Omega_\infty} G(x,\tau,x',\tau') \left[(\lambda - \lambda_C)\mathcal{H}U(x',\tau') \right] \, dx' \, d\tau' \\ E_2 &= \int_0^\tau \int_{\Omega_\infty} G(x,\tau,x',\tau') \left[\lambda(\mathcal{H} - \mathcal{H}_D)U(x',\tau') \right] \, dx' \, d\tau' \ . \end{aligned} \tag{C.1.5}$$

First, estimate E_1 as $\Omega_C \to \Omega_\infty$. This leads to a bound for the integral of the Green's function by noting that the solution to

$$\begin{aligned} \frac{\partial W}{\partial \tau} &= \mathcal{L}W + \lambda_C \mathcal{H}_D W \\ W(x,0) &= \mathcal{I}(x) \end{aligned} \tag{C.1.6}$$

on Ω_∞ can be written as

$$W(x,\tau) = \int_{\Omega_\infty} G(x,\tau,x',0)\,\mathcal{I}(x')\;dx' \;. \tag{C.1.7}$$

Assuming that a bounded solution (for fixed x) exists, then

$$\lim_{\Omega_C \to \Omega_\infty} \int_{\Omega_\infty \setminus \Omega_C} G(x,\tau,x',0)\,\mathcal{I}(x')\;dx' = 0 \tag{C.1.8}$$

must hold and, hence,

$$\lim_{\Omega_C \to \Omega_\infty} \int_0^\tau \int_{\Omega_\infty \setminus \Omega_C} G(x,\tau,x',\tau')\,\mathcal{I}(x')\;dx'\;d\tau' = 0 \;. \tag{C.1.9}$$

In particular, Assumption 2.3.1 is taken to hold so that a solution to Equation C.1.6 exists for any

$$\mathcal{I}(x) \le c_1 + c_2\left(e^{x_1} + e^{x_2}\right) \tag{C.1.10}$$

so that using Equations C.1.9 and C.1.10 gives

$$\lim_{\Omega_C \to \Omega_\infty} \int_0^\tau \int_{\Omega_\infty \setminus \Omega_C} G(x,\tau,x',\tau')\left[c_1 + c_2\left(e^{x'_1} + e^{x'_2}\right)\right]\;dx'\;d\tau' = 0 \;. \tag{C.1.11}$$

If Assumption 2.3.2 holds for the solution U to Equation 2.2.3 over $\Omega_\infty \setminus \Omega_C$

$$|\mathcal{H}U(x,\tau)| \le c_3 + c_4\left(e^{x_1} + e^{x_2}\right) \tag{C.1.12}$$

then, because $\lambda - \lambda_C = 0$ on Ω_C, it follows that

$$\begin{aligned} \lim_{\Omega_C \to \Omega_\infty} |E_1| &\le \lim_{\Omega_C \to \Omega_\infty} \int_0^\tau \int_{\Omega_\infty} G(x,\tau,x',\tau')\left|(\lambda - \lambda_C)\mathcal{H}U(x',\tau')\right|\;dx'\;d\tau' \\ &= \lim_{\Omega_C \to \Omega_\infty} \int_0^\tau \int_{\Omega_\infty \setminus \Omega_C} G(x,\tau,x',\tau')\left|\lambda \mathcal{H}U(x',\tau')\right|\;dx'\;d\tau' \\ &\le \lim_{\Omega_C \to \Omega_\infty} \int_0^\tau \int_{\Omega_\infty \setminus \Omega_C} G(x,\tau,x',\tau')\left[c_5 + c_6\left(e^{x'_1} + e^{x'_2}\right)\right]\;dx'\;d\tau' = 0 \end{aligned} \tag{C.1.13}$$

which, in turn, follows from Equation C.1.11.

Next, bound E_2 as $\Omega_D \to \Omega_\infty$. Note that from Equations 2.2.5 and 2.3.3

$$(\mathcal{H} - \mathcal{H}_D)U(x,\tau) = \int_{(x+J)\in\Omega_\infty \setminus \Omega_D} g(J)\,U(x+J,\tau)\;dJ \;.$$

The preconditions for the existence of

$$\int_{-\infty}^{\infty} g(J)\, U(x+J,\tau)\; dJ$$

imply that

$$\lim_{\Omega_D \to \Omega_\infty} \int_{(x+J)\in\Omega_\infty \setminus \Omega_D} g(J)\, U(x+J,\tau)\; dJ = 0$$

hence

$$\begin{aligned}\lim_{\Omega_D \to \Omega_\infty} E_2 &= \lim_{\Omega_D \to \Omega_\infty} \lambda \int_0^{\tau} \int_{\Omega_\infty} G(x,\tau,x',\tau')(\mathcal{H}-\mathcal{H}_D)\, U(x',\tau')\; dx'\; d\tau' \\ &= \lim_{\Omega_D \to \Omega_\infty} \lambda \int_0^{\tau} \int_{\Omega_\infty} G(x,\tau,x',\tau') \int_{(x'+J)\in\Omega_\infty \setminus \Omega_D} g(J)\, U(x'+J,\tau')\; dJ\; dx'\; d\tau' = 0\end{aligned} \tag{C.1.14}$$

and thus

$$\lim_{\Omega_C,\Omega_D \to \Omega_\infty} |E(x,\tau)| \le |E_1(x,\tau)| + |E_2(x,\tau)| = 0\ .$$

□

C.2 Proof of Theorem 2.3.8: Error Due to Artificial Dirichlet Condition is Bounded

Theorem 2.3.8 (Restated). *Let Y be the solution to Equation 2.3.2 with the approximate boundary condition $Y(x,\tau) = B(x,\tau), x \in \partial\Omega_D$. Let W be the solution to Equation 2.3.2 with the boundary is set to $W(x,\tau) = V(x,\tau), x \in \partial\Omega_D$, where V is the exact value from the solution of Equation 2.3.8. Define the error due to approximating the exact boundary condition $V(x,\tau)$ with the approximate boundary condition $B(x,\tau)$ on $\partial\Omega_D$ as $E_b = W - Y$.*
The error $E_b(x,\tau)$ is bounded as

$$\lim_{\Omega_D \to \Omega_\infty} |E_b(x,\tau)| = 0\ . \tag{2.3.10}$$

Proof: In this section, the error due to approximating the exact boundary condition for Equation 2.3.2 is bounded. Note that the previous section showed that Equation 2.3.8 converges, in theory, to the solution of Equation 2.2.3 as $\Omega_C, \Omega_D \to \Omega_\infty$.

Now, solve for W on the finite domain Ω_D

$$\begin{aligned}\frac{\partial W}{\partial \tau} &= \mathcal{L}W + \lambda_C \mathcal{H}_D W \\ W(x,0) &= \mathcal{I}(x) \\ W(x,\tau) &= V(x,\tau) \quad ; \quad x \in \partial\Omega_D\end{aligned} \tag{C.2.1}$$

where $V(x, \tau)$ is the exact Dirichlet boundary condition, given from the solution to Equation 2.3.8 embedded in Ω_∞. Therefore, noting that the correlation integral of $(x + J) \in \mathcal{H}_D$ of Equation C.2.1 is truncated to operate on Ω_D only, $W = V$ on Ω_D. The solution to Equation C.2.1 is [54] §IV

$$W(x, \tau) = \int_{\Omega_D} G^W(x, \tau, x', 0)\mathcal{I}(x')\, dx' + \int_0^\tau \int_{\partial\Omega_D} P(x, \tau, x', \tau')V(x', \tau')\, dx'\, d\tau' \qquad \text{(C.2.2)}$$

where the Green's function $G^W(x, \tau, x', \tau') \geq 0$ is the formal solution to

$$\begin{aligned} \frac{\partial G^W}{\partial \tau} &= \mathcal{L}G^W + \lambda_C \mathcal{H}_D G^W + \delta(x - x', \tau - \tau') \\ G^W(x, \tau, x', \tau') &= 0\ ; \quad x \in \partial\Omega_D \end{aligned}$$

and $P(x, \tau, x', \tau') \geq 0$ is the Poisson[1] function ([54] § IV) of Equation C.2.1. The solution for V depends only on initial conditions hence can be written

$$V(x, \tau) = \int_{\Omega_\infty} G(x, \tau, x', 0)\mathcal{I}(x')\, dx' \qquad \text{(C.2.3)}$$

where the Green's function G is the solution to Equation C.1.4. Since $V = W$ on Ω_D and, as $\Omega_D \to \Omega_\infty$, $G^W \to G$, then, for any fixed point (x, τ),

$$\lim_{\Omega_D \to \Omega_\infty} \int_0^\tau \int_{\partial\Omega_D} P(x, \tau, x', \tau')V(x', \tau')\, dx'\, d\tau' = 0\ . \qquad \text{(C.2.4)}$$

In particular, this holds for a value V constant in x. If $\mathcal{I}(x) = c_0 > 0$, then $V = c_0 e^{-r\tau}$ along with $c_1 = c_0 e^{-rT} \leq c_0 e^{-r\tau}$ (for $\tau \leq T$) and

$$\begin{aligned} &\lim_{\Omega_D \to \Omega_\infty} \int_0^\tau \int_{\partial\Omega_D} P(x, \tau, x', \tau')\ c_1\, dx'\, d\tau' \\ &\qquad \leq \lim_{\Omega_D \to \Omega_\infty} \int_0^\tau \int_{\partial\Omega_D} P(x, \tau, x', \tau') c_0 e^{-r\tau}\, dx'\, d\tau' = 0\ . \end{aligned} \qquad \text{(C.2.5)}$$

Now, suppose W on Ω_D is approximated by Y where

$$\begin{aligned} \frac{\partial Y}{\partial \tau} &= \mathcal{L}Y + \lambda_C \mathcal{H}_D Y \\ Y(x, 0) &= \mathcal{I}(x) \\ Y(x, \tau) &= B(x, \tau) \quad ; \quad x \in \partial\Omega_D \end{aligned}$$

so that the error $E = W - Y$ satisfies

$$\begin{aligned} \frac{\partial E}{\partial \tau} &= \mathcal{L}E + \lambda_C \mathcal{H}_D E \\ E(x, 0) &= 0 \\ E(x, \tau) &= V(x, \tau) - B(x, \tau) \quad ; \quad x \in \partial\Omega_D \end{aligned} \qquad \text{(C.2.6)}$$

[1] Intuitively, the Poisson function serves to encode the boundary conditions of the problem and Green's functions solve for the interior when the boundary condition is zero.

with solution [54]

$$E(x,\tau) = \int_0^\tau \int_{\partial\Omega_D} P(x,\tau,x',\tau') \left[V(x',\tau') - B(x',\tau')\right] \, dx' \, d\tau' \; . \tag{C.2.7}$$

Without loss of generality, assume that $\mathcal{I}(x) \geq 0$, so that $V(x,\tau) \geq 0$. Suppose that, by Assumption 2.3.3

$$|V(x,\tau) - B(x,\tau)| \leq c_1 + c_2 V(x,\tau)$$

(which is trivially satisfied if $B(x,\tau) = 0$), then Equation C.2.7 becomes

$$|E(x,\tau)| \leq \int_0^\tau \int_{\partial\Omega_D} P(x,\tau,x',\tau') \left[c_1 + c_2 V(x',\tau')\right] \, dx' \, d\tau' \; . \tag{C.2.8}$$

From Equations C.2.4 and C.2.5 it follows that

$$\lim_{\Omega_D \to \Omega_\infty} |E(x,\tau)| = 0 \; .$$

□

Appendix D

Convergence of the Fixed Point Iteration

In this appendix the convergence rate for the fixed point iteration of Section 2.6.1 is demonstrated both as a functional iteration using the continuous operators and as an iteration using the general discrete linear system operators as outlined in Section 2.4.1. The compact notation of Appendix C is also used here.

D.1 Proof of Theorem 2.6.1: Convergence of the Functional Fixed Point Iteration

Theorem 2.6.1 (Restated): *Let $E^k = W^{n+1} - Z^k$ be the error in the solution to the semi-discretized Equation 2.4.2 at iteration k of the functional iteration given in Equation 2.6.1. The iteration is convergent to zero as*

$$\|E^{k+1}\|_\infty \leq \frac{(1-\theta)\Delta\tau\,\lambda}{1+(1-\theta)\Delta\tau\,(r+\lambda)}\|E^k\|_\infty \ .$$

Proof: Note that operator $\mathcal{G}$ contains only differential terms so if c is a constant then $\mathcal{G}c = 0$. Impose a Dirichlet boundary condition over all of $\partial\Omega_D$.

Let $E^k = W^{n+1} - Z^k$ to obtain the error propagation equation

$$\begin{aligned}[1+(1-\theta)\Delta\tau\,(r+\lambda_C-\mathcal{G})]\,E^{k+1} &= (1-\theta)\Delta\tau\,\lambda_C\mathcal{J}_D E^k\\ E^{k+1}(x,\tau) &= 0\ ; \quad x\in\partial\Omega_D\end{aligned}$$

with solution [54] §IV

$$E^{k+1}(x) = \int_{\Omega_D} G(x,x')\,(1-\theta)\Delta\tau\,\lambda_C\mathcal{J}_D E^k(x')\,dx' \qquad \text{(D.1.1)}$$

where $G(x, x') > 0$ is the formal solution of

$$\begin{aligned} \left[1 + (1-\theta)\Delta\tau\,(r + \lambda_C - \mathcal{G})\right] G &= \delta(x - x') \\ G(x, x') &= 0\ ; \quad x \in \Omega_D\ . \end{aligned}$$

Note that $E^0 = W^{n+1} - W^n$ is the difference between two smooth functions, hence is also smooth and that $\mathcal{G}$ is a uniformly elliptic operator with bounded coefficients on Ω_D (see [54] §IV.2).

Now, consider the equation

$$\begin{aligned} \left[1 + (1-\theta)\Delta\tau\,(r + \lambda_C - \mathcal{G})\right] A &= f(x)\ ; \quad x \in \Omega_D \\ A(x) &= B(x)\ ; \quad x \in \partial\Omega_D \end{aligned} \tag{D.1.2}$$

which has the formal solution

$$A(x) = \int_{\Omega_D} G(x, x') f(x')\, dx' + \int_{\partial\Omega_D} P(x, x') B(x')\, dx' \tag{D.1.3}$$

where $P(x, x') \geq 0$ is a Poisson function induced by the non-zero boundary conditions ([54] §IV.3). Let $A(x) = 1$. We can then compute the RHS of Equation D.1.2 directly

$$\left[1 + (1-\theta)\Delta\tau\,(r + \lambda_C - \mathcal{G})\right] A = \left[1 + (1-\theta)\Delta\tau(r + \lambda_C)\right]$$

so that, if

$$\begin{aligned} f(x) &= 1 + (1-\theta)\Delta\tau\,(r + \lambda_C) \\ B(x) &= 1\ ; \quad A(x) = 1 \end{aligned} \tag{D.1.4}$$

then combining equations (D.1.3) and (D.1.4) the integral is split into two components based on the spatial value of λ_C. Note that $G(x, x') \geq 0$ and $P(x, x') \geq 0$ to obtain

$$\begin{aligned} \int_{\Omega_C} G(x, x')\left[1 + (1-\theta)\Delta\tau\,(r + \lambda_C)\right]\, dx' + \int_{\partial\Omega_D} P(x, x')\, dx' &= \\ \int_{\Omega_D \backslash \Omega_C} G(x, x')\left[1 + (1-\theta)\Delta\tau\, r\right]\, dx' + \int_{\Omega_C} G(x, x')\left[1 + (1-\theta)\Delta\tau\,(r + \lambda)\right]\, dx' & \\ + \int_{\partial\Omega_D} P(x, x')\, dx' &= 1 \\ \left[1 + (1-\theta)\Delta\tau\,(r + \lambda)\right] \int_{\Omega_C} G(x, x')\, dx' &\leq 1 \end{aligned}$$

or

$$\int_{\Omega_C} G(x, x')\, dx' \leq \frac{1}{1 + (1-\theta)\Delta\tau\,(r + \lambda)}\ . \tag{D.1.5}$$

Noting that

$$\mathcal{J}_D \|E^k\|_\infty \leq \|E^k\|_\infty$$

and $G(x,x') \geq 0$ then Equation D.1.1 gives

$$E^{k+1} \leq (1-\theta)\Delta\tau\,\lambda \|E^k\|_\infty \int_{\Omega_C} G(x,x')\,dx' \tag{D.1.6}$$

which becomes, by Equation D.1.5,

$$\|E^{k+1}\|_\infty \leq \frac{(1-\theta)\Delta\tau\,\lambda}{1+(1-\theta)\Delta\tau\,(r+\lambda)} \|E^k\|_\infty \ .$$

Hence the functional iteration (2.6.1) is unconditionally convergent, with rapid convergence in the usual case where $\lambda\Delta\tau \ll 1$.

D.2 Proof of Theorem 2.6.2: Convergence of the Discrete Fixed Point Iteration

Theorem 2.6.2 (Restated): *Let* $\mathsf{e}^k = \mathsf{w}^{n+1} - \mathsf{z}^k$ *be the error in the solution to Equation 2.4.9 at iteration* k *of the fixed-point iteration given in Equation 2.6.2. If* $\mathsf{J} \geq 0$ *has maximum row sum* $(\max_\mathsf{i} \sum_\mathsf{j} \mathsf{J}_\mathsf{ij}) \leq 1$ *and* $(-\mathsf{G})$ *is M-compatible with* $\|\mathsf{G}\cdot\underline{1}\|_\infty = 0$*, then the error in the iterative solution* z^{k+1} *in Equation 2.6.2 is convergent to zero as*

$$\|\mathsf{e}^{k+1}\|_\infty \leq \frac{(1-\theta)\Delta\tau\,\lambda}{1+(1-\theta)\Delta\tau\,(r+\lambda)} \|\mathsf{e}^k\|_\infty \ .$$

Proof: For the following, denote by $\underline{1}$ the vector $\underline{1}_\mathsf{i} = 1 \ \forall \ \mathsf{i}$. Three conditions must hold:

1. $\|\mathsf{G}\cdot\underline{1}\|_\infty = 0$, i.e. the differential approximation is exact for a constant vector,
2. $\mathsf{J} \geq 0$ and $\max_\mathsf{i} \sum_\mathsf{j} \mathsf{J}_\mathsf{ij} \leq 1$ thus $\|\mathsf{J}\cdot\underline{1}\|_\infty \leq 1$ and
3. $(-\mathsf{G})$ is M-compatible so that $\mathsf{N}^{-1} \geq 0$ exists ([93] Theorem F15) where

   $$\mathsf{N} = \{\mathsf{I} - (1-\theta)\Delta\tau\,[\mathsf{G} - r\,\mathsf{I} - \lambda\mathsf{I}_\mathsf{c}]\}$$

   is the matrix of the LHS of the iteration.

Let $\mathsf{e}^k = \mathsf{w}^{n+1} - \mathsf{z}^k$. Thus the error in the solution iteration Equation 2.6.2 propagates as

$$\mathsf{e}^{k+1} = \left[(1-\theta)\Delta\tau\,\lambda\right]\,\left[\mathsf{N}^{-1}\mathsf{I}_\mathsf{c}\mathsf{J}\right]\,\mathsf{e}^k \ . \tag{D.2.1}$$

Taking the norm of Equation D.2.1

$$\left\|\mathsf{e}^{k+1}\right\|_\infty = \left\|\left[(1-\theta)\Delta\tau\,\lambda\right]\,\left[\mathsf{N}^{-1}\mathsf{I}_\mathsf{c}\mathsf{J}\right]\,\mathsf{e}^k\right\|_\infty \leq \left[(1-\theta)\Delta\tau\,\lambda\right]\left\|\mathsf{N}^{-1}\mathsf{I}_\mathsf{c}\right\|_\infty \|\mathsf{J}\|_\infty \|\mathsf{e}^k\|_\infty \tag{D.2.2}$$

by the compatibility of the $\|\cdot\|_\infty$ norm and triangle inequality. Condition 1 holds, therefore

$$\mathsf{N}\cdot\underline{1} \geq [1+(1-\theta)\Delta\tau\,(r+\lambda)]\,\mathsf{I}_\mathsf{c}\cdot\underline{1} \quad \text{and} \quad \mathsf{N}^{-1}\mathsf{I}_\mathsf{c}\cdot\underline{1} \leq \frac{1}{1+(1-\theta)\Delta\tau\,(r+\lambda)}\underline{1}$$

and because Condition 3 holds $\mathsf{N}^{-1} \geq 0$ exists and it follows that

$$\left\|\mathsf{N}^{-1}\mathsf{I_c} \cdot \underline{1}\right\|_\infty = \left\|\mathsf{N}^{-1}\mathsf{I_c}\right\|_\infty$$

by the definition of the row maximum norm of a matrix. This leads in turn to $\left\|\mathsf{J}\right\|_\infty \leq 1$ by Condition 2, and Equation D.2.2 may be written as

$$\|\mathsf{e}^{k+1}\|_\infty \leq \frac{(1-\theta)\Delta\tau\,\lambda}{1+(1-\theta)\Delta\tau\,(r+\lambda)}\|\mathsf{e}^k\|_\infty \ . \tag{D.2.3}$$

Thus the fixed point iteration Equation 2.6.2 is convergent with the rate stated in Theorem 2.6.2.

Appendix E

Discretization Notes

This appendix contains details concerning the discretization of the partial differential terms and about convergence estimates.

E.1 Finite Difference Stencils

For reference, particularly in Appendix F, the following finite difference stencils are supplied over a function $U(x_1, x_2)$. For further information the reader may wish to consult [1] §25.3,[89] §9.4, or [40] §8. The extension of these operators to a non-constant grid spacing is omitted for brevity, but is straightforward.

The grid of points $x = (x_1, x_2)$, on which this set of finite difference stencils is defined, has constant spacing h_1 and h_2 in x_1 and x_2 with $h_1 \neq h_2$ in general. Each finite difference is written in a compact linear operator form, for example $\partial_1^{h_1}$.

When Dirichlet boundary conditions are applied, the boundary nodes are not included in the solution vector. Where entries in a finite difference stencil centered on an interior node refer to adjacent boundary nodes, they do not appear in the matrix G of the discrete differential operator and hence are, in effect, zero. The off-diagonal entries which cannot be represented in G are multiplied by the known boundary condition values then summed into the boundary enforcement vector b of Equation 2.4.9. Other methods of imposing Dirichlet boundary conditions are equally effective; this approach permits the ready application of the l_∞ norm stability analyses in [73, 77, 102].

First partial derivatives are approximated with a second order central difference

$$\frac{\partial U(x_1, x_2)}{\partial x_1} \simeq \frac{1}{2\,h_1}\left[U(x_1 + h_1, x_2) - U(x_1 - h_1, x_2)\right] = \partial_1^{h_1} U(x_1, x_2)\ . \tag{E.1.1}$$

Where the second order approximation results in negative values for the off-diagonal coefficients of matrix G of Equation 2.4.8, a first order approximation may be used. Either the forward

$$\frac{\partial U(x_1, x_2)}{\partial x_1} \simeq \frac{1}{h_1}\left[U(x_1 + h_1, x_2) - U(x_1, x_2)\right] = \partial_1^{h_1 +} U(x_1, x_2) \tag{E.1.2}$$

or backward

$$\frac{\partial U(x_1,x_2)}{\partial x_1} \simeq \frac{1}{h_1}\left[U(x_1,x_2) - U(x_1-h_1,x_2)\right] = \partial_1^{h_1-} U(x_1,x_2) \tag{E.1.3}$$

difference may be applied, depending on the leading coefficients. The second order partial difference is taken with the second order equation ([1] §25.3.23)

$$\frac{\partial^2 U(x_1,x_2)}{\partial {x_1}^2} \simeq \frac{1}{{h_1}^2}\left[U(x_1+h_1,x_2) + U(x_1-h_1,x_2) - 2U(x_1,x_2)\right] = \partial_{11}^{h_1} U(x_1,x_2)\ . \tag{E.1.4}$$

The above stencils are constructed analogously in the x_2 direction.

A cross-partial derivative on a seven-point stencil is taken by using one of the two, following, complementary choices([1] §25.3.27). The first is selected for problems where $\rho > 0$

$$\begin{aligned}\frac{\partial^2 U(x_1,x_2)}{\partial x_1\,\partial x_2} \simeq \frac{1}{2\,h_1\,h_2}\Big[&U(x_1+h_1,x_2+h_2) + U(x_1-h_1,x_2-h_2) + 2\,U(x_1,x_2)\\ &- U(x_1+h_1,x_2) - U(x_1-h_1,x_2)\\ &- U(x_1,x_2+h_2) - U(x_1,x_2-h_2)\Big] = \partial_{1,2}^{h_1h_2+} U(x_1,x_2)\end{aligned} \tag{E.1.5}$$

and the second is appropriate for $\rho < 0$

$$\begin{aligned}\frac{\partial^2 U(x_1,x_2)}{\partial x_1\,\partial x_2} \simeq \frac{-1}{2\,h_1\,h_2}\Big[&U(x_1-h_1,x_2+h_2) + U(x_1+h_1,x_2-h_2) + 2\,U(x_1,x_2)\\ &- U(x_1+h_1,x_2) - U(x_1-h_1,x_2)\\ &- U(x_1,x_2+h_2) - U(x_1,x_2-h_2)\Big] = \partial_{1,2}^{h_1h_2-} U(x_1,x_2)\ .\end{aligned} \tag{E.1.6}$$

A nine-point stencil may be used to obtain a four point cross-partial difference ([1] §25.3.26)

$$\begin{aligned}\frac{\partial^2 U(x_1,x_2)}{\partial x_1\,\partial x_2} \simeq \frac{1}{2\,h_2}\Bigg[&\frac{U(x_1+h_1,x_2+h_2) - U(x_1-h_1,x_2+h_2)}{2\,h_1}\\ &- \frac{U(x_1+h_1,x_2-h_2) - U(x_1-h_1,x_2-h_2)}{2\,h_1}\Bigg] = \partial_{1,2}^{h_1h_2\circ} U(x_1,x_2)\ .\end{aligned} \tag{E.1.7}$$

E.2 Coefficients for Skew Grids

Under the grid skew of Hull and White, Equation 2.5.2 or 3.4.7 [62, 112] all partial differential terms of the option valuation PIDE are transformed. Thus the first order terms are, in transformed co-ordinates

$$c_1\sigma_2\frac{\partial V}{\partial \psi_1} + c_2\sigma_1\frac{\partial V}{\partial \psi_2}$$

where c_1, c_2 are the coefficients of the original ∂x_1 and ∂x_2 terms respectively. The second order terms in transformed co-ordinates are

$$\left(d_{11}{\sigma_2}^2 + 2d_{12}\sigma_1\sigma_2 + d_{22}{\sigma_1}^2\right)\frac{\partial^2 V}{\partial {\psi_1}^2} + \left(d_{11}{\sigma_2}^2 - 2d_{12}\sigma_1\sigma_2 + d_{22}{\sigma_1}^2\right)\frac{\partial^2 V}{\partial {\psi_2}^2}$$

where d_{11} is the coefficient of the original $\partial {x_1}^2$ term, d_{12} is the coefficient of the original $\partial x_1 x_2$ term, and d_{22} is the coefficient of the original $\partial {x_2}^2$ term. For stochastic volatility problems, set $\sigma_1 = 1.0$ and $\sigma_2 = \sigma_v$ in the above equations.

E.3 Coefficients for Rotated Grids

Under the grid rotation of Equation 2.5.4 or 3.4.9 the first-order partial differential terms are, in the transformed co-ordinate system (r_1, r_2),

$$(c_1 \cos(\theta_r) + c_2 \sin(\theta_r))\frac{\partial V}{\partial r_1} - (-c_1 \sin(\theta_r) + c_2 \cos(\theta_r))\frac{\partial V}{\partial r_2}$$

where c_1, c_2 are the coefficients of the original ∂x_1 and ∂x_2 terms respectively. The second order terms in transformed co-ordinates are

$$\begin{aligned}&\left(d_{11}\cos^2(\theta_r) + 2d_{12}\sin(\theta_r)\cos(\theta_r) + d_{22}\sin^2(\theta_r)\right)\frac{\partial^2 V}{\partial {r_1}^2}\\ + &\left(d_{11}\sin^2(\theta_r) - 2d_{12}\sin(\theta_r)\cos(\theta_r) + d_{22}\cos^2(\theta_r)\right)\frac{\partial^2 V}{\partial {r_2}^2}\end{aligned}$$

where d_{11} is the coefficient of the original $\partial {x_1}^2$ term, d_{12} is the coefficient of the original $\partial x_1 x_2$ term, and d_{22} is the coefficient of the original $\partial {x_2}^2$ term. For stochastic volatility problems, set $\sigma_1 = 1.0$ and $\sigma_2 = \sigma_v$ in the above equations.

E.4 Estimating Convergence

In the numerical demonstrations of Sections 2.7, 3.5 and 4.5, estimates are given of a convergence parameter α as a grid control parameter h is refined. Assuming that $\Delta x_1, \Delta x_2, \Delta\tau = \mathrm{O}(h)$ for two-asset problems, $\Delta x, \Delta v, \Delta\tau = \mathrm{O}(h)$ for stochastic volatility problems or $\Delta x_1, \Delta x_2, \Delta\tau, h_y = \mathrm{O}(h)$ for two-asset problems under uncertain parameters, the error is assumed to be $\epsilon_i = ch_i^\alpha$ for a grid spacing h_i.

If two grids with $h_1 = 2h_2$ produced solutions V_1 and V_2, and where an exact solution V_e was available, then solving for α is straightforward. Where three grids with spacing $h_1 > h_2 > h_3$ were used to solve for values V_1, V_2 and V_3 respectively, but no exact solution was available, then the exponent α was estimated by solving the non-linear equation

$$\frac{V_1 - V_2}{V_2 - V_3} = \frac{c\,h_1^\alpha - c\,h_2^\alpha}{c\,h_2^\alpha - c\,h_3^\alpha} = \frac{h_1^\alpha - h_2^\alpha}{h_2^\alpha - h_3^\alpha}$$

by Newton's method.

Appendix F

Proof of Theorems 2.5.11 and 2.6.3: Stability and Convergence by von Neumann Analysis

Theorem 2.5.11 (Restated): *Consider a periodic formulation of Equation 2.3.2, discretized with a finite difference approximation on a grid with constant spacing. The problem is formed with constant coefficients in the drift and diffusion terms, $-1 \leq \rho \leq 1$ and $\lambda_C = \lambda \geq 0$ constant. Either the cross-partial finite difference of Equations E.1.5 or E.1.6 are used to form the approximation G_s to $\mathcal{G}$ on a seven-point stencil, or Equation E.1.7 is used to form G_n over a nine-point stencil.*

The time step Equation 2.4.9 is unconditionally von Neumann stable in the l_2 norm for $\theta = 0$ and for $\theta = 1/2$.

Theorem 2.6.3 (Restated): *Consider a periodic formulation of Equation 2.3.2, discretized with a finite difference approximation. Let $(-\mathsf{G})$ be formed either by the 7-point or 9-point finite difference stencil on a grid with constant spacing, as in Theorem 2.5.11.*

Then in the sense of von Neumann analysis the iterative solution to Equation 2.4.9 by Equation 2.6.2 is unconditionally convergent in the l_2 norm (i.e. regardless of whether $(-\mathsf{G})$ is M-compatible) at a rate which is rapid if $\lambda\Delta\tau \ll 1$.

The von Neumann analysis in this appendix applies to a periodic, initial value problem with the same, constant coefficients operators as the PIDE 2.2.3. Thus it applies to two-asset problems, rather than stochastic volatility problems. The following sections employ a number of elements in common, hence finish with both the time step and the fixed point iteration analysis only after some preliminary discussions. Sections F.2 and F.3 describe the basic mathematics used for the analysis. Section F.4 defines the problem and error propagation equations to be analyzed, then Section F.5 discusses the approach to the final analysis and the relations that must hold to prove Theorems 2.5.11 and 2.6.3. In Sections F.6 and F.7 those relations are demonstrated.

F.1 Structure of the Domain, Grid and DFT

The periodic, discrete form of the PIDE 2.2.3 localized over a periodic domain Ω_D° is the basis for von Neumann analysis. The problem is phrased as a pure initial value problem with nodes of $\partial\Omega_D$ included in Ω_D°. A regular rectangular grid of $N_F = N_1 \times N_2$ equally spaced nodes is required where, without loss of generality, N_1, N_2 are taken to be powers of 2. The finite difference discretization on this grid is defined in Appendix E above using node spacing (h_1, h_2). As in Section 2.4.3.1 the location of the points on the grid is defined as

$$(x_1, x_2)_i \in \Omega_D^\circ \ , \quad i = 1 \ldots N_F \ . \tag{F.1.1}$$

The DFT operation makes it convenient to introduce the double-subscript notation $U_{kl} = U((x_1, x_2)_i)$ to denote a point on the grid where (k, l) are the grid line coordinates. The grid point $(x_1, x_2)_i$, at position i in the solution vector, is related to grid line coordinates (k, l) by a mapping such as

$$i = k + l\,N_1 + 1 \ , \quad k = 0 \ldots (N_1 - 1) \ , \quad l = 0 \ldots (N_2 - 1) \ .$$

Where a vector u of a value over the grid is required, its components are denoted $\mathsf{u}_\mathsf{i} = U_{kl} = U((x_1, x_2)_i)$ assuming that this ordinate mapping holds (although different letters may be used in the subscripts). The same double-subscript notation is used to locate points on the grid in the Fourier-transformed space.

The DFT $\mathcal{D}(U)$ over periodic domain Ω_D° results in a $N_F = N_1 \times N_2$ grid of coefficients denoted as $\hat{U}_{mn}$ where the following, compact notation, is used

$$\begin{aligned} \hat{U}_{mn} &= \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} U_{ij}\,\xi(-im - jn) = \sum_{ij} U_{ij}\,\xi(-im - jn) \\ U_{ij} &= \frac{1}{N_F} \sum_{m=-N_1/2+1}^{N_1/2} \sum_{n=-N_1/2+1}^{N_2/2} \hat{U}_{mn}\,\xi(im + jn) = \frac{1}{N_F} \sum_{mn} \hat{U}_{mn}\,\xi(im + jn) \end{aligned} \tag{F.1.2}$$

where

$$\xi(k) = \exp\left\{\sqrt{-1}\,\zeta\,k\right\} \ , \text{ and } \ \zeta = \frac{2\pi}{N_F} \ .$$

Note that the correction for the grid node count is done during the inverse transform. Usually the ranges are dropped on the summations as in Equation F.1.2, assuming that the log-price space indices (i, j) and the Fourier-space indices (m, n) refer to their periodic image ("wrap around") under an addition which crosses the grid boundary. Note the following useful identities

$$\begin{aligned} \xi(+m) + \xi(-m) &= 2\cos(\zeta m) \ , \\ \xi(+m) - \xi(-m) &= 2\sqrt{-1}\sin(\zeta m) \ \text{ and} \\ \frac{\xi(m+i)}{\xi(m)} &= \xi(i) \end{aligned} \tag{F.1.3}$$

and note that, for example,

$$\begin{aligned} U_{i+1,j+1} - U_{i-1,j-1} &= \sum_{mn} \Big[\hat{U}_{mn}\, \xi((i+1)m + (j+1)n) - \hat{U}_{mn}\, \xi((i-1)m + (j-1)n) \Big] \\ &= \sum_{mn} \Big[\xi(+m+n) - \xi(-m-n) \Big] \hat{U}_{mn}\, \xi(m+n) \end{aligned} \tag{F.1.4}$$

which will be used to reduce finite difference expressions.

F.2 Discrete Fourier Transform of Finite Difference Stencils

The coefficients of partial differential component of Equation 2.3.2 are constant in the initial value problem. This means the DFT may be applied to each of the finite difference stencils in Appendix E ([17] §13). For clarity, write the value for a single coefficient $\hat{U}_{mn}$ of the Fourier transform $\hat{\partial}$ of difference operator ∂ at a single node U_{ij}, leaving off the leading sums and coefficients. The objective is to write the difference equations in terms of trigonometric functions, thus the identities of Equations F.1.3 through F.1.4 are applied. For brevity, write $\xi_p = \xi(im{+}jn)$ in the final form. Discretizations that require only a change of axes are omitted.

$$\begin{aligned} (h_1)\ \hat{\partial}_1^{h_1} \hat{U}_{mn}\Big|_{ij} &= \big\{ \xi((i+1)m + jn) - \xi((i-1)m + jn) \big\}\, \hat{U}_{mn} \\ &= 2\sqrt{-1}\, \sin(\zeta m)\, \xi_p\, \hat{U}_{mn}\ , \end{aligned} \tag{F.2.1}$$

$$\begin{aligned} \left({h_1}^2\right)\ \hat{\partial}_{1,1}^{h_1} \hat{U}_{mn}\Big|_{ij} &= \big\{ \xi((i+1)m + jn) - 2\xi(im + jn) + \xi((i-1)m + jn) \big\}\, \hat{U}_{mn} \\ &= 2 \big\{ \cos(\zeta m) - 1 \big\}\, \xi_p\, \hat{U}_{mn}\ . \end{aligned} \tag{F.2.2}$$

The following two cross-partial derivatives complete the seven point discretization:

$$\begin{aligned} (2\, h_1\, h_2)\ \hat{\partial}_{1,2}^{h_1,h_2+} \hat{U}_{mn}\Big|_{ij} = \big\{ &\xi((i+1)m + (j+1)n) + \xi((i-1)m + (j-1)n) \\ &+ 2\xi(im + jn) \\ &- \xi((i+1)m + jn) - \xi((i-1)m + jn) \\ &- \xi(im + (j+1)n) - \xi(im + (j-1)n) \big\}\, \hat{U}_{mn} \\ = 2 \big\{ &1 + \cos(\zeta m + \zeta n) - \cos(\zeta m) - \cos(\zeta n) \big\}\, \xi_p\, \hat{U}_{mn}\ , \end{aligned} \tag{F.2.3}$$

$$\begin{aligned} (2\, h_1\, h_2)\ \hat{\partial}_{1,2}^{h_1,h_2-} \hat{U}_{mn}\Big|_{ij} = -\big\{ &\xi((i+1)m + (j-1)n) + \xi((i-1)m + (j+1)n) \\ &+ 2\xi(im + jn) \\ &- \xi((i+1)m + jn) - \xi((i-1)m + jn) \\ &- \xi(im + (j+1)n) - \xi(im + (j-1)n) \big\}\, \hat{U}_{mn} \\ = -2 \big\{ &1 + \cos(\zeta m - \zeta n) - \cos(\zeta m) - \cos(\zeta n) \big\}\, \xi_p\, \hat{U}_{mn}\ . \end{aligned} \tag{F.2.4}$$

The following cross-partial derivative is used in the nine-point discretization:

$$\begin{aligned}(4\,h_1\,h_2)\ \hat{\partial}_{1,2}^{h_1,h_2\circ}\hat{U}_{mn}\Big|_{ij} &= \big\{\xi((i+1)m+(j+1)n) - \xi((i-1)m+(j-1)n) \\ &\qquad - \xi((i-1)m+(j+1)n) + \xi((i+1)m+(j-1)n)\big\}\,\hat{U}_{mn} \\ &= \big\{-4\sin(\zeta m)\sin(\zeta n)\big\}\ \xi_p\,\hat{U}_{mn}\ . \end{aligned} \tag{F.2.5}$$

F.3 Discrete Fourier Transform of the Correlation Term

Recall from Section 2.4.3.3 that the discrete version of the integral correlation term between values U and g may be written as a dense matrix-vector product Ju. In Equation 2.4.19 this product is written as an operation on the DFT of the jump distribution and the option value vector on the periodic grid.

Recall that $g(x_1, x_2)$ is a PDF and that the points on the grid defined by Equation 2.4.17 for the correlation Equation 2.4.19 are defined by integrating $g(x_1, x_2)$ over the DFT cell. Let

$$f_{ij} = f((x_1, x_2)_k) = \int_{-h_1/2}^{+h_1/2}\int_{-h_2/2}^{+h_2/2} g(x_1+z_1, x_2+z_2)\ dz_1\ dz_2$$

and note that

$$f_{ij}\in\mathbb{R}, \quad f_{ij}\geq 0 \quad \text{and} \quad \sum_{ij} f_{ij} = 1\ .$$

Thus, taking the Fourier transform

$$\hat{f}_{mn} = \sum_{ij} f_{ij}\ \xi(-im-jn) \quad \text{implies that} \quad \Big|\hat{f}_{0,0}\Big| \leq 1 \quad \text{and} \quad \Big|\hat{f}_{mn}\Big| \leq 1\ .$$

The magnitude of the DFT of the jump distribution and its real and imaginary components are denoted

$$\overline{(\hat{f}_{mn})} = \hat{f}_{-m,-n} = \left(\hat{f}^{R}_{-m,-n} + \hat{f}^{I}_{-m,-n}\sqrt{-1}\,\right) \tag{F.3.1}$$

in the final proof. Using this notation, note that

$$0 \leq \left(1-\hat{f}^{R}_{mn}\right) \leq 2\ . \tag{F.3.2}$$

F.4 Discrete Option Value and Error Propagation PIDE

In this section it is useful to relate this analysis to the original form of the problem by using a matrix, vector notation similar to that of Section 2.4.1. Write the approximation to PIDE 2.2.3 as a periodic initial value problem on a grid with a discrete solution written in a vector u. It is discretized using a theta method from step t to $t+1$ with time step weight θ as in

Equation 2.4.9 and a finite difference method selected from Appendix E. The time step is taken by solving an equation in the form of Equation 2.4.9

$$[\mathsf{I} + (1-\theta)\Delta\tau\ \mathsf{T}]\,\mathsf{u}^{t+1} = [\mathsf{I} - \theta\Delta\tau\ \mathsf{T}]\,\mathsf{u}^{t} \tag{F.4.1}$$

where

$$\mathsf{T} = -\left[\mathsf{G} + \lambda(\mathsf{J} - \mathsf{I}) - r\,\mathsf{I}\right] = -\mathsf{G} - \lambda\,\mathsf{J} + \lambda\,\mathsf{I} + r\,\mathsf{I}\ . \tag{F.4.2}$$

Matrix G is defined by Equation 2.4.1, where the partial differential terms have constant coefficients and difference Equations E.1.1 and E.1.4 with either Equation E.1.5, E.1.6 or E.1.7 for the cross-partial derivative. J is as defined in Equation 2.4.19. Note that λ_C or $\mathsf{I_c}$ do not appear since, for this analysis, the matrix coefficients must be the same for each point in the system.

Let e^t be an arbitrary perturbation error to the solution u. The error propagates by

$$[\mathsf{I} + (1-\theta)\Delta\tau\ \mathsf{T}]\,\mathsf{e}^{t+1} = [\mathsf{I} - \theta\Delta\tau\ \mathsf{T}]\,\mathsf{e}^{t}\ . \tag{F.4.3}$$

The fixed point iteration method solves Equation F.4.1 with approximate solution z^k at the k-th fixed point iteration solving from time step t to $t+1$

$$\begin{aligned}&\{\mathsf{I} + (1-\theta)\Delta\tau\ [-\mathsf{G} + \lambda\mathsf{I} + r\,\mathsf{I}]\}\,\mathsf{z}^{k+1}\\ &\qquad = (1-\theta)\Delta\tau\lambda\mathsf{J}\mathsf{z}^k + \{\mathsf{I} - \theta\Delta\tau\,[-\mathsf{G} - \lambda\mathsf{J} + \lambda\mathsf{I} + r\,\mathsf{I}]\}\,\mathsf{u}^t\ .\end{aligned} \tag{F.4.4}$$

For the fixed point iteration, write the solution error $\mathsf{e}^k = \mathsf{u}^{t+1} - \mathsf{z}^k$. The error e^k of the intermediate solution vector z^k propagates by

$$\{\mathsf{I} + (1-\theta)\Delta\tau\,[-\mathsf{G} + r\,\mathsf{I} + \lambda\mathsf{I}]\}\,\mathsf{e}^{k+1} = (1-\theta)\Delta\tau\lambda\mathsf{J}\mathsf{e}^k\ . \tag{F.4.5}$$

F.5 General Approach to the Proof

Arrange the Fourier transform of the time step Equation F.4.3 into a complex valued form for a single coefficient $\hat{E}_{mn}$ of the transform. This must not increase during the time step. The ratio of this coefficient between time step t and $t+1$ is given by

$$\frac{\hat{E}^{t+1}_{mn}}{\hat{E}^{t}_{mn}} = \frac{1 - \theta\Delta\tau\left(-a - b\sqrt{-1} + r + \lambda\right) + \theta\Delta\tau\lambda\hat{f}_{-m,-n}}{1 + (1-\theta)\,\Delta\tau\left(-a - b\sqrt{-1} + r + \lambda\right) - (1-\theta)\,\Delta\tau\lambda\hat{f}_{-m,-n}} \tag{F.5.1}$$

where a and b represent the real and imaginary contributions of the Fourier transform of the finite difference approximation G. Take the magnitude, which must satisfy

$$\begin{aligned}\left|\frac{\hat{E}^{t+1}_{mn}}{\hat{E}^{t}_{mn}}\right|^2 &= \frac{\left|\hat{E}^{t+1}_{mn}\right|^2}{\left|\hat{E}^{t}_{mn}\right|^2}\\ &= \frac{\left\{1 - \theta\Delta\tau\left[-a + r + \lambda\left(1 - \hat{f}^R_{-m,-n}\right)\right]\right\}^2 + \left\{\theta\Delta\tau\left(b + \lambda\hat{f}^I_{-m,-n}\right)\right\}^2}{\left\{1 + (1-\theta)\,\Delta\tau\left[-a + r + \lambda\left(1 - \hat{f}^R_{-m,-n}\right)\right]\right\}^2 + \left\{(1-\theta)\,\Delta\tau\left(b + \lambda\hat{f}^I_{-m,-n}\right)\right\}^2} \le 1\end{aligned} \tag{F.5.2}$$

with $\hat{f}_{-m,-n} = \left(\hat{f}^R_{-m,-n} + \hat{f}^I_{-m,-n}\sqrt{-1}\right)$ as in Equation F.3.1 and by Equation F.3.2 the term $\lambda\left(1 - \hat{f}^R_{-m,-n}\right) > 0$ when $\lambda > 0$.

Remark F.5.1 *The term $(-a + r)$ has a different sign in the numerator and denominator of Equation F.5.2. Note that $r, \lambda > 0$. If $a \leq 0$ then Equation F.5.2 is satisfied for $\theta = 0$ and $\theta = 1/2$ and Theorem 2.5.11 is proved: the time step is unconditionally strictly stable in the l_2 norm by von Neumann analysis.*

The Fourier transform of fixed point iteration Equation F.4.5 must be arranged into a complex valued form for a single coefficient E^k_{mn}. This coefficient must decrease in the iteration. This supplies the condition

$$\begin{aligned}
\frac{\left|\hat{E}^{k+1}_{mn}\right|^2}{\left|\hat{E}^{k}_{mn}\right|^2} &= \left|\frac{(1-\theta)\Delta\tau\,\lambda\hat{f}}{1 + (1-\theta)\,\Delta\tau\,(-a + r + \lambda) - (1-\theta)\,\Delta\tau b\sqrt{-1}}\right|^2 \\
&\leq \frac{\left[(1-\theta)\Delta\tau\,\lambda\right]^2}{\left[1 + (1-\theta)\,\Delta\tau\,(-a + r + \lambda)\right]^2 + \left[\sqrt{-1}\,(1-\theta)\,b\right]^2} \\
&\leq \frac{\left[(1-\theta)\Delta\tau\,\lambda\right]^2}{\left[1 + (1-\theta)\,\Delta\tau\,(-a + r + \lambda)\right]^2} < 1\ .
\end{aligned} \tag{F.5.3}$$

Note that Equation F.3.2 has been used again and that the a of Equation F.5.3 is identical to that of Equation F.5.2.

Remark F.5.2 *For $\theta = 0$ and $\theta = 1/2$ it is sufficient, using Equation F.5.3, to demonstrate that $a \leq 0$ for the error magnitude to be reduced at each iteration. Thus if $a \leq 0$ then Theorem 2.6.3 is proved: the fixed point iteration is unconditionally convergent in the l_2 norm by von Neumann analysis.*

In both Remark F.5.1 and F.5.2, the stability or, respectively, convergence, is dependent only on the real component of the Fourier transform of the finite difference approximation used to generate G. It shall be shown that this depends, in turn, only on the discrete form of the diffusion term.

F.6 Analysis of the 7-pt Stencil

To prove the seven-point stencil variant of Remarks F.5.1 and F.5.2, expand the partial derivative term of Equation F.4.1 in finite differences in Fourier space. Use the cross-partial derivative

Equation E.1.5, which normally would apply for $0 < \rho \leq 1$,

$$\begin{aligned}
\Big[v_1\, \hat{\partial}_1^{h_1} + v_2\, \hat{\partial}_2^{h_2} + d_{11}\, \hat{\partial}_{11}^{{h_1}^2} + d_{22}\, \hat{\partial}_{22}^{{h_2}^2} + 2\, d_{12}\, \hat{\partial}_{12}^{h_1 h_2 +}\Big] \hat{U}_{mn} \\
= 2 \Bigg\{ \left[\frac{v_1}{h_1} \sin(\zeta m) + \frac{v_2}{h_2} \sin(\zeta n) \right] \sqrt{-1} \\
+ \frac{d_{11}}{{h_1}^2} \left[\cos(\zeta m) - 1\right] + \frac{d_{22}}{{h_2}^2} \left[\cos(\zeta n) - 1\right] \\
+ \frac{d_{12}}{h_1 h_2} \left[1 + \cos(\zeta m + \zeta n) - \cos(\zeta m) - \cos(\zeta n)\right] \Bigg\} \xi_p\, \hat{U}_{mn} \\
= 2 \left\{ a + b\sqrt{-1}\ \right\} \xi_p\, \hat{U}_{mn}
\end{aligned} \tag{F.6.1}$$

where v_1, v_2 are the coefficients of the first partial derivatives and d_{11}, d_{12}, d_{22} are the coefficients of the diffusion terms. From this expression

$$\begin{aligned}
a = \left(\frac{\sigma_1}{h_1}\right)^2 \left[\cos(\zeta m) - 1\right] + \left(\frac{\sigma_2}{h_2}\right)^2 \left[\cos(\zeta n) - 1\right] \\
+ \rho \frac{\sigma_1}{h_1} \frac{\sigma_2}{h_2} \left[1 + \cos(\zeta m + \zeta n) - \cos(\zeta m) - \cos(\zeta n)\right] \ .
\end{aligned} \tag{F.6.2}$$

Differentiate a to find its extrema by solving for the simultaneous zeros of

$$\begin{aligned}
\frac{\partial a}{\partial(\zeta m)} &= -\left(\frac{\sigma_1}{h_1}\right)^2 \sin(\zeta m) + \rho \frac{\sigma_1}{h_1} \frac{\sigma_2}{h_2} \left(\sin(\zeta m) - \sin(\zeta m + \zeta n)\right) = 0 \\
\frac{\partial a}{\partial(\zeta n)} &= -\left(\frac{\sigma_2}{h_2}\right)^2 \sin(\zeta n) + \rho \frac{\sigma_1}{h_1} \frac{\sigma_2}{h_2} \left(\sin(\zeta n) - \sin(\zeta m + \zeta n)\right) = 0 \ .
\end{aligned}$$

There are four solutions at $(\zeta m, \zeta n) = (\{0, \pi\}, \{0, \pi\})$. Of these, the maximum is $a = 0$ at $(0, 0)$ where all $\cos(\cdot)$ terms of a equal 1. Thus $a \leq 0$ and Remarks F.5.1 and F.5.2 hold.

The case for $-1 \leq \rho < 0$, using the finite difference $\hat{\partial}_{12}^{h_1 h_2 -}$ for the cross-partial term (Equation E.1.6), differs only in the leading sign of the d_{12} term and the sign of ζn in the first cosine term. The final result is the same: $a \leq 0$ unconditionally. In neither case is there a time step or correlation sign restriction for strict stability in the l_2 norm and the coefficients of the advection term do not appear in the final condition.

F.7 Analysis of the 9-pt Stencil

To prove the nine-point stencil variant of Remarks F.5.1 and F.5.2, expand the partial derivative term of Equation F.4.1 in finite differences in Fourier space. Use the cross-partial derivative

Equation E.1.7:

$$\begin{aligned}\Big[v_1\,\hat{\partial}_1^{h_1} + v_2\,\hat{\partial}_2^{h_2} + d_{11}\,\hat{\partial}_{11}^{{h_1}^2} + d_{22}\,\hat{\partial}_{22}^{{h_2}^2} + 2\,d_{12}\,\hat{\partial}_{12}^{h_1 h_2 \circ}\Big]\hat{U}_{mn} & \\ = 2\bigg\{\bigg[\frac{v_1}{h_1}\sin(\zeta m) + \frac{v_2}{h_2}\sin(\zeta n)\bigg]\sqrt{-1} & \\ + \frac{d_{11}}{{h_1}^2}\left[\cos(\zeta m) - 1\right] & \\ + \frac{d_{22}}{{h_2}^2}\left[\cos(\zeta n) - 1\right] & \\ - \frac{d_{12}}{h_1 h_2}\sin(\zeta m)\sin(\zeta n)\bigg\}\,\xi_p\,\hat{U}_{mn} & \\ = 2\left\{a + b\sqrt{-1}\,\right\}\xi_p\,\hat{U}_{mn} &\end{aligned} \qquad \text{(F.7.1)}$$

where v_1, v_2 are the coefficients of the first partial derivatives and d_{11}, d_{12}, d_{22} are the coefficients of the diffusion terms. This expression supplies the requirement that

$$a = \left(\frac{\sigma_1}{h_1}\right)^2\left[\cos(\zeta m) - 1\right] + \left(\frac{\sigma_2}{h_2}\right)^2\left[\cos(\zeta n) - 1\right] + \rho\frac{\sigma_1}{h_1}\frac{\sigma_2}{h_2}\left[\sin(\zeta m)\sin(\zeta n)\right] \leq 0\ . \qquad \text{(F.7.2)}$$

Again, proceed as in the previous section, differentiating a and finding the zeros which solve

$$\begin{aligned}\frac{\partial a}{\partial(\zeta m)} &= -\left(\frac{\sigma_1}{h_1}\right)^2\sin(\zeta m) + \rho\frac{\sigma_1}{h_1}\frac{\sigma_2}{h_2}\left(\cos(\zeta m)\sin(\zeta n)\right) = 0 \\ \frac{\partial a}{\partial(\zeta n)} &= -\left(\frac{\sigma_2}{h_2}\right)^2\sin(\zeta n) + \rho\frac{\sigma_1}{h_1}\frac{\sigma_2}{h_2}\left(\sin(\zeta m)\cos(\zeta n)\right) = 0\ .\end{aligned}$$

There are four solutions at $(\zeta m, \zeta n) = (\{0, \pi\}, \{0, \pi\})$. Of these, $(0, 0)$ is the maximum with $a = 0$. Thus $a \leq 0$ and Remarks F.5.1 and F.5.2 hold.

Appendix G

Notation

The details for some notation change for each specific, concrete example used in the main Chapters 2, 3 and 4. For example, although operator $\mathcal{G}$ always represents a continuous partial differential operator and G represents its discrete counterpart, these will have different specific definitions between Chapter 2 for two-asset problems and Chapter 3 for stochastic volatility problems. For each main chapter the specific valuation problem does not change, so symbols do not change their detailed definition.

Values, Domains and Coordinates

(S_1, S_2)	Spatial co-ordinates for two-asset problems, price scaling.
(x_1, x_2)	Spatial co-ordinates for two-asset problems, log-price scaling.
(S, v)	Spatial co-ordinates for stochastic volatility models, price scaling.
(x, v)	Spatial co-ordinates for stochastic volatility models, log-price scaling.
τ	Time remaining to contract expiry where $\tau \in [0, T]$.
Ω_∞	Two-asset unbounded spatial solution domain $[-\infty, \infty] \times [-\infty, \infty]$.
$\Omega_{\overline{\infty}}$	Stochastic vol. unbounded spatial solution domain $[-\infty, \infty] \times [0, \infty]$.
Ω_D	The finite spatial solution domain $\Omega_D \subset \Omega_\infty$.
Ω_D^*	The finite spatial solution domain for the Discrete Fourier Transform, $\Omega_D \subset \Omega_D^*$.
Ω_D°	The finite spatial solution domain for the Discrete Fourier Transform in periodic form.
Ω_C	The core spatial solution domain $\Omega_C \subset \Omega_D$.
$\partial\Omega_C,\ \partial\Omega_D$	Finite domain boundaries.
Q	Policy parameter for an operator in the HJB PIDE case.
U	Continuous solution over infinite domain Ω_∞.
V	Continuous solution over finite domain Ω_D.
W^n	Semi-discrete, continuous solution over finite domain Ω_D at time step τ_n (Chapter 2).

Continuous Functions and Operators: Script Font $\mathcal{A}, \mathcal{B}, \ldots$

$\mathcal{L}$	Continuous linear operator with partial differential and source terms.
$\mathcal{G}$	Continuous linear operator with partial differential terms only.
$\mathcal{H}$	Integral operator, including point-wise and drift-correction terms.
$\mathcal{I}$	Initial condition of the PIDE, usually the option payoff.
$\mathcal{J}$	Integral operator, first correlation term only.
$\tilde{\mathcal{L}}$	A tilde indicates an operator, function or value defined over $S = e^x$ co-ordinates (price scaling).
$\mathcal{J}_D$	Subscripted operators are truncated to a finite domain, in this case Ω_D
$\mathcal{H}_D(Q)$	An operator subject to a control vector, in this case Q.

Discrete Operators and Values: Sans-serif Font $\mathsf{A}, \mathsf{a}, \ldots$

$\mathsf{A}, \mathsf{A_{ij}}$	Matrices are denoted in upper case sans-serif font, optionally with subscripted indices.
$\mathsf{v}, \mathsf{v_i}$	Vectors are denoted in lower case sans-serif font, optionally with a subscripted index.
D	Diagonal matrix formed from a term in integral operator $\mathcal{H}$.
G	Finite difference matrix of partial differential terms.
F	Fast Fourier transform computation of a correlation integral.
I	Identity matrix.
$\mathsf{I_c}$	Modified identity matrix to select nodes in Ω_C.
J	Dense matrix corresponding to the correlation integral $\mathcal{J}$.
K, L	Interpolation operations.
T	Time step matrix, the sum of several components.
P	Finite difference matrix for the partial differential terms of operator $\mathcal{H}_D(Q)$.
c^k	Penalty vector for American option solution at iteration k.
$\mathsf{q}^n, \mathsf{q}^k$	Discrete control vector for a policy in the HJB PIDE case, at time step τ_n or iteration k respectively.
$\mathsf{R}^n, \mathsf{R}^k$	Discrete policy selection matrix, at time step τ_n or non-linear iteration k respectively.
$\mathsf{v}^n, \mathsf{w}^n$	Time step solution vectors at time step $\tau = \tau_n$.
z^k	Intermediate, iterate values at iteration $k = 0 \ldots$

Miscellaneous Values

N	The number of finite difference grid points.
$N_F = N_1 \cdot N_2$	The number of discrete Fourier transform grid points, total and in each dimension.
N_τ	The number of time steps after the initial value.
N_q	Node count in discrete policies for one control dimension.
N_Y	The total number of discrete, uncertain parameter policies.
$(x_1, x_2)_\mathsf{i}$	Discrete spatial co-ordinates for two-asset problems, log-price scaling $\mathsf{i} \in 1 \dots N$. Used in place of $(x_{(1,\mathsf{i})}, x_{(2,\mathsf{i})})$ as $W((x_1, x_2)_\mathsf{i})$ rather than $W(x_{(1,\mathsf{i})}, x_{(2,\mathsf{i})})$ to avoid doubling subscripts.
$(x, v)_\mathsf{i}$	Discrete spatial co-ordinates for stochastic volatility problems, log-price scaling $\mathsf{i} \in 1 \dots N$.
τ_n	Discerete time to contract expiry where $\tau_n \in [0, T]$, $\tau_n = n\Delta\tau$ and $n \in 0 \dots N_\tau$.
$(\Delta x_1, \Delta x_2)$	Finite difference grid spacing for two-asset problems.
$(\Delta x, \Delta v)$	Finite difference grid spacing for stochastic volatility problems.
$(\Delta x_{f,1}, \Delta x_{f,2})$	Discrete Fourier Transform grid spacing for two-asset numerical demonstrations.
$(\Delta x_f, \Delta v_f)$	Discrete Fourier Transform grid spacing for stochastic volatility numerical demonstrations.
r	Risk free rate of return.
	Two-asset Case.
σ_1, σ_2	Brownian motion volatilities.
ρ	Brownian motion correlation between assets.
	Stochastic Volatility Case.
σ_v	Brownian motion volatility of variance.
ρ_v	Brownian motion correlation.
θ_v	Variance reversion level.
κ_v	Rate of reversion to θ_v.
	Finite Activity Jumps.
λ	Finite activity jump arrival rate, the mean of Poisson process.
$\check{\mu}_1, \check{\sigma}_1, \check{\rho}$	Jump distribution parameters are denoted with a check.

Bibliography

[1] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, no. 55 in National Bureau of Standards, Applied Mathematics Series, United States Department of Commerce, December 1972.

[2] A. Almendral and C. Oosterlee, *Numerical valuation of options with jumps in the underlying*, Applied Numerical Mathematics, 53 (2005), pp. 1–18.

[3] A. L. Amadori, *Differential and integro–differential nonlinear equations of degenerate parabolic type arising in the pricing of derivatives in incomplete market*, PhD thesis, University of Roma I - La Sapienza, 2001.

[4] K. I. Amin, *Jump diffusion option valuation in discrete time*, Journal of Finance, 48 (1993), pp. 1833–1863.

[5] L. Andersen and J. Andreasen, *Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing*, Review of Derivatives Research, 4 (2000), pp. 231–262.

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

[7] L. Bachelier, *Théorie de la spéculation*, Annales scientifiques de l'École Normale Supérieure, 3 (1900), pp. 21–86.

[8] B. Bakshi, C. Cao, and Z. Chen, *Empirical performance of alternative option pricing models*, The Journal of Finance, 52 (1997), pp. 2003–2049.

[9] G. Barles, *Convergence of numerical schemes for degenerate parabolic equations arising in finance theory*, in Numerical Methods in Finance, L. C. G. Rogers and D. Talay, eds., Cambridge University Press, June 1997.

[10] G. Barles and E. R. Jakobsen, *Error bounds for monotone approximation schemes for parabolic Hamilton-Jacobi-Bellman equations.*, Mathematics of Computation, (2007). S 0025-5718(07)02000-5. Posted on April 20, 2007. To appear in print.

[11] G. Barles and P. E. Souganidis, *Convergence of approximation schemes for fully nonlinear second order equations*, Asymptotic Analysis, 4 (1991), pp. 271–283.

[12] R. K. Beatson and L. Greengard, *A short course on fast multipole methods*, Oxford University Press, 1997, pp. 1–37.

[13] F. Black and M. Scholes, *The pricing of options and corporate liabilities*, The Journal of Political Economy, 81 (1973), pp. 637–654.

[14] J. F. Bonnans, E. Ottenwaelter, and H. Zidani, *A fast algorithm for the two dimensional HJB equation of stochastic control*, ESAIM: Mathematical Modelling and Numerical Analysis, 38 (2004), pp. 723–735.

[15] J. F. Bonnans and H. Zidani, *Consistency of generalized finite difference schemes for the stochastic HJB equation*, SIAM Journal on Numerical Analysis, 41 (2003), pp. 1008–1021.

[16] S. Boyarchenko and S. Levendorskii, *Barrier options and touch-and-out options under regular Lévy processes of exponential type*, Annals of Applied Probability, 12 (2002), pp. 1261–1298.

[17] R. N. Bracewell, *The Fourier transform and its applications*, McGraw-Hill, 3 ed., 2000.

[18] M. Briani, C. L. Chioma, and R. Natalini, *Convergence of numerical schemes for viscosity solutions to integro-differential degenerate parabolic problems arising in financial theory*, Numerische Mathematik, 98 (2004), pp. 607–646.

[19] M. Briani, R. Natalini, and G. Russo, *Implicit-explicit numerical schemes for jump–diffusion processes*, Calcolo, 44 (2007), pp. 33–57.

[20] M. Broadie, M. Chernov, and M. S. Johannes, *Model specification and risk premiums: The evidence from the futures options*, Journal of Finance, 62 (2007), pp. 1453–1490.

[21] M. Broadie and J. Detemple, *Option pricing: Valuation models and applications*, Management Science, 50 (2004), pp. 1145–1177.

[22] M. Broadie and Y. Yamamoto, *Application of the fast Gauss transform to option pricing*, Management Science, 49 (2003), pp. 1071–1088.

[23] F. Camilli, *Approximation of integro-differential equations associated with piecewise deterministic proces*, Optimal Control Applications and Methods, 18 (1998), pp. 423–444.

[24] C. Cao and G. S. Bakshi, *Risk-neutral kurtosis, jumps and option pricing: Evidence from 100 most actively traded firms on the CBOE*, in Proceedings of the European Finance Association 2003, Glasgow, Scotland, October 2003.

[25] R. Carmona and V. Durrleman, *Pricing and hedging spread options*, SIAM Review, 45 (2003), pp. 627–685.

[26] P. Carr and D. B. Madan, *Option valuation using the fast Fourier transform*, Journal of Computational Finance, (1999).

[27] Z. Chen and P. A. Forsyth, *Stochastic models of natural gas prices and applications to natural gas storage valuation*. David R. Cheriton School of Computer Science, University of Waterloo, November 2006.

[28] M. Chernov, R. Gallant, E. Ghysels, and G. Tauchen, *Alternative models for stock price dynamics*, Journal of Econometrics, 116 (2003), pp. 225–257.

[29] C. Chiarella and A. Ziogas, *Pricing American options under stochastic volatility*, in SCE: 11th Intl Conference - Computing in Economics and Finance, June 2005.

[30] S. S. Clift and P. A. Forsyth, *Monotone methods for option valuation under stochastic volatility, jump diffusion models*. David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario, February 2007.

[31] ——, *Numerical solution of two asset jump diffusion models for option valuation.*, Applied Numerical Mathematics, (2007). Available on line, forthcoming in print.

[32] S. S. Clift and W.-P. Tang, *Weighted graph based ordering techniques for preconditioned conjugate gradient methods*, BIT Numerical Mathematics, 35 (1995), pp. 30–47.

[33] R. Cont and P. Tankov, *Financial Modelling with Jump Processes*, CRC Financial Mathematics Series, Chapman and Hall, 2004.

[34] R. Cont and E. Voltchkova, *A finite difference scheme for option pricing in jump diffusion and exponential Lévy models*, SIAM Journal on Numerical Analysis, 43 (2005), pp. 1596–1626.

[35] M. A. H. Dempster and S. S. G. Hong, *Spread Option Valuation and the Fast Fourier Transform*, Springer Verlag, June 2000, pp. 203–220.

[36] Y. d'Halluin, *Numerical Methods for Real Options in Telecommunications*, PhD thesis, School of Computer Science, Faculty of Mathematics, University of Waterloo, Waterloo, Ontario, Canada, 2004.

[37] Y. d'Halluin, P. A. Forsyth, and G. Labahn, *A penalty method for American options with jump diffusion processes*, Numerische Mathematik, 97 (2004), pp. 321–352.

[38] Y. d'Halluin, P. A. Forsyth, and K. R. Vetzal, *Robust numerical methods for contingent claims under jump diffusion processes*, IMA Journal of Numerical Analysis, 25 (2005), pp. 87–112.

[39] J.-C. Duan, I. Popova, and P. Ritchken, *Option pricing under regime switching*, Quantitative Finance, 2 (2002), pp. 116–132.

[40] P. DuChateau and D. Zachmann, *Applied partial differential equations*, Dover Publications, Mineola, New York, U.S.A., 2002. (1989 edition, reprint).

[41] D. Duffie, J. Pan, and K. Singleton, *Transform analysis and asset pricing for affine jump-diffusions*, Econometrica, 68 (2000), pp. 1343–1376.

[42] A. J. W. Duijndam and M. A. Schonewille, *Nonuniform fast Fourier transform*, Geophysics, 64 (1999), pp. 539–551.

[43] B. Eraker, *Do stock prices and volatility jump? Reconciling evidence from spot and option prices*, Journal of Finance, 59 (2004), pp. 1367–1404.

[44] B. Eraker, M. Johannes, and N. Polson, *The impact of jumps in volatility and returns*, Journal of Finance, 58 (2003), pp. 1269–1300.

[45] S. Fedotov and S. Panayides, *An adaptive method for valuing an option on assets with uncertainty in volatility.* The School of Mathematics, The University of Manchester, January 2006.

[46] J. D. Fonseca, M. Grasselli, and C. Tebaldi, *Option pricing when correlations are stochastic: an analytical framework*, Working Paper RR-33, Ecole Supérieure d'Ingénieurs Léonard de Vinci, Paris, July 2006.

[47] P. A. Forsyth and G. Labahn, *Numerical methods for controlled Hamilton-Jacobi-Bellman PDEs in finance*, tech. rep., University of Waterloo, David R. Cheriton School of Computer Science, May 2006.

[48] P. A. Forsyth and K. R. Vetzal, *Quadratic convergence for valuing American options using a penalty method*, SIAM Journal on Scientific Computing, 23 (2002), pp. 2095–2122.

[49] P. A. Forsyth, K. R. Vetzal, and R. Zvan, *A finite element approach to the pricing of discrete lookbacks with stochastic volatility*, Applied Mathematical Finance, 6 (1999), pp. 87–106.

[50] N. C. Framstad, B. Oksendal, and A. Sulem, *Sufficient stochastic maximum principle for the optimal control of jump diffusions and applications to finance*, Journal of Optimization Theory and Applications, 121 (2004), pp. 77–98.

[51] A. Friedman, *Partial Differential Equations of Parabolic Type*, Prentice-Hall, Englewood Cliffs, NJ, U.S.A., 1964.

[52] K. O. Friedrichs, *The identity of weak and strong extensions of differential operators*, Transactions of the American Mathematical Society, 55 (1944), pp. 132–151.

[53] P. Friz and J. Gatheral, *Valuing volatility derivatives as an inverse problem*, Quantitative Finance, 5 (2005), pp. 531–542.

[54] M. G. Garroni and J. L. Menaldi, *Green functions for second order parabolic integro-differential problems*, no. 275 in Pitman Research Notes in Mathematics, Longman Scientific and Technical, Harlow, Essex, UK, 1992.

[55] E. Ghysels, M. Chernov, A. R. Gallant, and G. Tauchen, *A new class of stochastic volatility models with jumps: Theory and estimation*, Technical Report 99s-48, CIRANO, November 1999.

[56] L. Greengard and X. Sun, *A new version of the fast Gauss transform*, Documenta Mathematica, Extra Volume ICM III (1998), pp. 353–364.

[57] W. Hackbusch, *A sparse matrix arithmetic based on H-matrices. part I: Introduction to H-matrices*, Computing, 62 (1999), pp. 89–108.

[58] W. Hackbusch and B. N. Khoromskij, *A sparse H-matrix arithmetic. Part II: Application to multi-dimensional problems*, Computing, 64 (2000), pp. 21–47.

[59] C. He, J. Kennedy, T. Coleman, P. A. Forsyth, Y. Li, and K. Vetzalk, *Calibration and hedging under jump diffusion*, Review of Derivatives Research, 9 (2006), pp. 1–35.

[60] S. L. Heston, *A closed-form solution for options with stochastic volatility with applications to bond and currency options*, Review of Financial Studies, 6 (1993), pp. 327–343.

[61] J. Huang and L. Wu, *Specification analysis of option pricing models based on time-changed Lévy processes*, The Journal of Finance, 59 (2004), pp. 1405–1440.

[62] J. Hull and A. White, *Valuing derivative securities using the explicit finite difference method*, Journal of Financial and Quantitative Analysis, 25 (1990), pp. 87–100.

[63] S. Ikonen and J. Toivanen, *Componentwise splitting methods for pricing American options under stochastic volatility*, International Journal of Theoretical and Applied Finance, 10 (2007), pp. 331–361.

[64] ——, *Efficient numerical methods for pricing American options under stochastic volatility*, Numerical Methods for Partial Differential Equations, (2007). Online, forthcoming in print.

[65] Y. Ishikawa, *Optimal control problem associated with jump processes*, Applied Mathematics and Optimization, 50 (2004), pp. 21–65.

[66] E. R. Jakobsen, *On the rate of convergence of approximation schemes for Bellman equations associated with optimal stopping time problems*, Mathematical Models and Methods in Applied Sciences (M3AS), 13 (2003), pp. 613–644.

[67] E. R. Jakobsen and K. H. Karlsen, *A "maximum principle for semicontinuous functions" applicable to integro-partial differential equations*, NoDEA : Nonlinear Differential Equations and Applications, 13 (2006), pp. 137–165.

[68] E. R. Jakobsen, K. H. Karlsen, and C. L. Chioma, *Error estimates for approximate solutions to Bellman equations associated with controlled jump-diffusions.* Department of Mathematical Sciences, Norwegian University of Science and Technology, October 2006.

[69] R. Kangro and R. Nicolaides, *Far field boundary conditions for Black–Scholes equations*, SIAM Journal on Numerical Analysis, 38 (2000), pp. 1357–1368.

[70] J. S. Kennedy, P. A. Forsyth, and K. R. Vetzal, *Dynamic hedging under jump diffusion with transaction costs.* Submitted to Operations Research., April 2007.

[71] S. Kotz, *Continuous multivariate distributions, 2 ed.*, vol. 1, Wiley, New York, 2000.

[72] S. G. Kou, *A jump-diffusion model for option pricing*, Management Science, 48 (2002), pp. 1086–1101.

[73] J. F. B. M. Kraaijevanger, H. W. J. Lenferink, and M. N. Spijker, *Stepsize restrictions for stability in the numerical solution of ordinary and partial differential equations*, Journal of Computational and Applied Mathematics, (1987), pp. 67–81.

[74] N. Krylov, *Approximating value functions for controlled degenerate diffusion processes by using piece-wise constant policies*, Electronic Journal of Probability, 4 (1999), pp. 1–19.

[75] P. K. Lamm, *A survey of regularization methods for first-kind Volterra equations*, in Surveys on Solution Methods for Inverse Problems, D. Colton, H. W. Engl, A. Louis, J. R. McLaughlin, and W. Rundell, eds., Springer, Vienna, New York, 2000, pp. 53–82.

[76] R. Lee, *Option pricing by transform methods: Extensions, unification, and error control*, Journal of Computational Finance, 7 (2005), pp. 51–86.

[77] H. W. J. Lenferink and M. N. Spijker, *On the use of stability regions in the numerical analysis of initial value problems*, Mathematics of Computation, 57 (1991), pp. 221–237.

[78] A. L. Lewis, *A simple option formula for general jump-diffusion and other exponential Lévy processes*, tech. rep., Envision Financial Systems and OptionCity.net, August 2001.

[79] ——, *Fear of jumps*, Wilmott Magazine, (2002), pp. 60–67.

[80] A. W. Marshall and I. Olkin, *A multivariate exponential distribution*, Journal of the American Statistical Association, 62 (1967), pp. 30–44.

[81] S. H. Martzoukos, *Contingent claims on foreign assets following jump-diffusion processes*, Review of Derivatives Research, 6 (2003), pp. 27–45.

[82] A.-M. Matache, P.-A. Nitsche, and C. Schwab, *Wavelet Galerkin pricing of American options on Lévy driven assets*, Quantitative Finanace, 5 (2005), pp. 403–424.

[83] A.-M. Matache, C. Schwab, and T. P. Wihler, *Fast numerical solution of parabolic integrodifferential equations with applications in finance*, SIAM Journal on Scientific Computing, 27 (2005), pp. 369–393.

[84] A.-M. Matache, T. von Petersdorff, and C. Schwab, *Fast deterministic pricing of options on Lévy driven assets*, Mathematical Modelling and Numerical Analysis, 38 (2004), pp. 37–72.

[85] R. C. Merton, *Theory of rational option pricing*, The Bell Journal of Economics and Management Science, 4 (1973), pp. 141–183.

[86] ——, *Option pricing when underlying stock returns are discontinuous*, Journal of Financial Economics, 3 (1976), pp. 125–144.

[87] M. Mnif and A. Sulem, *Optimal risk control and dividend policies under excess of loss reinsurance*, Stochastics: An International Journal of Probability and Stochastic Processes, 77 (2005), pp. 455–476.

[88] S. Mulinacci, *An approximation of American option prices in a jump-diffusion model*, Stochastic Processes and their Applications, 62 (1996), pp. 1–17.

[89] B. Oksendal and A. Sulem, *Applied Stochastic Control of Jump Diffusions*, Springer Verlag, Berlin, 2000.

[90] A. Oztukel and P. Wilmott, *Uncertain parameters, an empirical stochastic volatility model and confidence limits*, International Journal of Theoretical and Applied Finance, 1 (1998), pp. 175–189.

[91] H. Pham, *Optimal stopping of controlled jump-diffusion processes: a viscosity solutions approach*, Journal of Mathematical Systems, Estimation and Control, 8 (1998), pp. 1–27.

[92] H. X. Pham, *On some recent aspects of stochastic control and their applications*, Probability Surveys, 2 (2005), pp. 506–549.

[93] R. J. Plemmons, *M-matrix characterizations. I–nonsingular M-matrices*, Linear Algebra and its Applications, 18 (1977), pp. 175–188.

[94] 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 (2003), pp. 241–267.

[95] D. M. Pooley, P. A. Forsyth, and K. R. Vetzal, *Two factor option pricing with uncertain volatility*, in Lecture Notes in Computer Science, Springer, January 2003, pp. 158–167. Volume 2669 / 2003, ICCSA 2003: International Conference.

[96] D. Potts, G. Steidl, and M. Tasche, *Fast Fourier Transforms for Nonequispaced Data: A Tutorial*, Applied and Numerical Harmonic Analysis Series, Birkhauser, Boston, 2001, ch. 12, pp. 249–274.

[97] R. D. Richtmyer and K. W. Morton, *Difference methods for initial-value problems, 2nd edition*, Interscience Publishers, New York, 1967.

[98] Y. Saad, *Iterative Methods for Sparse Linear Systems (2nd ed.)*, Society for Industrial and Applied Mathematics, Philadelphia, PA, U.S.A., January 2003.

[99] W. Schoutens, E. Simons, and J. Tistaert, *A perfect calibration! Now what?*, Wilmott Magazine, (2004), pp. 66–78.

[100] A. Sepp, R. Kangro, and K. Parna, *Pricing European-style options under jump diffusion processes with stochastic volatility: Applications of Fourier transform*, Acta et Commentationes Universitatis Tartuensis de Mathematica, 8 (2004), pp. 123–133.

[101] M. N. Spijker, *Stepsize restrictions for stability of one-step methods in the numerical solution of initial value problems*, Mathematics of Computation, 45 (1985), pp. 377–392.

[102] F. Straetemans, *Resolvent conditions for discretizations of diffusion-convection-reaction equations in several space dimensions*, Applied Numerical Mathematics, 28 (1998), pp. 45–67.

[103] R. M. Stulz, *Options on the minimum or the maximum of two risky assets*, Journal of Financial Economics, 10 (1982), pp. 161–185.

[104] E. Tzavalis and S. Wang, *Pricing American options under stochastic volatility: A new method using Chebyshev polynomials to approximate the early exercise boundary*, Working Papers 488, Queen Mary, University of London, Department of Economics, 2003.

[105] J. Wang and P. A. Forsyth, *Maximal use of central differencing for Hamilton-Jacobi-Bellman PDEs in finance*. Submitted to SIAM Journal on Numerical Analysis, November 2006.

[106] A. F. Ware, *Fast approximate Fourier transforms for irregularly spaced data*, SIAM Review, 40 (1998), pp. 838–856.

[107] R. E. Whaley and G. Barone-Adesi, *Efficient analytic approximation of American option values*, Journal of Finance, 42 (1987), pp. 301–320.

[108] H. A. Windcliff, P. A. Forsyth, and K. R. Vetzal, *Analysis of the stability of the linear boundary condition for the Black–Scholes equation*, Journal of Computational Finance, 8 (2004), pp. 65–92.

[109] C. Yang, R. Duraiswami, and N. Gumerov, *Improved fast Gauss transform.* Technical Report CS-TR-4495, Dept. of Computer Science, University of Maryland, College Park, 2003.

[110] X. L. Zhang, *Numerical analysis of American option pricing in a jump-diffusion model*, Mathematics of Operations Research, 22 (1997), pp. 668–690.

[111] R. Zvan, P. A. Forsyth, and K. R. Vetzal, *Penalty methods for American options with stochastic volatility*, Journal of Computational and Applied Mathematics, 91 (1998), pp. 199–218.

[112] ——, *A finite volume approach for contingent claims valuation*, IMA Journal of Numerical Analysis, 21 (2001), pp. 703–731.

[113] R. Zvan, P. A. Forsyth, and K. R. Vetzal, *Negative coefficients in two-factor option pricing models*, Journal of Computational Finance, 7 (2003), pp. 37–73.