Wraparound error in Fourier Space Timestepping

by

Durgesh Saraph

A paper
presented to the University of Waterloo
in fulfillment of the
research requirement for the degree of
Master of Mathematics
in
Computational Mathematics

Waterloo, Ontario, Canada, 2010

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

Spectral methods are a class of methods for solving partial differential equations (PDEs) or partial integrodifferential equations (PIDEs) using the Fourier transform. Numerous option valuation problems in quantitative finance can be formulated as PDEs or PIDEs. As a result, spectral methods can be applied to the valuation of options. This paper investigates the application of a Fourier space timestepping (FST) method in option valuation. Wraparound errors that affect spectral methods are studied intensively.

Acknowledgements

I would like to thank my supervisors, Peter Forsyth and George Labahn, for their help and guidance this summer, and my second reader, Justin Wan, for reading this paper and providing valuable feedback. I would also like to thank all members of the Computational Mathematics department for their support throughout this year.

Contents

List of Tables

Chapter 1

Introduction

Complex structured products and exotic derivatives have become increasingly common in the financial markets. Practitioners require fast and accurate prices and sensitivities when valuing and hedging these products. Increasing complexity in financial models and contracts has led to a demand for efficient computational methods in modelling. Furthermore, since most vanilla options traded in markets are American options and practical pricing and hedging algorithms must be able to calibrate to vanilla options, fast and accurate methods of valuing American options can be extremely useful.

An option is a financial instrument that gives the holder the right but not the obligation to enter into a transaction by a specific date at a specific price. The specified date is known as the expiry or maturity. The asset on which the transaction is specified is known as the underlying. The price at which the holder may choose to transact is known as the exercise or strike price. An option that gives the holder the right to buy an asset is known as a call option whereas one that gives the holder the right to sell an asset is known as a put option. American options may be exercised at any point of time until expiry whereas European options may only be exercised at expiry. Besides American and European options, there are more complicated options that are often known as exotic options. For example, the payoff of an Asian option is stipulated as some average of the underlying prices over a certain period. The payoff of a lookback option depends on the minimum or maximum price over a given period. Barrier options can become active or worthless if the price of the underlying asset reaches a specified value prior to maturity. Asian options, lookbacks, and barrier options are a few examples of exotic options.

Using replicating portfolios and risk-neutral valuation, it can be shown that the value of any European option can be written as the expected value of its discounted payoff. Any one of various numerical methods can then be applied to determine the price of the option. These include numerical integration, Monte Carlo simulation, and direct numerical solution of the corresponding partial differential equation (PDE) or partial integro-differential equation (PIDE).

The pricing of various financial options can be modelled as a PDE or PIDE. Black, Scholes and Merton (BSM) demonstrated that by assuming that asset prices follow a geometric Brownian motion (GBM) and using a replicating portfolio strategy the problem of pricing financial options can be reduced to that of solving a PDE. Specifically, the Black Scholes Merton model is equivalent to solving a second-order parabolic PDE in two independent variables, namely asset price and time. The PDE can be used to price path-dependent options by introducing early exercise constraints or changing terminal conditions or boundary conditions.

However, assuming that asset prices follow a GBM as per the Black Scholes Merton model in pricing options leads to inconsistencies with observed market prices, such as the implied volatility smile. Three main lines of research attempt to remove pricing biases. One line of research aims to model correlations between prices and volatility [7, 9, 5, 13, 21]. A second line of research models volatility as a continuous time stochastic process [12] or assumes volatility undergoes regime changes [20]. A third line of research includes jumps in the price process leading to jump diffusion models such as the Merton model [19, 17]. Each of these three lines of research has its own advantages and disadvantages. All three approaches aim to resolve different aspects of the implied volatility surface.

Jackson et al. in their paper focus on pricing options where the underlying assets follow a Levy process. They introduce a numerical method using the Fourier transform for solving the pricing PIDE. They test their algorithm on various European and path-dependent options and aim to improve the convergence order for American options via a penalty method [14]. They also incorporate regime changes into their model.

PDEs and PIDEs can be solved numerically using a variety of methods, including finite difference schemes and spectral methods. Spectral methods involve applying a Fourier tranform to the original equation to convert it into a set of ordinary differential equations (ODEs) in Fourier space, solving the system of ODEs in Fourier space, and applying an inverse transform to obtain the solution to the PDE or PIDE in the original space.

Under the BSM model the option pricing problem is reduced to solving a PDE in two underlying variables, namely asset price and time. Under jump diffusion models the option pricing problem involves solving a PIDE with a non-local integral term. While a diverse array of finite difference methods for solving PIDEs have been proposed in literature, all of these methods handle the diffusion and integral terms of the PIDE separately. They also require several approximations. For instance, small jumps in the infinite activity Levy process are approximated by a diffusion and incorporated into the diffusion term. The integral term must be localized so that large jumps are truncated, assumptions must be made regarding the behaviour of the option outside the solution domain, and the diffusion and integral terms are often treated separately so that function values must be interpolated between the two grids in order to compute the convolution term. As a result, finite difference methods for option pricing under jump models, particularly for pricing path-dependent options, can be significantly complex and hence susceptible to accuracy and stability issues.

One solution to the accuracy and stability issues is provided by Wang, Wan and Forsyth [24]. They develop an implicit discretization method for pricing American options when the underlying asset follows an infinite activity Levy process. They obtain quadratic convergence for processes of finite variation. Wang, Wan, and Forsyth also obtain better than first-order accuracy for infinite variation processes. They treat the jump component near a log jump size of zero using a Taylor approximation and apply a semi-Lagrangian scheme to the drift term of the stochastic process. They solve the PIDE using a preconditioned BiCGSTAB method combined with a discrete Fourier transform, and prove the stability and monotonicity of fully implicit timestepping.

Jackson et al. propose a different solution for pricing options under a general Levy process such as a Merton jump diffusion or CGMY process. They present a Fourier space timestepping (FST) algorithm for option pricing under a Levy process with and without regime changes [14]. The algorithm utilizes Fourier transforms to transform the PIDE into Fourier space.

The FST algorithm has several significant advantages in pricing options. By applying the Fourier transform to the PDE or PIDE one can obtain a linear system of ordinary differential equations (ODEs), because the characteristic exponent of an independent increment stochastic process can be factored out of the Fourier transform of the PIDE. The Levy-Khintchine formula provides a closed form for the characteristic exponents of all independent increment stochastic processes, so the FST method can be used to price contingent claims on any exponential-Levy price process without modifying the algorithm, even in cases where an explicit formulation of the probability density function of the underlying stochastic process does not exist. Moreover, he FST algorithm treats the jump and diffusion terms in a symmetric manner and avoids any strong assumptions on the option price outside the restricted domain. It can be extended and applied to the pricing of multidimensional and exotic path dependent options.

The FST algorithm can generate option prices for a range of spots in a single time step for European options and other path independent options. It can easily be applied to non-standard payouts. It can handle American and barrier features in exotic path dependent options and produce exact results between monitoring times. Unlike finite difference methods, it can project prices between monitoring times in one step and hence does not require timestepping between monitoring times.

A method similar to the FST algorithm was developed simultaneously by Lord et al. and called Convolution [16]. Lord et al. use the convolution representation of the PDE or PIDE to derive an option pricing method. Jackson et al. show that their FST algorithm can be used to derive an analogous penalty method for American options and the method can be extended to a regime-switching setting [14]. They demonstrate through numerical experiments that when applied to single-asset options, the order of convergence of the FST algorithm is quadratic in space and linear in time.

Spectral methods are usually applicable on a strictly periodic domain. Since the price of the underlying security does not constitute a periodic domain, spectral methods should not be directly applicable to option pricing, particularly at the boundaries. Applying a Fourier transform to option pricing involves implicitly assuming that the domain can be converted into a periodic domain by concatenating it to itself repeatedly to the left and right. The period of the new periodic domain is then the length of the original domain, or $S_{max} - S_{min}$. Using a truncated finite discrete grid to represent a price domain that is in reality infinite and continuous leads to wraparound errors near the boundaries.

The principal aims of this paper are as follows:

- Use a Fourier transform approach to derive the solution to the European and American option pricing problems under the Black Scholes Merton model and the Merton jump diffusion model.
- Implement the Fourier space timestepping (FST) method described by Jackson et al. in [14] and verify that the results correspond to those of Jackson et al.
- Investigate any adverse effects of wraparound errors in the FST method and suggest alternatives.

Chapter 2

Analytical Solution

2.1 Black Scholes Merton Model

When using the Black Scholes Merton model, we assume that the underlying asset price follows a geometric Brownian motion (GBM). The stochastic process for the underlying asset price S can be written as

$$\frac{dS}{S} = \mu dt + \sigma dZ \tag{2.1}$$

where μ is the drift parameter and σ is the diffusion parameter. Using replicating portfolios and risk-neutral valuation and applying Ito's Lemma to the above stochastic process as described by Black and Scholes [1] yields the Black Scholes Merton option pricing partial differential equation (PDE)

$$V_t + \frac{1}{2}\sigma^2 S^2 V_{SS} + rSV_S - rV = 0. \tag{2.2}$$

The pricing PDE can be expressed concisely using operator notation. Let

$$\mathcal{L}[V] = \frac{1}{2}\sigma^2 S^2 V_{SS} + rSV_S - rV. \tag{2.3}$$

Then we have the following PDE:

$$V_t + \mathcal{L}[V] = 0. \tag{2.4}$$

where V is the option value, S is the underlying asset price, t is time, r is the interest rate, and σ is the volatility. We also have the terminal condition specified by the option payoff at maturity:

$$V(T, S) = \phi(S). \tag{2.5}$$

2.2 Merton Jump Diffusion Model

Continuous models of asset price behaviour such as GBM typically lead to a complete market or a market that can be made complete by adding a few assets. In a complete market, every terminal payoff can be replicated exactly using portfolios of traded assets.

A more realistic model of asset price behaviour can be obtained by introducing jumps into the stochastic process of asset prices in addition to diffusion and drift. The earliest such jump diffusion model was introduced by Merton in 1976 [19].

Merton proposed a log-normal jump diffusion model, where the logarithm of the jump size is assumed to be normally distributed. There are other kinds of jump diffusion models such as double-exponential jump diffusions proposed by Kou in [15] and jump diffusions with a mixture of independent jumps; however, they are outside the scope of this paper. It is also possible to formulate stochastic volatility models that assume volatility itself is random and volatile. The most popular of these is the square root model developed by Heston [12].

Jump diffusion models have recently become popular for modelling asset price dynamics for several important reasons. The asset price process in a continuous time model such as the Black Scholes Merton model locally behaves like a Brownian motion, so that the price moves by a large amount over a short period with a very low probability. Such a model leads to option prices of short term out of the money options that are significantly lower than those observed in real markets. If jumps are added to asset prices, however, then even with a very short time to maturity, there is a non-trivial probability of the option being in the money after a sudden change in the asset price. Consequently, option prices obtained using a jump diffusion model are more consistent with observed market prices of the same options.

Furthermore, since continuous time models of asset prices lead to complete markets in which every payoff can be replicated using traded assets, options seem like redundant assets. In such a case even the existence of traded options becomes a puzzle. In real markets, however, there are jumps in asset prices, so perfect hedging of market positions becomes impossible. Options then allow market participants to hedge market positions that cannot be hedged by solely using the underlying assets. Jumps in asset prices enable the quantification of the risk of sudden asset price movements over short time intervals for risk management, a feature notably absent from the standard Black Scholes Merton model.

The Merton jump diffusion model incorporates jumps into the price process of the underlying asset via a Poisson process. It has drift and diffusion components similar to a geometric Brownian motion. Additionally, it contains a compound Poisson process to model jumps. The Poisson process is independent of the Brownian motion. The stochastic process for the underlying asset price S can be written as

$$\frac{dS}{S} = \mu dt + \sigma dZ + (\eta - 1)dq \tag{2.6}$$

where μ is the drift parameter and σ is the volatility of the diffusion in the Brownian motion part of the process, $\eta - 1$ is an impulse function, dZ is the increment of a standard Wiener process and dq is the increment of a Poisson process that is independent of the Brownian motion.

If λ is the Poisson arrival intensity then $dq = 0$ with probability $1 - \lambda dt$ and $dq = 1$ with probability λdt. The impulse function $\eta - 1$ produces a jump in the asset price from S to $S\eta$. The expected relative jump size is denoted by $\kappa = E(\eta - 1)$.

If $V(S,t)$ is the value of a contingent claim on the underlying asset price S at time t then using replicating portfolios and risk-neutral valuation and applying Ito's Lemma, it can be shown that V follows the partial integrodifferential equation (PIDE)

$$V_t + \frac{1}{2}\sigma^2 S^2 V_{SS} + (r - \lambda\kappa)SV_S - rV + \left(\lambda \int_0^\infty V(S\eta)g(\eta)d\eta - \lambda V\right) = 0 \tag{2.7}$$

where t is the time, r is the continuously compounded risk-free interest rate, and $g(\eta)$ is the probability density function of the jump amplitude. In the Merton model, the jump amplitude has a log-normal distribution. The PIDE describing the dynamics of V can be rewritten as

$$V_t + \frac{1}{2}\sigma^2 S^2 V_{SS} + (r - \lambda\kappa)SV_S - (r + \lambda)V + \lambda \int_0^\infty V(S\eta)g(\eta)d\eta = 0. \tag{2.8}$$

The terminal condition is specified by the payoff at maturity and is

$$V(T,S) = \phi(S). \tag{2.9}$$

This PIDE is similar to the Black Scholes Merton PDE in equation 2.2 with an additional correlation integral term. In fact, setting the Poisson arrival intensity λ equal to zero effectively removes jumps from the stochastic process and yields the Black Scholes Merton PDE of the previous section. Hence the Black Scholes Merton model can be seen as a special case of the Merton jump diffusion model in the absence of jumps. The PIDE formulation described in equation 2.8 is as described by d'Halluin et al. [8].

2.3 Fourier transform

The one-dimensional Fourier transform of a function $f(x)$ is defined as

$$\mathcal{F}(f(x))(k) = \hat{f}(k) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f(x)e^{-ikx}dx \tag{2.10}$$

The one-dimensional inverse Fourier transform of a function $f(x)$ is defined as

$$\mathcal{F}^{-1}(\hat{f}(k))(x) = f(x) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{f}(k) e^{ikx} dk \tag{2.11}$$

The Fourier transform of the n^{th} derivative of a function f can be calculated as

$$\mathcal{F}\left(f^{(n)}\right)(k) = (ik)^n \mathcal{F}\left(\left(f(x)\right)(k)\right) \tag{2.12}$$

In other words, differentiation in real space corresponds to multiplication in frequency space.

2.4 Convolution theorem

The convolution theorem provides one property of the Fourier transform that is useful in option pricing. It states that the Fourier transform of the convolution product of two functions f and g is equal to the product of the Fourier transforms of f and g. Let $f * g$ denote the convolution product of f and g, then

$$(f * g)(x) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f(x-y) g(y) dy \tag{2.13}$$

and the Fourier transform of the convolution product is

$$\mathcal{F}(f * g) = \mathcal{F}(f)\mathcal{F}(g). \tag{2.14}$$

2.5 Solving the PIDE analytically

We first attempt to solve the Merton jump diffusion PIDE analytically. A solution to the Merton jump diffusion PIDE also provides a solution to the Black Scholes Merton PDE when the latter is viewed as a special case of the former. Within this chapter, $\mathcal{F}[V]$ is used to denote the Fourier transform of V. In order to solve the Merton jump diffusion PIDE analytically, we will take the following steps:

1. Log-transform the equation to obtain a constant coefficient PIDE.
2. Apply a Fourier transform to the PIDE to obtain a system of ordinary differential equations (ODE).
3. Apply an inverse Fourier transform to the solution.

2.5.1 Obtaining constant coefficients

The Merton jump diffusion PIDE does not, in general, have constant coefficients. However, a logarithmic transform can be applied to convert this PIDE into a constant coefficient PIDE. Since S follows a geometric Brownian motion, then $x = \log S$ follows a standard Brownian motion. Partial derivatives with respect to S need to be determined in terms of partial derivatives with respect to x in order to implement a change of variable. Applying the chain rule yields

$$V_S = e^{-x} V_x \tag{2.15}$$

and

$$V_{SS} = e^{-2x} \left(V_{xx} - V_x \right). \tag{2.16}$$

The Merton jump diffusion PIDE is similar to the Black Scholes Merton PDE discussed previously with the additional correlation integral term

$$\int_0^\infty V(S\eta) g(\eta) d\eta. \tag{2.17}$$

Applying a logarithmic transform to the asset price S and jump amplitude η by setting $x = \log S$ and $y = \log \eta$, the integral term becomes

$$I = \int_{-\infty}^\infty V(x+y) f(y) dy \tag{2.18}$$

where, with some abuse of notation, $f(y) = g(e^y)e^y$ is the probability density of a jump of amplitude $y = \log \eta$ and $V(y) = V_{old}(e^y)$. The integral in equation 2.18 is the correlation product of $V(y)$ and $f(y)$ and can be written as

$$I = V(y) \otimes f(y). \tag{2.19}$$

The partial derivatives in equations 2.15 and 2.16 can substituted into the PIDE along with $S = e^x$, and the integral term can be replaced using equation 2.18 to yield a constant coefficient PIDE. A change of variable from S to x and η to y with abuse of notation yields the constant coefficient PIDE in x

$$V_t + \frac{1}{2}\sigma^2 (V_{xx} - V_x) + r(V_x - V) + \lambda \int_{-\infty}^\infty V(x+y) f(y) dy = 0 \tag{2.20}$$

with terminal condition

$$V(T, x) = \phi(x).$$

Since S follows the Merton jump diffusion model, η is log-normally distributed with mean of jumps $\tilde{\mu}$ and standard deviation of jumps $\tilde{\sigma}$. Consequently, $y = \log \eta$ follows the Normal distribution with mean $\tilde{\mu}$ and standard deviation $\tilde{\sigma}$. Therefore $f(y)$ is the probability density function of the Normal distribution with mean $\tilde{\mu}$ and standard deviation $\tilde{\sigma}$. The probability density is then

$$f(x) = \tfrac{1}{\sqrt{2\pi\tilde{\sigma}^2}} e^{\frac{-(x-\tilde{\mu})^2}{2\tilde{\sigma}^2}}. \tag{2.21}$$

2.5.2 Applying the Fourier transform

A Fourier transform can be applied to equation 2.20 in order to resolve the spatial derivatives on the right hand side. Taking spatial derivatives in real space is equivalent to multiplication in Fourier space. For this section and subsequent sections, we will define $i = \sqrt{-1}$.

Taking the Fourier transform of the constant coefficient PIDE yields

$$\begin{aligned}
\mathcal{F}(V_t) + \frac{1}{2}\sigma^2(\mathcal{F}(V_{xx}) - \mathcal{F}(V_x)) + r(\mathcal{F}(V_x) - \mathcal{F}(V)) + \mathcal{F}(V(y) \otimes f(y)) = 0 \\
\mathcal{F}(V_t) - \frac{1}{2}\sigma^2(\omega^2\mathcal{F}(V) + i\omega\mathcal{F}(V)) + r(i\omega\mathcal{F}(V) - \mathcal{F}(V)) + \lambda(\mathcal{F}(V) \cdot \mathcal{F}(f)) = 0 \\
\mathcal{F}(V_t) + \left[i\omega\left(r - \frac{1}{2}\sigma^2\right) - \frac{1}{2}\omega^2\sigma^2 - r\right]\mathcal{F}(V) + \lambda(\mathcal{F}(V) \cdot \mathcal{F}(f)) = 0.
\end{aligned}$$

By the convolution theorem, the Fourier transform of the convolution of $V(y)$ and $f(y)$ is equal to the product of the Fourier transforms of V and f. The Fourier transform of the probability density function $f(x)$ can be computed as follows:

$$\begin{aligned}
\mathcal{F}[f](x) =& \mathcal{F}[\tfrac{1}{\sqrt{2\pi\tilde{\sigma}^2}} e^{\frac{-(x-\tilde{\mu})^2}{2\tilde{\sigma}^2}}] \\
=& \int_{-\infty}^{\infty} f(x) e^{-2\pi i x\omega} dx \\
=& \int_{-\infty}^{\infty} \tfrac{1}{\sqrt{2\pi\tilde{\sigma}^2}} e^{\frac{-(x-\tilde{\mu})^2}{2\tilde{\sigma}^2}} e^{-2\pi i x\omega} dx \\
=& e^{i\mu\omega - \frac{1}{2}\sigma^2\omega^2} - 1.
\end{aligned} \tag{2.22}$$

The Fourier-transformed PIDE is then

$$\mathcal{F}(V_t) + \left[i\omega\left(r - \frac{1}{2}\sigma^2\right) - \frac{1}{2}\omega^2\sigma^2 + \lambda(e^{i\mu\omega - \frac{1}{2}\sigma^2\omega^2} - 1) - r\right]\mathcal{F}(V) = 0. \tag{2.23}$$

2.5.3 Ordinary Differential Equations

Applying a Fourier transform to the constant coefficient PIDE in x produces the following system of ordinary differential equations (ODE) parameterized by ω:

$$\begin{aligned} \mathcal{F}[V_t](t,\omega) + \Psi(\omega)\mathcal{F}[V](t,\omega) = 0 \\ \mathcal{F}[V](T,\omega) = \mathcal{F}[\phi](\omega) \end{aligned} \tag{2.24}$$

where

$$\Psi(\omega) = i\omega\left(r - \frac{1}{2}\sigma^2\right) - \frac{1}{2}\omega^2\sigma^2 + \lambda\left(e^{i\mu\omega - \frac{1}{2}\sigma^2\omega^2} - 1\right) - r. \tag{2.25}$$

Note that when the Poisson arrival rate of jumps λ is zero we obtain the Black Scholes Merton model and

$$\Psi(\omega) = i\omega\left(r - \frac{1}{2}\sigma^2\right) - \frac{1}{2}\omega^2\sigma^2 - r. \tag{2.26}$$

We can solve this system of ODEs by multiplying both sides of the equation by an integrating factor. We then write an expression for the solution of this system of ODEs in Fourier space given the value of $\mathcal{F}[V](t,\omega)$ at time $t_2 \leq T$ to yield the value at time $t_1 < t_2$:

$$\begin{aligned} \mathcal{F}[V](t_1,\omega) &= \mathcal{F}[V](t_2,\omega) \cdot e^{\Psi(\omega)(t_2-t_1)}(x) \\ \mathcal{F}[V](T,\omega) &= \mathcal{F}[\phi](\omega). \end{aligned} \tag{2.27}$$

Note that we solve equation 2.2 backwards in real time. Taking the inverse Fourier transform of the above equation produces

$$\begin{aligned} V(t_1,x) &= \mathcal{F}^{-1}\{\mathcal{F}[V](t_2,\omega) \cdot e^{\Psi(\omega)(t_2-t_1)}\}(x) \\ \mathcal{F}[V](T,\omega) &= \mathcal{F}[\phi](\omega). \end{aligned} \tag{2.28}$$

Chapter 3

Numerical Solution

The system of ODEs obtained analytically cannot, in general, be solved in closed form, because an explicit analytic expression for the Fourier transform of the option payout may not exist. Furthermore, path dependent features such as early exercise for American options cannot be directly resolved using the analytical method in Fourier space. Specifically, the American option pricing PIDE includes a non-linear term that cannot be handled using an analytical Fourier method. Due to the difficulties encountered in attempting an analytical solution to the Merton jump diffusion PIDE, we now attempt to solve the PIDE numerically. Within this chapter, $\mathcal{F}[V]$ is used to denote the discrete Fourier transform (DFT) of V.

In order to solve the Merton jump diffusion PIDE numerically, we will take the following steps:

1. Log-transform the equation to obtain a constant coefficient PIDE in $x = \log S$.
2. Discretize the PIDE in x to obtain an equation in x_j for $i = 1, \ldots, N$.
3. Apply a DFT to the PIDE to obtain a system of discrete ordinary differential equations (ODEs).
4. Apply an Inverse Discrete Fourier Transform (IDFT) to the solution.

We begin with the log-transformed constant coefficient PIDE in $x = \log S$ that was derived in equation 2.20 in the previous chapter:

$$V_t + \frac{1}{2}\sigma^2(V_{xx} - V_x) + r(V_x - V) + \lambda \int_{-\infty}^{\infty} V(x+y)f(y)dy = 0 \tag{3.1}$$

with terminal condition

$$V(T, x) = \phi(x).$$

3.1 Grid selection and discretization

Pricing path dependent options requires a timestepping algorithm to enforce constraints, impose boundary conditions, or optimize over a policy domain. The spatial and temporal domain must be discretized and partitioned into a finite mesh for computation. While Fourier techniques can be extended to multiple spatial dimensions, only the one-dimensional case is examined in this paper.

The time and asset price domain can be truncated and written as the real asset price space domain $\Omega = [0, T] \times [x_{min}, x_{max}]$. It can be partitioned into the finite mesh $\{t_m | m = 0, ..., M\} \times \{x_j | i = 0, ...N-1\}$ so that $t_m = m\Delta t, x_j = x_{min} + i\Delta x$ and $\Delta t = T/M, \Delta x = (x_{max} - x_{min})/(N-1)$, where $x = \log(S/S_0)$, and S_0 is a user-specified scaling factor. We use $S_0 = 100$ in our experiments.

The time and frequency domain can similarly be truncated and written as $\hat{\Omega} = [0, T] \times [\omega_{min}, \omega_{max}]$. It can then be partitioned into the finite mesh $\{t_m | m = 0, ..., M\} \times \{\omega_n | n = 0, ..., N/2\}$ where $\omega_n = n\Delta\omega$ and $\Delta\omega = 2\omega_{max}/N$. The maximum frequency is selected to be equal to the Nyquist critical frequency, so $\omega_{max} = \frac{1}{2\Delta x}$. Since $V(t, x)$ is a real-valued function, its Fourier transform is equal to the Fourier transform of its own complex conjugate at all points in the domain $\mathcal{F}[V](t, -\omega) = \mathcal{F}[V](t, \omega)$. As a result, the frequency grid needs only half as many points as the spatial grid, because the Fourier transform for negative frequencies need not be computed.

Jackson et al. provide only heuristic guidelines for optimal grid selection [14]. These are based on the Nyquist critical frequency. The Nyquist critical frequency implies a relationship between the spatial and frequency grids that can be written as $\omega_{max} \cdot (x_{max} - x_{min}) = N/2$. The transformation to logarithmic variables is chosen so that the log-transformed asset price domain is centered on $x = 0$. In order to meet this objective, it is convenient to choose $x_{min} = -x_{max}$ and therefore $\omega_{max} \cdot x_{max} = N/4$. The size of the real spatial domain $[x_{min}, x_{max}]$ needs to be selected so that it is large enough to capture the behaviour of the option value function, but small enough that the computed option price is accurate around the centre of the grid. The frequency space, $[0, \omega_{max}]$ likewise needs to be selected large enough to capture high frequencies, but not so large as to cause inaccuracies in computed option values. Jackson et al. estimate through numerical experiments that $x_{max} \in [2, 5]$ works well for diffusion models with low volatility and short maturity, whereas $x_{max} \in [4, 8]$ works better for models with high volatility or a dominant jump component [14].

Discretization on a finite grid involves truncation of the true spatial domain and introduces discretization error into the numerical algorithm. While the real spatial domain of $(0, \infty)$ is infinite, the domain used for pricing is the finite $[S_{min}, S_{max}]$, where $S_{min} > 0$ and $S_{max} < \infty$. Asset price values less than S_{min} and greater than S_{max} on the true spatial domain are not represented on the truncated discrete spatial domain. The discrete spatial domain is further transformed using the change of variable $x = \log(S/S(0))$ in order to work with a constant coefficient PDE.

3.2 Discrete Fourier transform

The discrete Fourier transform (DFT) can be defined in a manner analogous to the Fourier transform. The one-dimensional Fourier transform of a discrete function x_n is defined as

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}kn} \qquad k = 0, \dots, N-1. \tag{3.2}$$

The one-dimensional inverse discrete Fourier transform (IDFT) of a function X_k is defined as

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N}kn} \qquad n = 0, \dots, N-1. \tag{3.3}$$

Let $\bar{f} = [f_0, \dots, f_{N-1}]$ be a discrete function defined on a finite discrete grid. With some abuse of notation, the DFT of the m^{th} discrete derivative of the discrete function $\bar{f}$ can be calculated as

$$\mathcal{F}[\bar{f}^{(m)}](k) = (ik)^m \mathcal{F}[\bar{f}](k) \tag{3.4}$$

In other words, differentiation in real space corresponds to multiplication in frequency space for the discrete Fourier transform [23].

3.3 Discretizing the x variable

In order to apply a DFT to the PIDE we must first discretize it in terms of the x variable.

If we let

$$\mathcal{L}(V) = \frac{1}{2}\sigma^2(V_{xx} - V_x) + r(V_x - V) + \lambda \int_{-\infty}^{\infty} V(x+y) f(y) dy \tag{3.5}$$

then we can discretize the PIDE in the spatial variable x by applying the transformation $\mathcal{L} \to \mathcal{L}^h(x_0, \dots, x_{N-1})$.

Let D_x^h represent the discrete differentiation operator for the first partial derivative with respect to the spatial variable x and let D_{xx}^h represent the discrete differentiation operator for the second partial derivative with respect to the spatial variable x. The option value V at grid node x_j and time t is denoted by $V_j = V(t, x_j)$. The discretized form of the operator $\mathcal{L}(V)$ is denoted by $\mathcal{L}^h(V_j)$.

We set

$$\mathcal{L}^h(V_j) = \frac{1}{2}\sigma^2(D^h_{xx}V_j - D^h_x V_j) + r(D^h_x V_j - V_j) + \lambda I_j \tag{3.6}$$

where

$$I_j = I(j\Delta x) = \sum_{k=-N/2+1}^{k=N/2} V_{j+k} f_k \Delta y + O((\Delta y)^2) \tag{3.7}$$

is the discretization of the correlation integral following the method used by d'Halluin et al. [8]. It is assumed that $x_j = j\Delta x$ and $\Delta y = \Delta x$.

The discretized PIDE can then be written as the following set of equations:

$$D_t V_j + \frac{1}{2}\sigma^2(D^h_{xx}V_j - D^h_x V_j) + r(D^h_x V_j - V_j) + \lambda I_j = 0. \tag{3.8}$$

3.4 Discrete Fourier transform of PIDE

A DFT can be applied to equation 2.20 in order to facilitate the discretization of the spatial derivatives on the right hand side. Taking spatial derivatives in real space is equivalent to multiplication in Fourier space. Moreover, the DFT can be implemented efficiently using the Fast Fourier Transform (FFT) algorithm.

Taking the DFT of the constant coefficient PIDE yields

$$\begin{aligned}
\mathcal{F}(V_t) + \frac{1}{2}\sigma^2(\mathcal{F}(D^h_{xx}V_j) - \mathcal{F}(D^h_x V_j)) + r(\mathcal{F}(D^h_x V_j) - \mathcal{F}(V)) + \mathcal{F}(V(y) \otimes f(y)) = 0 \\
\mathcal{F}(V_t) - \frac{1}{2}\sigma^2(\omega_n^2 \mathcal{F}(V) + i\omega_n \mathcal{F}(V)) + r(i\omega_n \mathcal{F}(V) - \mathcal{F}(V)) + \lambda(\mathcal{F}(V) \cdot \mathcal{F}(f)) = 0 \\
\mathcal{F}(V_t) + \left[i\omega_n \left(r - \frac{1}{2}\sigma^2 \right) - \frac{1}{2}\omega_n^2\sigma^2 - r \right] \mathcal{F}(V) + \lambda(\mathcal{F}(V) \cdot \mathcal{F}(f)) = 0
\end{aligned}$$

for $j = 0, \ldots, N-1$ and $n = 0, \ldots, N/2-1$.

As in the analytic continuous case, by the convolution theorem, the DFT of the convolution of $V(y)$ and $f(y)$ is equal to the product of the DFTs of V and f. The DFT of the probability density function $f(x_j)$ is given by

$$\mathcal{F}[f](x_j) = e^{i\mu\omega_n - \frac{1}{2}\sigma^2\omega_n^2} - 1$$

for $j = 0, \ldots, N-1$ and $n = 0, \ldots, N/2-1$ analogous to the continuous case in the analytical solution. The PIDE after applying the DFT is

$$\mathcal{F}(V_t) + \left[i\omega_n \left(r - \frac{1}{2}\sigma^2 \right) - \frac{1}{2}\omega_n^2\sigma^2 + \lambda(e^{i\mu\omega_n - \frac{1}{2}\sigma^2\omega_n^2} - 1) - r \right] \mathcal{F}(V) = 0 \tag{3.9}$$

for $j = 0, \ldots, N-1$ and $n = 0, \ldots, N/2-1$. Note that if $\hat{V} = \mathcal{F}(V)$ then we can write $\hat{V}_t = \mathcal{F}(V_t)$ and solve a system of ODEs in frequency space.

3.5 Ordinary Differential Equations

Applying a DFT to the constant coefficient PIDE in x_j produces the following system of discrete ordinary differential equations (ODE) parameterized by ω_n:

$$[\mathcal{F}(V)]_t(t, \omega_n) + \Psi(\omega_n)\mathcal{F}[V](t, \omega_n) = 0 \tag{3.10}$$
$$\mathcal{F}[V](T, \omega_n) = \mathcal{F}[\phi](\omega_n)$$

where

$$\Psi(\omega_n) = i\omega_n \left(r - \frac{1}{2}\sigma^2 \right) - \frac{1}{2}\omega_n^2\sigma^2 + \lambda \left(e^{i\mu\omega_n - \frac{1}{2}\sigma^2\omega_n^2} - 1 \right) - r \tag{3.11}$$

for $n = 0, \ldots, N/2-1$.

Note that as in the continuous case, when the Poisson arrival rate of jumps $\lambda = 0$ we obtain the Black Scholes Merton model and

$$\Psi(\omega_n) = i\omega_n \left(r - \frac{1}{2}\sigma^2 \right) - \frac{1}{2}\omega_n^2\sigma^2 - r. \tag{3.12}$$

We can write an expression for the solution of this system of ODEs given the value of $\mathcal{F}[V](t, \omega_n)$ at time $t_2 \leq T$ to yield the value at time $t_1 < t_2$:

$$\mathcal{F}[V](t_1, \omega_n) = \mathcal{F}[V](t_2, \omega_n) \cdot e^{\Psi(\omega_n)(t_2 - t_1)}(x_j). \tag{3.13}$$

for $j = 0, \ldots, N-1$ and $n = 0, \ldots, N/2-1$.

Taking the inverse Fourier transform of the above equation produces

$$V(t_1, x_j) = \mathcal{F}^{-1}\{\mathcal{F}[V](t_2, \omega_n) \cdot e^{\Psi(\omega_n)(t_2 - t_1)}\}(x_j) \tag{3.14}$$

for $j = 0, \ldots, N-1$ and $n = 0, \ldots, N/2-1$.

Hence, for an option without path dependent features written on an asset that follows the Merton jump diffusion process, it is possible to find the value of the option at a specified time t_1 given the value at a later time t_2. In particular, this method can be used to find the option value at $t = 0$ given the payoff at $t = T$. The method can be formalized as the Fourier Space Timestepping (FST) algorithm developed by Jackson et al. [14].

3.6 Fourier Space Timestepping Algorithm

Let $V_j^m \equiv V(t_m, x_j)$ represent the option value $V(t,x)$ at the nodes on the partition of Ω in the real spatial domain and $\hat{V}_n^m \equiv \hat{V}(t_m, \omega_n)$ represent $\mathcal{F}[V](t,\omega)$ at the nodes on the partition of Ω in the frequency domain. V^m can be expressed as the vector $[V_0^m, \ldots, V_{N-1}^m]$ and $\hat{V}^m$ can be expressed as the vector $[\hat{V}_0^m, \ldots, \hat{V}_{N/2-1}^m]$. The frequency domain prices can be obtained from the spatial domain prices by applying the DFT to the spatial domain prices. This suggests the following procedure as described by Jackson et al. in [14]:

$$\begin{aligned}\hat{V}_n^m = \mathcal{F}[V](t_m, \omega_n) =& \sum_{k=0}^{N-1} V(t_m, x_k) e^{-i\omega_n x_k} \Delta x \\ =& e^{-i\omega_n x_{min}} \Delta x \sum_{k=0}^{N-1} V_k^m e^{-ink/N} \\ =& \alpha_n \sum_{k=0}^{N-1} V_k^m e^{-ink/N} \\ =& \alpha_n \left[\mathcal{F}[V^m]\right]_n\end{aligned} \tag{3.15}$$

where $\alpha_n = e^{-i\omega_n x_{min}} \Delta x$ and $\left[\mathcal{F}[V^m]\right]_n$ represents the nth component of the DFT of the vector V^m.

Spatial domain prices can be computed from frequency domain prices in an analogous manner using the inverse discrete Fourier transform (IDFT). We denote the IDFT by $\mathcal{F}^{-1}$ and use it to compute spatial domain prices from frequency domain prices as follows:

$$V_n^m = \left[\mathcal{F}^{-1}[\alpha^{-1} \cdot \hat{V}^m]\right]_n . \tag{3.16}$$

Note that if $\hat{V} = \mathcal{F}(V)$ then $\hat{V}_t = \mathcal{F}(V_t)$ in this notation. By combining the conversions between the spatial and frequency domains in equations 3.15 and 3.16 with the transformed PDE or PIDE from equation 3.9 in the previous section, we obtain a timestepping scheme in the frequency domain. Specifically, one step backward in time can be computed as follows:

$$\begin{aligned}V^{m-1} =& \mathcal{F}^{-1}[\alpha^{-1} \cdot \hat{V}^{m-1}] \\ =& \mathcal{F}^{-1}[\alpha^{-1} \cdot \hat{V}^m \cdot e^{\Psi \Delta t}] \\ =& \mathcal{F}^{-1}[\alpha^{-1} \cdot \alpha \cdot \mathcal{F}[V^m] \cdot e^{\Psi \Delta t}] \\ =& \mathcal{F}^{-1}[\mathcal{F}[V^m] \cdot e^{\Psi \Delta t}].\end{aligned} \tag{3.17}$$

Since α cancels in the above equation, it can be omitted from the numerical computation.

This algorithm can easily be extended to a multidimensional setting.

3.7 FST method for European options

The FST algorithm provides a method that is valid for any timestep size Δt. Truncation of the spatial and frequency domain contribute to discretization error. Path independent options such as European options can hence be priced in a single time step using the FST method as follows:

1. Set $M = 1$ and $V^1 = \phi(S(0)e^x)$.
2. Step back in time with $V^0 = \mathcal{F}^{-1}[\mathcal{F}[V^1] \cdot e^{\Psi(\omega)\Delta t}]$.

In the above method, $\Psi(\omega)$ depends on the stochastic process followed by the underlying asset. In previous sections it was determined for the Black Scholes Merton model and the Merton jump diffusion model. This method does not require timestepping and is hence computationally efficient compared to spatial PIDE methods. Numerical experiments indicate that the method achieves quadratic convergence, due to the error in the spatial discretization only.

3.8 FST penalty method for American options

The Fourier space timestepping method can be applied to price various options under a variety of different stochastic processes for the underlying asset. In particular, it can be extended for pricing American options. The American option pricing problem is a simple example of a Hamilton-Jacobi-Bellman PDE and as such can be treated as a problem in stochastic optimal control. Forsyth and Vetzal (2002) developed a penalty method for pricing American options [11]. The FST penalty method combines the penalty method of Forsyth and Vetzal with Fourier space timestepping. The penalty algorithm applies the American constraint implicitly through penalty iterations. It has several advantages over traditional numerical PDE methods of pricing American options that apply the American constraint explicitly. It achieves quadratic convergence in space and time. It can also be easily extended to multi-asset options. The American pricing problem is written as a PIDE by appending a penalty term. In order to apply the Fourier space timestepping method to the American option pricing problem, one must carry out the following steps:

1. Rewrite the pricing PIDE as a penalty equation.
2. Log-transform the penalty PIDE.
3. Apply Fourier Space Timestepping.

At each timestep:

- Apply a DFT and IDFT to the PIDE to obtain a nonlinear system of equations.
- Approximate the integration of equations in time using a first order method.
- Run penalty iterations to solve the discrete equations using DFT/IDFT at each iteration.

3.8.1 Penalty equation

The American option pricing problem is commonly expressed as a linear complementarity problem (LCP) as follows:

$$\begin{aligned} V_t + \mathcal{L}[V] &\leq 0 \\ V^* - V &\leq 0 \\ (V^* - V)(V_t + \mathcal{L}[V]) &= 0 \end{aligned}$$

where

$$\mathcal{L}[V] = \frac{1}{2}\sigma^2 S^2 V_{SS} + (r - \lambda\kappa)SV_S - (r + \lambda)V + \lambda \int_0^\infty V(S\eta)g(\eta)d\eta. \tag{3.18}$$

This pricing problem can be expressed more concisely using the penalty equation

$$V_t + \mathcal{L}[V] + \theta P(V) = 0 \tag{3.19}$$

where $P(V) = \max\{V^* - V, 0\}$ is the penalty function, θ is a penalty coefficient, all other variables are as before, and the terminal condition is given by the payoff function

$$V(T, S) = \phi(S). \tag{3.20}$$

3.8.2 Obtaining constant coefficients

A logarithmic transform can be applied to convert the penalty PIDE. Since S follows a geometric Brownian motion, then $x = \log S$ follows a standard Brownian motion. As before, partial derivatives with respect to S need to be determined in terms of partial derivatives with respect to x in order to implement a change of variable. The first and second partial derivatives V_S and V_{SS} can be determined as in equations 2.15 and 2.16, so that a change of variable from S to x and η to y with some abuse of notation yields the constant coefficient PIDE

$$V_t + \frac{1}{2}\sigma^2(V_{xx} - V_x) + r(V_x - V) + \lambda \int_{-\infty}^{\infty} V(x+y)f(y)dy + \theta P(V) = 0 \tag{3.21}$$

with terminal condition

$$V(T,x) = \phi(x).$$

3.8.3 Discretizing the x variable

In order to apply a discrete Fourier transform to the penalty PIDE we must discretize it in terms of the x variable.

If we let

$$\mathcal{L}(V(t,x)) = \frac{1}{2}\sigma^2(V_{xx} - V_x) + r(V_x - V) + \lambda \int_{-\infty}^{\infty} V(x+y)f(y)dy \tag{3.22}$$

then we can discretize the PIDE in the spatial variable x by applying the transformation $\mathcal{L} \to \mathcal{L}^h(x_0, \ldots x_{N-1})$ as in the simpler European option case.

In the American option case we can write the penalty PIDE as

$$V_t + \mathcal{L}(V(t,x)) + \theta P(V(t,x)) = 0. \tag{3.23}$$

We set

$$\mathcal{L}^h(V_j) = \frac{1}{2}\sigma^2(D^h_{xx}V_j - D^h_x V_j) + r(D^h_x V_j - V_j) + \lambda I_j \tag{3.24}$$

where

$$I_j = I(j\Delta x) = \sum_{k=-N/2+1}^{k=N/2} V_{j+k} f_k \Delta y + O((\Delta y)^2) \tag{3.25}$$

as before. The discretization of the correlation integral follows the method used by d'Halluin et al. [8]. It is assumed that $x_j = j\Delta x$ and $\Delta y = \Delta x$.

The discretized penalty PIDE is then

$$V_t(t,x_j) + \mathcal{L}^h(V(t,x_j)) + \theta P(V(t,x_j)) = 0 \tag{3.26}$$

3.8.4 Discrete Fourier Transform of penalty PIDE

A DFT can be applied to equation 3.26 in order to facilitate the discretization of the spatial derivatives with respect to x.

Taking the DFT of the constant coefficient PIDE yields

$$[\mathcal{F}(V)]_t + \left[i\omega_n\left(r - \frac{1}{2}\sigma^2\right) - \frac{1}{2}\omega_n^2\sigma^2 + \lambda(e^{i\mu\omega_n - \frac{1}{2}\sigma^2\omega_n^2} - 1) - r\right]\mathcal{F}(V) + \theta\mathcal{F}(P(V)) = 0 \tag{3.27}$$

for $j = 0, \ldots, N-1$ and $n = 0, \ldots, N/2 - 1$.

Note that there is no simple expression for $\mathcal{F}(P(V))$ due to the nonlinearity introduced through the max operator.

3.8.5 Penalty ODE

Applying a DFT to the constant coefficient PIDE in x_j produces the following system of discrete ordinary differential equations (ODE) parameterized by ω_n:

$$[\mathcal{F}(V)]_t(t, \omega_n) + \Psi(\omega_n)\mathcal{F}[V](t, \omega_n) + \theta\mathcal{F}(P(V(t, x_j))) = 0 \tag{3.28}$$

with terminal condition

$$\mathcal{F}[V](T, \omega_n) = \mathcal{F}[\phi](\omega_n)$$

where

$$\Psi(\omega_n) = i\omega_n\left(r - \frac{1}{2}\sigma^2\right) - \frac{1}{2}\omega_n^2\sigma^2 + \lambda\left(e^{i\mu\omega_n - \frac{1}{2}\sigma^2\omega_n^2} - 1\right) - r \tag{3.29}$$

for $n = 0, \ldots, N/2 - 1$.

3.8.6 Penalty ODE Solution

We can re-arrange the penalty ODE and write it as

$$\hat{V}_t(t, \omega_n) + \Psi(\omega_n)\hat{V}(t, \omega_n) = -\theta\mathcal{F}[P(V(t, x_j))] \tag{3.30}$$

where $\hat{V} = \mathcal{F}[V]$ as before. Multiplying both sides of the equation by the integrating factor $e^{\Psi(\omega_n)t}$ yields

$$[\hat{V}(t, \omega_n)e^{\Psi(\omega_n)t}]_t = -\theta e^{\Psi(\omega_n)t}\mathcal{F}[P(V(t, x_j))]. \tag{3.31}$$

Let $\hat{V}^m \equiv \hat{V}(t+\Delta t)$ and $\hat{V}^{m-1} \equiv \hat{V}(t)$. Let $t = 0$ for the sake of simplicity. Integrating equation 3.31 backwards in time from $t = \Delta t$ to $t = 0$ yields:

$$\begin{aligned} \hat{V}^{m-1} - \hat{V}^m e^{\Psi(\omega_n)\Delta t} =& -\theta \int_{\Delta t}^{0} e^{\Psi(\omega_n)t} \mathcal{F}[P(V(t, x_j))] dt \\ \approx& -\theta \mathcal{F}[P(V^{m-1})] \int_{\Delta t}^{0} e^{\Psi(\omega_n)t} dt \\ =& \theta \mathcal{F}[P(V^{m-1})] \left(\frac{e^{\Psi(\omega_n)\Delta t} - 1}{\Psi(\omega_n)} \right). \end{aligned} \tag{3.32}$$

We obtain the equation:

$$\hat{V}^{m-1} = \hat{V}^m e^{\Psi(\omega_n)\Delta t} + \theta \mathcal{F}[P(V^{m-1})] \left(\frac{e^{\Psi(\omega_n)\Delta t} - 1}{\Psi(\omega_n)} \right). \tag{3.33}$$

Applying the IDFT yields the following:

$$V^{m-1} = \mathcal{F}^{-1}[\hat{V}^m e^{\Psi(\omega_n)\Delta t}] + \theta \left[\mathcal{F}[P(V^{m-1})] \left(\frac{e^{\Psi(\omega_n)\Delta t} - 1}{\Psi(\omega_n)} \right) \right]. \tag{3.34}$$

Since the ODE in equation 3.33 is non-linear in V, it is difficult to solve through direct analytical methods. Instead it can be solved using an iteration scheme in which the fixed point of the scheme is the solution to the ODE. In the following iteration scheme, $(V^m)^{(k)}$ is the kth iterate of V^m. The iterative equation can be expressed as:

$$(V^{m-1})^{(k)} = (V^{m-1})^{(0)} + \theta \mathcal{F}^{-1} \left[\mathcal{F}[P((V^{m-1})^{(k-1)})] \left(\frac{e^{\Psi(\omega_n)\Delta t} - 1}{\Psi(\omega_n)} \right) \right]. \tag{3.35}$$

The ODE in discrete Fourier space can be solved for $V^{(k)}$ by taking $V^{(k-1)}$ computed in the prior iteration. The Fourier transform of the penalty term from the prior iteration is the source for computation at each iteration. The initial value $V^{(0)}$ to seed the iteration scheme is found using a timestep of the standard FST method:

$$(V^{m-1})^{(0)} = \mathcal{F}^{-1}[\hat{V}^m e^{\Psi(\omega_n)\Delta t}] = \mathcal{F}^{-1}[\mathcal{F}[V^m] e^{\Psi(\omega_n)\Delta t}]. \tag{3.36}$$

3.9 Pricing at $S = 0$

One important case in which it is useful to be able to compute the option price is when the price of the underlying asset reaches zero. This can happen, for example, when the underlying asset is the stock of a company filing for bankruptcy. While the Black Scholes

Merton model does not allow for this possibility, the Merton jump diffusion model may under certain circumstances produce a jump to zero. A jump to default process can also result in a discontinuous jump of the asset price to zero [10]. We will assume that the stock pays no dividends. The payoff of a call option with strike price $K > 0$ written on the stock can be calculated as $\max\{S-K, 0\}$ When the price of a stock reaches zero, such a call option, whether European or American, becomes worthless and its price should also be zero. On the other hand, the payoff of a put option with strike price $K > 0$ written on the stock can be calculated as $\max\{S - K, 0\}$.

Since the FST method uses the log-transform $x = \log(S/S(0))$ to create the real spatial grid and $\log 0$ is undefined, then we must have $S_{min} > 0$ and the asset price $S = 0$ is not represented on the pricing grid. As a result, one drawback of the FST algorithm is that it cannot directly price an option at $S = 0$. Approximations to the price at $S = 0$ can be obtained by expanding the grid and pricing at arbitrarily small values of S_{min}.

Chapter 4

Numerical Experiments

A series of numerical experiments were conducted to test the efficacy of spectral methods in pricing financial options. The first set of experiments was designed to test the convergence of the FST algorithms for American and European options and compare the results with those published by Jackson et al. in [14]. The second set of experiments was designed to test the effects of aliasing on option prices when using the FST algorithm.

4.1 Convergence of the FST Algorithm

The FST algorithm was implemented and run for European and American put options. Convergence studies were conducted and the results are presented in tables below. The FST algorithm achieved quadratic convergence for European put options under the Black Scholes Merton model. Results are summarized in Table 4.1. For American put options, the FST algorithm achieved nearly quadratic convergence as seen from the log(ratio) of close to 2. Results are summarized in Table 4.2. Quadratic convergence was also obtained when applying the FST algorithm to European put option pricing under the Merton jump diffusion model, and the results of these experiments are summarized in Table 4.3.

Table 4.1: Convergence results for pricing of a European put ($S = 100.0$, $K = 100.0$, $T = 0.25$) under Black Scholes Merton model ($r = 0.10$, $\sigma = 0.2$, $q = 0$)

N	Value	Change	$\log_2$(ratio)
2,048	2.827214		
4,096	2.826573	0.006414	
8,192	2.826413	0.001600	2.003136
16,384	2.826373	0.000399	2.000989
32,768	2.826363	0.000099	2.000350

Table 4.2: Convergence results for pricing of an American put ($S = 100.0$, $K = 100.0$, $T = 0.25$) under Black Scholes Merton model ($r = 0.10$, $\sigma = 0.2$, $q = 0$)

N	Value	Change	$\log_2$(ratio)
2,048	3.070062		
4,096	3.069671	0.003909	
8,192	3.069576	0.000951	2.040124
16,384	3.069552	0.000240	1.988379
32,768	3.069546	0.000065	1.892049

Table 4.3: Convergence results for pricing of a European put (S = 100.0, K = 100.0, T = 10) under Merton jump diffusion model (σ = 0.15, λ = 0.1, $\hat{\mu}$ = -1.08, $\hat{\sigma}$ = 0.4, $r = 0.05$, $q = 0.02$).

N	Value	Change	$\log_2$(ratio)
2,048	18.00339796		
4,096	18.00357084	0.0001729	
8,192	18.00357084	0.0000432	2.0008
16,384	18.00357084	0.0000108	2.0004
32,768	18.00357084	0.0000027	2.0002

4.2 Wraparound error

Representing an infinite and continuous asset price domain using a finite and discrete grid leads to truncation error. Moreover, the Fourier transform assumes a periodic domain, whereas the real domain of asset prices is aperiodic. Applying a Fourier transform method such as FST to asset prices therefore causes values at the far right end of the domain, in other words, the asset prices S such that $S >> S_{max}$, to wrap around and produce spurious option prices at the extreme left hand side of the grid. Likewise, the Fourier transform-based method causes asset prices at the extreme left end of the domain, in other words, S such that $S << S_{min}$, to wrap around and produce spurious option prices at the far right hand side of the grid. This phenomenon results in wraparound error and yields incorrect option prices at the extreme left and right sides of the asset price grid.

4.3 Studying wraparound error

To study the effects of wraparound error on option pricing, the FST algorithm was run for European and American options under the Black Scholes Merton model and European options under the Merton Jump diffusion model. The results were compared with other standard methods of option pricing such as closed form solutions and finite difference methods. The FST algorithm was run with $x_{max} = 7.5$ and $x_{min} = -x_{max} = -7.5$ in all cases in this section.

For European options under the Black Scholes Merton model, the method used as a basis for comparison was the closed-form solution obtained using the blsprice() function in Matlab. For American options under the Black Scholes Merton model, the comparative solution was generated using an implementation of the penalty method of Forsyth and Vetzal. For European options under the Merton jump diffusion model, the comparative algorithm was a closed-form solution implemented in Matlab.

4.3.1 European options under Black Scholes Merton model

Significant wraparound error was observed for European put options under the Black Scholes Merton model. Wraparound error was prominent at the left boundary of the domain near $S = 0$. Prices obtained using Fourier space timestepping matched those generated using the blsprice() function for values of S above 0.5. Results obtained from running numerical experiments on European put options under the Black Scholes Merton model are summarized in Table 4.4.

Significant wraparound error was also observed for European call options under the Black Scholes Merton model. Wraparound error was prominent at the left boundary of the domain near $S = 0$. Prices obtained using Fourier space timestepping matched those generated using the blsprice() function for values of S above approximately 0.2. Results obtained from running numerical experiments on European call options under the Black Scholes Merton model are summarized in table 4.5.

Table 4.4: Wraparound error effects for pricing of a European put ($S = 100.0$, $K = 100.0$, $T = 0.25$, $\Delta t = 0.25$, number of nodes $N = 32,768$) under Black Scholes Merton model ($r = 0.10$, $\sigma = 0.2$, $q = 0$)

S (asset price)	Option price using FST	Option price using blsprice
0.01	-47.5052	99.9161
0.10	97.4310	99.1806
1	96.5310	96.5310
10	87.5310	87.5310
100	2.8264	2.8264
1000	0.0000	0.0000

Table 4.5: Wraparound error effects for pricing of a European call (S = 100.0, K = 100.0, T = 0.25, Δt = 0.25, number of nodes $N = 32,768$) under Black Scholes Merton model (r = 0.10, σ = 0.2, q = 0)

S (asset price)	Option price using FST	Option price using blsprice
0.01	474,405.6755	0.0000
0.10	0.0001	0.0000
1	0.0000	0.0000
10	0.0000	0.0000
100	5.2954	5.2954
1000	902.4690	902.4690

4.3.2 American options under Black Scholes Merton model

No wraparound error was observed for American put options under the Black Scholes Merton model. Prices generated using Fourier space timestepping closely matched those produced by a separately implemented penalty algorithm. The absence of wraparound error was most likely due to the application of the American constraint through penalty iterations at every timestep, which required transforming between Fourier space and real space. Application of the American constraint prevented the accumulation of error. Results obtained from running numerical experiments on American put options under the Black Scholes Merton model are summarized in Table 4.6.

Table 4.6: Wraparound error effects for pricing of an American put ($S = 100.0$, $K = 100.0$, $T = 0.25$, $\Delta t = 1/128$, number of nodes $N = 32,768$) under Black Scholes Merton model ($r = 0.10$, $\sigma = 0.2$, $q = 0$)

S (asset price)	Option price using FST	Option price using penalty method
0.01	99.9900	99.9900
0.10	99.9000	99.9000
1	99.0000	99.0000
10	90.0000	90.0000
100	3.0699	3.0699
1000	0.0000	0.0000

Significant wraparound error was observed for American call options under the Black Scholes Merton model. Wraparound error was prominent at the left boundary of the domain near $S = 0$. Prices obtained using Fourier space timestepping matched those generated using a separately implemented penalty method for values of S above 0.2. Results obtained from running numerical experiments on American call options under the Black Scholes Merton model are summarized in Table 4.7.

Table 4.7: Wraparound error effects for pricing of an American call ($S = 100.0$, $K = 100.0$, $T = 0.25$, $\Delta t = 1/128$, number of nodes $N = 32,768$) under Black Scholes Merton model ($r = 0.10$, $\sigma = 0.2$, $q = 0$)

S (asset price)	Option price using FST	Option price using penalty method
0.01	-2,180,209,174.8817	0.0000
0.10	0.00014	0.0000
1	0.0000	0.0000
10	0.0000	0.0000
100	5.2954	5.2953
1000	902.4690	902.4690

4.3.3 European options under Merton jump diffusion model

Wraparound error was observed for European put options under the Merton jump diffusion model. Wraparound error was prominent at the left boundary of the domain near $S = 0$. Prices obtained using Fourier space timestepping approached those generated using the closed-form solution for values of S above 5. However, for values of S as high as 50 the prices were still differ from the closed-form solution by about a cent. Results obtained from running numerical experiments on European put options under the Merton jump diffusion model are summarized in Table 4.8.

Table 4.8: Wraparound error effects for pricing of a European put ($S = 100.0$, $K = 100.0$, $T = 10$, $\Delta t = 10$, number of nodes $N = 32,768$) under the Merton jump diffusion model ($\sigma = 0.15$, $\lambda = 0.1$, $\hat{\mu} =$ -1.08, $\hat{\sigma} = 0.4$, $r = 0.05$, $q = 0.02$)

S (asset price)	Option price using FST	Option price using closed-form
0.01	16.3928	60.6431
0.10	42.8138	60.5531
1	57.9421	59.6531
10	50.5842	50.6661
100	15.4379	15.4403
1000	1.5586	1.5586

Significant wraparound error was also observed for European call options under the Merton jump diffusion model. Wraparound error was prominent at the left boundary of the domain near $S = 0$. Prices obtained using Fourier space timestepping approached those generated using the closed-form solution for values of S above 200, so inaccurate results were obtained for prices both above and below the strike price of 100. However, for values of S as high as 500 the prices were still off by about 0.18. Results obtained from running numerical experiments on European put options under the Merton jump diffusion model are summarized in Table 4.9.

Table 4.9: Wraparound error effects for pricing of a European call ($S = 100.0$, $K = 100.0$, $T = 10$, $\Delta t = 10$, number of nodes $N = 32,768$) under the Merton jump diffusion model ($\sigma = 0.15$, $\lambda = 0.1$, $\hat{\mu}$ = -1.08, $\hat{\sigma} = 0.4$, $r = 0.05$, $q = 0.02$)

S (asset price)	Option price using FST	Option price using closed-form
0.01	31,425.6333	2.5841×10^{-67}
0.10	15,288.1945	9.6830×10^{-36}
1	1,711.5541	8.6619×10^{-14}
10	88.1405	1.3024×10^{-2}
100	57.4817	54.7873
1000	940.9614	940.9055

4.4 Spatial grid expansion

One potential solution to the inaccuracies in option prices near the boundaries is to expand the real space boundaries. Expansion of boundaries generally implies a trade off between size of the pricing domain and accuracy of the option value generated near the strike price of interest. While such an expansion of boundaries allows us to capture the behaviour of the option value function over a greater domain, the accuracy of the option price in the range of interest may be reduced. In the results of the previous section, the real space boundaries were set at $x_{max} = 7.5$ and $x_{min} = -x_{max} = -7.5$. If we instead augment the boundaries to $x_{max} = 15.5$ and $x_{min} = -x_{max} = -15.5$ then we obtain the following results.

4.4.1 European options under Black Scholes Merton model with $x_{max} = 15.5$

Substantially lower wraparound error was observed for European put options priced under the Black Scholes Merton model using an expanded spatial grid as compared with those priced on the original grid. Whereas prices diverged widely from the closed-form solution in pricing with the original spatial grid, on the expanded grid no price differed from the closed-form solution by more than 2.50. The closed-form solution was implemented using the function blsprice(). All option prices generated for various asset price values were positive and relatively close to true prices, but the level of accuracy obtained was not satisfactory. Wraparound error for European put options likely persists due to the shape of the put payoff function, which at $S = 0$ has an intercept equal to the strike price. Moreover, unlike an American put option, the European put does not allow early exercise and hence does not benefit from the correcting effect of the max function at each timestep. Results obtained from running numerical experiments on European put options under the Black Scholes Merton model on an expanded spatial grid are summarized in Table 4.10.

Table 4.10: Wraparound error effects for pricing of a European put (S = 100.0, K = 100.0, T = 0.25, Δt = 0.25, number of nodes $N = 32,768$) under Black Scholes Merton model (r = 0.10, σ = 0.2, q = 0) with x_{max} = 15.5

S (asset price)	Option price using FST	Option price using blsprice
0.01	97.5210	99.9161
0.10	97.4310	99.1806
1	96.5310	96.5310
10	87.5310	87.5310
100	2.8264	2.8264
1000	0.0000	0.0000

No wraparound error was observed for European call options when pricing under the Black Scholes Merton model using an expanded spatial grid. Prices thus obtained very closely matched those generated using the closed-form solution implemented using blsprice(). All option prices generated for various asset price values were positive and the level of accuracy attained was satisfactory. Results obtained from running numerical experiments on European call options under the Black Scholes Merton model on an expanded spatial grid are summarized in Table 4.11.

Table 4.11: Wraparound error effects for pricing of a European call ($S = 100.0$, $K = 100.0$, $T = 0.25$, $\Delta t = 0.25$, number of nodes $N = 32,768$) under Black Scholes Merton model ($r = 0.10$, $\sigma = 0.2$, $q = 0$) with $x_{max} = 15.5$

S (asset price)	Option price using FST	Option price using blsprice
0.01	0.0000	0.0000
0.10	0.0000	0.0000
1	0.0000	0.0000
10	0.0000	0.0000
100	5.2954	5.2954
1000	902.4690	902.4690

4.4.2 American options under Black Scholes Merton model with $x_{max} = 15.5$

There was no wraparound error observed for American put options on the original grid, so they were not priced using an expanded grid since further reduction in error was not required.

No lower wraparound error was observed for American call options under the Black Scholes Merton model when using an expanded spatial grid. Prices obtained using Fourier space timestepping approximately matched those generated using a separately implemented penalty method. Results obtained from running numerical experiments on American call options under the Black Scholes Merton model on an expanded spatial grid are summarized in Table 4.12.

Table 4.12: Wraparound error effects for pricing of an American call ($S = 100.0$, $K = 100.0$, $T = 0.25$, $\Delta t = 1/128$, number of nodes $N = 32,768$) under Black Scholes Merton model ($r = 0.10$, $\sigma = 0.2$, $q = 0$) with $x_{max} = 15.5$

S (asset price)	Option price using FST	Option price using penalty method
0.01	0.0000	0.0000
0.10	0.0000	0.0000
1	0.0000	0.0000
10	0.0000	0.0000
100	5.2954	5.2954
1000	902.4690	902.4690

4.4.3 European options under Merton jump diffusion model with $x_{max} = 15.5$

Substantially lower wraparound error was observed for European put options under the Merton jump diffusion model using an expanded spatial grid. The expanded grid covers a larger domain of asset prices and hence allows more accurate pricing of options, particularly near the boundaries. Prices obtained using Fourier space timestepping differed from those generated using the closed-form solution for values of S below 0.10. For $S = 0.01$ the difference was nearly 0.02. Results obtained from running numerical experiments on European put options under the Merton jump diffusion model on an expanded spatial grid are summarized in Table 4.13.

Table 4.13: Wraparound error effects for pricing of a European put ($S = 100.0$, $K = 100.0$, $T = 10$, $\Delta t = 10$, number of nodes $N = 32,768$) under the Merton jump diffusion model ($\sigma = 0.15$, $\lambda = 0.1$, $\hat{\mu} =$ -1.08, $\hat{\sigma} = 0.4$, $r = 0.05$, $q = 0$) with $x_{max} = 15.5$

S (asset price)	Option price using FST	Option price using closed-form
0.01	60.6269	60.6431
0.10	60.5527	60.5531
1	59.6531	59.6531
10	50.6661	50.6661
100	15.4404	15.4404
1000	1.5586	1.5586

Substantially lower wraparound error was observed for European call options priced under the Merton jump diffusion model using an expanded spatial grid as compared with those priced on the original grid. Whereas prices diverged widely from the closed-form solution in pricing with the original spatial grid, the expanded grid ensured that no price differed from the closed-form solution by more than 2.50. All option prices generated for various asset price values were positive and relatively close to true prices, but large errors remained and the level of accuracy obtained was not entirely satisfactory. Results obtained from running numerical experiments on European call options under the Merton model on an expanded spatial grid are summarized in Tables 4.14 and 4.14.

Table 4.14: Wraparound error effects for pricing of a European call ($S = 100.0$, $K = 100.0$, $T = 10$, $\Delta t = 10$, number of nodes $N = 32,768$) under the Merton jump diffusion model ($\sigma = 0.15$, $\lambda = 0.1$, $\hat{\mu} =$ -1.08, $\hat{\sigma} = 0.4$, $r = 0.05$, $q = 0$)with $x_{max} = 31.5$

S (asset price)	Option price using FST	Option price using closed-form
0.01	7.3568×10^{-2}	2.5841×10^{-67}
0.10	5.4435×10^{-2}	9.6830×10^{-36}
1	3.4758×10^{-2}	8.6619×10^{-14}
10	1.7560×10^{-3}	1.3024×10^{-2}
100	54.8156	54.7873
1000	940.9389	940.9055

4.5 Augmenting grid with zeros

Another method of potentially reducing wraparound error is to add zeros to the left and right sides of the real spatial domain. If $N/2$ zeros are added to the left of the domain and $N/2$ zeros to the right then the size of the real spatial domain is doubled. Unlike the previous section, where the size of the spatial domain was doubled and all points in the real spatial domain were used to calculate the option price, in this case the solution points corresponding to the zeros added to the left and right of the domain are discarded so that the solution is finally interpolated from a range of width N instead of $2N$. Padding with zeros can be applied after first attempting to correct wraparound error by expanding the real spatial domain.

4.5.1 European options under Black Scholes Merton model with $x_{max} = 15.5$ and added zeros

European put options were priced under the Black Scholes Merton model using an expanded spatial grid augmented with zeros. There was no noticeable reduction in wraparound error from the method using an expanded spatial grid. The closed-form solution was implemented using the function blsprice(). All option prices generated for various asset price values were positive and relatively close to true prices, but the level of accuracy obtained was not completely satisfactory. Wraparound error for European put options is once again most likely due to the shape of the put payoff function, which at $S = 0$ has an intercept equal to the strike price, with no constraint enforcement at each timestep unlike American put options. Results obtained from running numerical experiments on European put options under the Black Scholes Merton model on an expanded spatial grid are summarized in Table 4.15.

Since there was no wraparound error for European call options on the expanded spatial grid, they were not priced using added zeros since further reduction in error was not required.

4.5.2 American options under Black Scholes Merton model with $x_{max} = 15.5$ and added zeros

Significant wraparound error was not observed for American put and call options on the original grid, so they were not priced using added zeros since further reduction in error was not required.

Table 4.15: Wraparound error effects for pricing of a European put (S = 100.0, K = 100.0, T = 0.25, Δt = 0.25, number of nodes $N = 32,768$) under Black Scholes Merton model (r = 0.10, σ = 0.2, q = 0) with x_{max} = 15.5 and added zeros

S (asset price)	Option price using FST	Option price using blsprice
0.01	97.5210	99.9161
0.10	97.4310	99.1806
1	96.5310	96.5310
10	87.5310	87.5310
100	2.8264	2.8264
1000	0.0000	0.0000

4.5.3 European options under Merton jump diffusion model with $x_{max} = 15.5$ and added zeros

Wraparound error observed for European put options under the Merton jump diffusion model using an expanded spatial grid was not significantly reduced by augmenting with zeros. Prices obtained using Fourier space timestepping closely matched those generated using the closed-form solution for all values of S tested, but the results were no better than those obtained using only an expanded spatial grid. Results obtained from running numerical experiments on European put options under the Merton jump diffusion model on an expanded spatial grid are summarized in Table 4.16.

Table 4.16: Wraparound error effects for pricing of a European put ($S = 100.0$, $K = 100.0$, $T = 10$, $\Delta t = 10$, number of nodes $N = 32,768$) under the Merton jump diffusion model ($\sigma = 0.15$, $\lambda = 0.1$, $\hat{\mu} =$ -1.08, $\hat{\sigma} = 0.4$, $r = 0.05$, $q = 0$) with $x_{max} = 15.5$

S (asset price)	Option price using FST	Option price using closed-form
0.01	60.6269	60.6431
0.10	60.5527	60.5531
1	59.6531	59.6531
10	50.6661	50.6661
100	15.4404	15.4404
1000	1.5586	1.5586

Substantially lower wraparound error was observed for European call options priced under the Merton jump diffusion model using an expanded spatial grid with added zeros as compared with those priced on the original grid. All option prices generated for various asset price values were positive and relatively close to true prices, but large errors remained and the level of accuracy obtained was not entirely satisfactory. Results obtained from running numerical experiments on European call options under the Merton model on an expanded spatial grid are summarized in Table 4.17.

Table 4.17: Wraparound error effects for pricing of a European call ($S = 100.0$, $K = 100.0$, $T = 10$, $\Delta t = 10$, number of nodes $N = 32,768$) under the Merton jump diffusion model ($\sigma = 0.15$, $\lambda = 0.1$, $\hat{\mu} =$ -1.08, $\hat{\sigma} = 0.4$, $r = 0.05$, $q = 0$)with $x_{max} = 31.5$

S (asset price)	Option price using FST	Option price using closed-form
0.01	0.0000	2.5841×10^{-67}
0.10	0.0000	9.6830×10^{-36}
1	0.0000	8.6619×10^{-14}
10	1.3024×10^{-2}	1.3024×10^{-2}
100	54.7873	54.7873
1000	940.9055	940.9055

Chapter 5

Conclusions

American and European options priced using the Fourier space timestepping algorithm often demonstrate a degree of wraparound error. American put options are immune to such error due to the application of the American constraint at each timestep.

Upon running numerical experiments with American and European put and call options with an underlying geometric Brownian motion it was found that in several cases the wraparound error was considerably reduced by increasing the size of the real spatial grid. Such an expansion of the grid increases the size of the domain over which accurate pricing is possible. Any inaccuracies are pushed further out to the boundaries of the domain. Reduction in error through grid expansion was obtained in all cases except for American put options, which had no wraparound error initially. Wraparound error for European and American call options under the Black Scholes Merton model was virtually eliminated. There is still noticeable wraparound error in the case of European put options.

In the case of European call options priced under the Merton jump diffusion model, the wraparound error was further reduced when the expanded spatial grid was augmented by adding zeros near the left and right boundaries and solution points corresponding to zeros were discarded. Reduction in error was not obtained for European put options, most likely due to the shape and positive option price intercept of the put payoff function. The zero padding method was not applied to European and American call options, since error was successfully eliminated using spatial grid expansion alone.

Simple methods such as grid expansion and zero padding did not eliminate wraparound error for the Merton jump diffusion model. This is likely due to the presence of a non-local jump term in the Merton jump diffusion PIDE. Careful further research is necessary in order to determine precisely why the wraparound error for options priced under the Merton jump diffusion model could not be eliminated and to investigate other methods of reducing wraparound error in this case.

Bibliography

[1] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973. 5

[2] R. Carmona and V. Durrleman. Pricing and hedging spread options. *Siam Review*, 45(4):627–685, 2003.

[3] P. Carr and D. Madan. Option valuation using the fast fourier transform. *Journal of Computational Finance*, 2(4):61–73, 1999.

[4] R. Cont and P. Tankov. Financial modelling with jump processes. *Chapman and Hall/CRC*, 2004.

[5] J. Cox and A. Ross. The valuation of options for alternative stochastic processes. *Journal of Financial Economics*, 3:145–166, 1976. 2

[6] M. A. Dempster and S. Hong. Spread option valuation and the fast fourier transform. *Technical Report 26, Cambridge University*, 2000.

[7] E. Derman and I. Kani. Riding on a smile. *Risk*, 7:32–39, 1994. 2

[8] Y. d'Halluin, P. A. Forsyth, and G. Labahn. A penalty method for american options with jump diffusion processes. *Numerische Mathematik*, 97(2):321–352, 2003. 7, 15, 20

[9] B. Duprie. Pricing with a smile. *Risk*, 7:18–20, 1994. 2

[10] P. A. Forsyth. Jump processes and regime switching. *Unpublished working paper*, 2005. 23

[11] P. A. Forsyth and K. R. Vetzal. Quadratic convergence of a penalty method for valuing american options. *SIAM Journal on Scientific Computing*, 23(2):2096–2123, 2002. 18

[12] S. L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, 6(2):327–343, 1993. 2, 6

[13] J. Ingersoll. Valuing foreign exchange rate derivatives with a bounded exchange process. *Review of Derivatives Research*, 1:159–81, 1997. 2

[14] Kenneth R. Jackson, Sebastian Jaimungal, and Vladimir Surkov. Fourier space time-stepping for option pricing with Lévy models. *Journal of Computational Finance*, 12(2):1–28, 2008. 2, 3, 4, 13, 16, 17, 24

[15] S. G. Kou. A jump-diffusion model for option pricing. *Management Science*, 48(8):1086–1101, 2002. 6

[16] R. Lord, F. Fang, F. Bervoets, and K. Oosterlee. A fast and accurate fft-based method for pricing early-exercise options under Lévy processes. *SIAM Journal on Scientific Computing*, 30(4):1678–1705, 2008. 3

[17] D. Madan and E. Seneta. The vg model for share market returns. *Journal of Business*, 63(4):511–524, 1990. 2

[18] R. C. Merton. Theory of rational option pricing. *The Bell Journal of Economics and Management Science*, 4(1):141–183, 1973.

[19] R. C. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Computational Finance*, 3:125–144, 1976. 2, 6

[20] V. Naik. Option valuation and hedging strategies with jumps in the volatility of assets returns. *Journal of Finance*, 48:1969–1984, 1993. 2

[21] S. Rady. Option pricing in the presence of natural boundaries and a quadratic diffusion term. *Finance and Stochastics*, 1:331–344, 1997. 2

[22] S. Raible. Levy processes in finance: theory, numerics, and empirical facts. *PhD Thesis, University of Freiburg*, 2000.

[23] L. Trefethen. Spectral methods in matlab. *Society for Industrial and Applied Mathematics*, pages 17–24, 2000. 14

[24] I. R. Wang, J. W. L. Wan, and P. A. Forsyth. Robust numerical valuation of european and american options under the cgmy process. *Journal of Computational Finance*, 10:31–69, 2007. 3