Options Pricing under Shared-jump Diffusion Model by Fourier Space Time-stepping Method

by

Bangyao Xu

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

Supervisors: Prof. Peter Forsyth & Prof. George Labahn

Waterloo, Ontario, Canada, 2018

© Bangyao Xu 2018
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

In this essay, a shared-jump diffusion model is introduced to describe financial shocks and contagion effects. Under the shared-jump diffusion model for underlying assets, the derivation of the Fourier Space Time-stepping (FST) method is demonstrated to solve the corresponding partial integro-differential equations (PIDE). Numerical results of pricing single- and multi-asset European options under three kinds of jump diffusion models are presented. In addition, constant padding is proposed to reduce the wrap-around error in one- and two-dimensional cases. The data from global financial markets in recent decades are used to conduct the empirical analysis.
Acknowledgements

I would like to thank my supervisors, Professor Peter Forsyth and Professor George Labahn, for their guidance, support and encouragement throughout this year.

Also, I want to express my sincere thanks to Professor Yuying Li, for reading this essay and providing valuable suggestions.

To my parents, for their love and inspiration throughout my life.

To my dear friends, for always sharing ideas and making progress with me: Ershi, Xukun, Wenqing, Kasun, Boon, Jessie, Fan, Linqi, Edward and Emily.

Last but not least, I want to thank our director, Dr. Jeff Orchard and coordinator, Amanda Guderian, for their information and suggestion on daily life.
Dedication

This is dedicated to Cheryl, the one I love.
Table of Contents

List of Tables ix

List of Figures xi

1 Introduction 1

2 Mathematical Models 4

2.1 Introduction ................................................. 4
2.2 One-factor Jump Diffusion Model ................................ 5
2.3 Two-factor Jump Diffusion Model ................................ 6
2.4 Shared-jump Diffusion Model .................................. 8

3 Fourier Space Time-stepping Method 14

3.1 Introduction .................................................. 14
3.2 Continuous Fourier Transform ................................ 14
3.3 Discrete Fourier Transform .................................. 15
3.4 FST Method under a Shared-jump Diffusion Model .......... 18
  3.4.1 Fourier Transform ........................................... 18
  3.4.2 Solving the Ordinary Differential Equation ............... 20
  3.4.3 Fourier Space Time-stepping ............................. 21
  3.4.4 Illustration of Method .................................... 22
C Monte Carlo Approach

C.1 Methodology ................................................. 60
C.2 Algorithm for Monte Carlo Simulation .................. 63

References ................................................. 64
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Parameters for a European put under the one-factor Merton jump diffusion</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>model</td>
<td></td>
</tr>
<tr>
<td>4.2</td>
<td>Pricing results of a European put under the one-factor Merton jump diffusion</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>model: $[x_{\text{min}}, x_{\text{max}}] = [-7.5, 7.5]$</td>
<td></td>
</tr>
<tr>
<td>4.3</td>
<td>Parameters for a European call under the one-factor Kou jump diffusion model</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.4</td>
<td>Pricing results of a European call under the one-factor Kou jump diffusion</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>model: $[x_{\text{min}}, x_{\text{max}}] = [-7.5, 7.5]$</td>
<td></td>
</tr>
<tr>
<td>4.5</td>
<td>Parameters for wrap-around analysis: a European put option</td>
<td>28</td>
</tr>
<tr>
<td>4.6</td>
<td>Effect of wrap-around error on the value of a European put option: zero</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>padding and constant padding</td>
<td></td>
</tr>
<tr>
<td>4.7</td>
<td>Parameters for wrap-around analysis: a European call option</td>
<td>31</td>
</tr>
<tr>
<td>4.8</td>
<td>Effect of wrap-around error on the value of a European call option:</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>asymptotic padding</td>
<td></td>
</tr>
<tr>
<td>4.9</td>
<td>Parameters for a European spread call under the two-factor Merton jump</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>diffusion model</td>
<td></td>
</tr>
<tr>
<td>4.10</td>
<td>Pricing results of a European spread call under the two-factor Merton jump</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>diffusion model: $[x_{\text{min}}, x_{\text{max}}]^2 = [-7.5, 7.5]^2$</td>
<td></td>
</tr>
<tr>
<td>4.11</td>
<td>Effect of wrap-around error on the value of a European spread call option:</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>no padding</td>
<td></td>
</tr>
<tr>
<td>4.12</td>
<td>Effect of wrap-around error on the value of a European spread call option:</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>constant padding</td>
<td></td>
</tr>
<tr>
<td>4.13</td>
<td>Parameters for a European spread call under the shared-jump diffusion model</td>
<td>37</td>
</tr>
</tbody>
</table>
4.14 Pricing results of a European spread call under the shared-jump diffusion model: \textit{FST method}, \([x_{\text{min}}, x_{\text{max}}]^2 = [-7.5, 7.5]^2\) ........................................ 38

4.15 Pricing results of a European spread call under the shared-jump diffusion model: \textit{Monte Carlo simulation} .................................................. 39

4.16 Parameters for wrap-around analysis: a European spread put option ................................................................. 40

4.17 Effect of wrap-around error on the value of a European spread put option: \textit{no padding} ................................. 41

4.18 Effect of wrap-around error on the value of a European spread put option: \textit{constant padding} ................................. 41

4.19 Convergence table of a European spread put under the shared-jump diffusion model using constant padding: \textit{constant} \(\Omega\), \textit{various} \(\Delta x\) ........................................ 42

4.20 Convergence table of a European spread put under the shared-jump diffusion model using constant padding: \textit{constant} \(\Delta x\), \textit{various} \(\Omega\) ........................................ 42

5.1 Descriptive statistics for daily log returns of S&P 500, Eurostox and FTSE ................................. 46

5.2 Descriptive statistics for monthly log returns of S&P 500, Eurostox and FTSE ................................................ 47

5.3 Parameters estimated from daily data ........................................ 49

5.4 Parameters estimated from monthly data ........................................ 50

5.5 Specific dates of jumps detected from monthly data ........................................ 50
List of Figures

4.1 European call option pricing using zero padding and constant padding . . . 27
4.2 European put option pricing using zero padding and constant padding . . . 27
4.3 European call option pricing using asymptotic padding . . . . . . . . . . . 30
4.4 Intuitive idea for constant padding in two dimensions . . . . . . . . . . . . 35

5.1 Prices of S&P 500, Eurostox and FTSE . . . . . . . . . . . . . . . . . . . . . 44
5.2 Log returns of S&P 500, Eurostox and FTSE . . . . . . . . . . . . . . . . . 44
5.3 Scaled observed density and standard normal density of S&P 500 log returns. *Left: daily, Right: monthly* . . . . . . . . . . . . . . . . . . . . . . . . . . 45
5.4 Scaled observed density and standard normal density of Eurostox log returns. *Left: daily, Right: monthly* . . . . . . . . . . . . . . . . . . . . . . . . . . 45
5.5 Scaled observed density and standard normal density of FTSE log returns. *Left: daily, Right: monthly* . . . . . . . . . . . . . . . . . . . . . . . . . . 46
Chapter 1

Introduction

In finance, a derivative is a contract between two or more parties whose value is based on a particular underlying asset or a basket of underlying assets. The most common underlying instruments include bonds, commodities, currencies, interest rates, market indices and stocks. Some of the more common derivatives include forwards, futures, options, swaps, and variations such as synthetic collateralized debt obligations (CDOs) and credit default swaps (CDSs).

Derivatives are mainly used for speculating and purpose of risk management. By holding derivatives, speculators take risks to pursue potential quick, large profits. Meanwhile, investors can effectively hedge various risks through the use of derivatives. That is, derivatives offer market participants the chance to customize satisfactory risk profiles for their own businesses.

An option is a contract which gives the buyer the right, but not the obligation, to buy or sell an underlying asset at a specified strike price, on or before expiration. A European option can only be exercised on expiration while an American option can be exercised on any trading day before expiry.

To price derivatives, the assumption of a stochastic process model for the underlying asset is required. The well-known Black-Scholes model was first proposed by Black and Scholes (1973)[3]. In this model, the underlying asset price is assumed to follow a geometric Brownian motion and the no-arbitrage price of options can be determined as the solution of a partial differential equation (PDE). The jump diffusion model was first suggested by Merton (1976)[14]. A Poisson-driven random jump component is added to geometric Brownian motion in order to represent the abnormal changes in underlying price. When
the underlying asset follows a jump diffusion model, the pricing problem reduces to solving a partial integro-differential equation (PIDE).

The Fourier Space Time-stepping (FST) method was first developed by Jackson, Jaimungal and Surkov (2008)[8]. FST method uses the Fourier transform to solve the PDE or PIDE numerically. The continuous Fourier transform (CFT) is a linear operator which maps spatial derivatives into multiplications in Fourier space. Because of the convenient properties of the Fourier transform, the PDE or PIDE in real space can be converted into a linear first-order ordinary differential equation (ODE) in Fourier space, which can be easily solved in closed-form. In practice, the discrete Fourier transform (DFT) is used to approximate the CFT so the continuous domain is discretized for computational purpose. Unfortunately, the FST method causes wrap-around error in the option value solution, which needs to be addressed.

With the development of economic globalization, contagion effects tend to be inevitable as the result of financial shocks. The contagion literature identifies at least three possible mechanisms by which shocks in one market may spill over into other markets. First, Allen and Gale (2000)[1] suggested that contagion occurs through a liquidity shock across all markets. Second, Kiyotaki and Moore (2002)[9] claimed that contagion can be viewed as the transmission of information from more-liquid markets or markets with more rapid price discovery to other markets. Third, Vayanos (2004)[18] concluded that contagion occurs as negative returns in the distressed market affect subsequent returns in other markets via a time-varying risk premium.

Besides the theoretical research on financial contagion, plenty of empirical tests are conducted as well. By constructing a set of dummy variables using daily news to capture the impact of own-country and cross-border news on the markets, Baig and Goldfajn (1999)[2] found that correlations in currency and sovereign spreads between the financial markets of Thailand, Malaysia, Indonesia, Korea, and the Philippines increased significantly during the Asian crisis period. Longstaff (2010)[12] found strong evidence of contagion in the financial markets by investigating the pricing of subprime asset-backed collateralized debt obligations (CDOs) and their contagion effects on other markets.

In this essay, a shared-jump diffusion model is introduced with the motivation of describing financial shocks and contagion effects. Under this model, two underlying assets are assumed to go through abnormal changes at the same time. In other words, two underlying assets share the same jump driven by unique Poisson process. The FST method is implemented to solve the corresponding PIDE under the shared-jump diffusion model. Also, constant padding is developed to remedy the wrap-around error in one- and two-dimensional cases. Last but not least, empirical data analysis is carried out to estimate
the parameters and provide evidence for developing the shared-jump diffusion model.

The remainder of this essay is structured as follows. Chapter 2 introduces one-factor jump diffusion model, two-factor jump diffusion model and the new shared-jump diffusion model. Chapter 3 presents the derivation of the Fourier Space Time-stepping (FST) method under shared-jump diffusion model. Chapter 4 provides numerical examples under various models and the appropriate treatment of wrap-around error. Chapter 5 gives the empirical analysis based on the real data from global financial markets. Chapter 6 lists the conclusions. The Appendix provides the Fourier transform of distributions, the FST algorithms for European options and methodology of Monte Carlo simulation.
Chapter 2

Mathematical Models

2.1 Introduction

The Black-Scholes-Merton (BSM) model for pricing of options, assumes the prices of underlying assets follow geometric Brownian motion. By assuming certain ideal conditions on financial markets and constructing a self-financing replicating portfolio, Black and Scholes (1973)[3] show that the option pricing problem under the BSM model can be reduced to solving a second-order partial differential equation (PDE).

In order to take random jumps into consideration, Merton (1976)[14] suggested a jump diffusion model to describe the dramatic changes in underlying prices within a very short time period. That model adds a Poisson process, which may cause discontinuities in sample paths, to the geometric Brownian motion. The distribution of random jump sizes are commonly chosen to follow a log-normal distribution or a double exponential distribution by Kou (2002)[10]. Under the jump diffusion model, as with the BSM model, the option pricing problem reduces to solving a second-order partial integro-differential equation (PIDE).

In this chapter, a new two-asset jump diffusion model is introduced to capture the extreme cases which happened in global financial markets such as the dot-com bubble\(^1\) in 2002 and the financial crisis\(^2\) in 2008. Under the shared-jump diffusion model, the

---

\(^1\)A historic economic bubble and period of excessive speculation that occurred roughly from 1997 to 2001, a period of extreme growth in the usage and adaptation of the Internet.

\(^2\)A crisis first started in the subprime mortgage market in the United States, and then developed into a full-blown international banking crisis.
random jumps and their jump sizes are driven by the single Poisson process, which is quite reasonable to characterize the financial shocks and the corresponding contagion effects on global markets.

### 2.2 One-factor Jump Diffusion Model

The sample paths of the underlying price $S$ are modelled by a stochastic differential equation:

$$\frac{dS}{S} = \mu dt + \sigma dZ + (\eta - 1)dq,$$  \hspace{1cm} (2.1)

where

- $\mu$ = drift rate,
- $\sigma$ = underlying volatility,
- $dZ$ = increment of standard Brownian motion,
- $\eta - 1$ = impulse function producing a jump from $S$ to $S\eta$,
- $dq = \begin{cases} 0, & \text{with probability } 1 - \lambda dt, \\ 1, & \text{with probability } \lambda dt, \end{cases}$
- $\lambda$ = mean arrival rate of Poisson jump.

Let $V(S, \tau)$ be the option value, with $\tau = T - t$, the time to expiry $T$. By Ito’s Lemma and no-arbitrage arguments, a partial integro-differential equation (PIDE) for $V(S, \tau)$\cite{15, 19} can be written as:

$$V_r = \frac{\sigma^2 S^2}{2} V_{SS} + (r - \lambda \kappa) S V_S - (r + \lambda) V + \left( \lambda \int_0^\infty V(S\eta) g(\eta) d\eta \right)$$  \hspace{1cm} (2.2)

where

- $T$ = expiry time,
- $r$ = risk free rate,
- $\tau = T - t$, where $t$ is current time,
- $\kappa = E[\eta - 1]$, where $E[\eta] = \int_0^\infty \eta g(\eta) d\eta$,
- $g(\eta)$ = probability density function of the jump magnitude.
with the initial condition:

\[ V(S,0) = \max(S - K, 0), \quad \text{for call option} \]  

(2.3) 

or 

\[ V(S,0) = \max(K - S, 0), \quad \text{for put option} \]  

(2.4) 

where \( K \) is the strike price. Here, introduce \( f(x) = g(e^x)e^x \) as the density function of \( \log(\eta) \). So, with Merton jump density (normal distribution), we have 

\[ f(y) = \frac{1}{\sqrt{2\pi}\gamma} e^{-\frac{1}{2}(\frac{y-\mu}{\gamma})^2}, \]

where \( \mu \) and \( \gamma \) are constant parameters. Using the Kou jump density (double exponential distribution), we get 

\[ f(y) = p\eta_1 e^{-y\eta_1} \cdot 1_{\{y \geq 0\}} + (1-p)\eta_2 e^{y\eta_2} \cdot 1_{\{y \leq 0\}}, \]

where \( p \) is the probability of an upward jump and \( \eta_1, \eta_2 \) are constant parameters.

2.3 Two-factor Jump Diffusion Model

Intuitively, it is easy to extend single-asset cases to multi-asset cases. For example, in two dimensions, by introducing the correlation coefficient between the Brownian motions of two different underlying prices, the stochastic differential equations for the underlying prices \( S_1, S_2 \) can be written as:

\[ \frac{dS_1}{S_1} = \mu_1 dt + \sigma_1 dZ_1 + (\eta_1 - 1)dQ_1, \]  

(2.5) 

\[ \frac{dS_2}{S_2} = \mu_2 dt + \sigma_2 dZ_2 + (\eta_2 - 1)dQ_2, \]  

(2.6) 

\[ dZ_1 dZ_2 = \rho dt \]  

(2.7)
where

\( \mu_1, \mu_2 = \text{drift rates}, \)
\( \sigma_1, \sigma_2 = \text{underlying volatilities}, \)
\( dZ_1, dZ_2 = \text{increments of standard Brownian motions}, \)
\( \eta_1 - 1, \eta_2 - 1 = \text{impulse functions producing jumps from } (S_1, S_2) \text{ to } (S_1 \eta_1, S_2 \eta_2), \)
\( dq_1 = \begin{cases} 
0, & \text{with probability } 1 - \lambda_1 dt, \\
1, & \text{with probability } \lambda_1 dt,
\end{cases} \)
\( dq_2 = \begin{cases} 
0, & \text{with probability } 1 - \lambda_2 dt, \\
1, & \text{with probability } \lambda_2 dt,
\end{cases} \)
\( \lambda_1, \lambda_2 = \text{mean arrival rates of Poisson jumps}, \)
\( \rho = \text{correlation of two standard Brownian motions}. \)

Define \( V(S_1, S_2, \tau) \) as the two-asset option value. As before, the partial integro-differential equation (PIDE) for \( V(S_1, S_2, \tau) \) can be obtained by Ito’s Lemma and no-arbitrage arguments as:

\[
V_{\tau} = \frac{\sigma_1^2 S_1^2}{2} V_{S_1 S_1} + \frac{\sigma_2^2 S_2^2}{2} V_{S_2 S_2} + (r - \lambda_1 \kappa_1) S_1 V_{S_1} + (r - \lambda_2 \kappa_2) S_2 V_{S_2} + \rho \sigma_1 \sigma_2 S_1 S_2 V_{S_1 S_2}
\]
\[
- (r + \lambda_1 + \lambda_2) V + \left( \lambda_1 \int_0^\infty V(S_1 \eta_1) g(\eta_1) d\eta_1 \right) + \left( \lambda_2 \int_0^\infty V(S_2 \eta_2) g(\eta_2) d\eta_2 \right)
\]

where

\( T = \text{expiry time}, \)
\( r = \text{risk free rate}, \)
\( \tau = T - t, \text{ where } t \text{ is current time}, \)
\( \kappa_1 = E[\eta_1 - 1], \text{ where } E[\eta_1] = \int_0^\infty \eta_1 g(\eta_1) d\eta_1, \)
\( \kappa_2 = E[\eta_2 - 1], \text{ where } E[\eta_2] = \int_0^\infty \eta_2 g(\eta_2) d\eta_2, \)
\( g(\eta_1) = \text{probability density function of the jump magnitude of } S_1, \)
\( g(\eta_2) = \text{probability density function of the jump magnitude of } S_2. \)
Examples of the initial conditions include:

\[ V(S_1, S_2, 0) = \max(B_2 S_2 - B_1 S_1 - K, 0), \quad \text{for spread call option} \quad (2.9) \]

or

\[ V(S_1, S_2, 0) = \max(K - B_2 S_2 + B_1 S_1, 0), \quad \text{for spread put option} \quad (2.10) \]

or

\[ V(S_1, S_2, 0) = \max(S_1 - K_1, K_2 - S_2, 0), \quad \text{for dual strike option} \quad (2.11) \]

where \(K, K_1, K_2, B_1\) and \(B_2\) are parameters for various multi-asset options.

### 2.4 Shared-jump Diffusion Model

Slightly different from equations (2.5) and (2.6), the shared-jump here is driven by a single Poisson process. Assume two underlying assets \(S_1, S_2\) follow the stochastic differential equations:

\[ \frac{dS_1}{S_1} = \mu_1 dt + \sigma_1 dZ_1 + (\eta - 1)dq, \quad (2.12) \]

\[ \frac{dS_2}{S_2} = \mu_2 dt + \sigma_2 dZ_2 + (\eta - 1)dq, \quad (2.13) \]

\[ dZ_1 dZ_2 = \rho dt \quad (2.14) \]

where

\[ dq = \begin{cases} 0, & \text{with probability } 1 - \lambda dt, \\ 1, & \text{with probability } \lambda dt, \end{cases} \]

\(\mu_1, \mu_2 = \text{drift rates},\)

\(\sigma_1, \sigma_2 = \text{underlying volatilities},\)

\(dZ_1, dZ_2 = \text{increments of standard Brownian motions},\)

\(\eta - 1 = \text{impulse function producing the shared-jump from } (S_1, S_2) \text{ to } (S_1 \eta, S_2 \eta),\)

\(\lambda = \text{mean arrival rate of Poisson jump},\)

\(\rho = \text{correlation of two standard Brownian motions}.\)

Let \(V(S_1, S_2, t)\) be the two-asset option value under the shared-jump diffusion model. The partial integro-differential equation (PIDE) for \(V(S_1, S_2, t)\) can be derived by Ito’s Lemma and constructing a hedging portfolio.
By Ito’s Lemma for jump processes, the total variation of $\mathcal{V}(S_1, S_2, t)$ can be written as:

$$
\begin{align*}
\mathcal{V} &= V_i dt + V_{S_1} (\mu_1 S_1 dt + \sigma_1 S_1 dZ_1) + V_{S_2} (\mu_2 S_2 dt + \sigma_2 S_2 dZ_2) + V_{S_1 S_2} \rho \sigma_1 \sigma_2 S_1 S_2 dt \\
&+ \frac{1}{2} V_{S_1 S_1} \sigma_1^2 S_1^2 dt + \frac{1}{2} V_{S_2 S_2} \sigma_2^2 S_2^2 dt + [\mathcal{V}(S_1 \eta, S_2 \eta, t) - \mathcal{V}(S_1, S_2, t)] dq.
\end{align*}
$$

By setting

$$
\begin{align*}
d\mathcal{V} &= \alpha dt + \beta dZ_1 + \gamma dZ_2 + \Delta \mathcal{V} dq, \\
\alpha &= V_i + V_{S_1} \mu_1 S_1 + V_{S_2} \mu_2 S_2 \\
&+ \frac{1}{2} V_{S_1 S_1} \sigma_1^2 S_1^2 + \frac{1}{2} V_{S_2 S_2} \sigma_2^2 S_2^2 + V_{S_1 S_2} \rho \sigma_1 \sigma_2 S_1 S_2, \\
\beta &= V_{S_1} \sigma_1 S_1, \\
\gamma &= V_{S_2} \sigma_2 S_2, \\
\Delta \mathcal{V} &= [\mathcal{V}(S_1 \eta, S_2 \eta, t) - \mathcal{V}(S_1, S_2, t)].
\end{align*}
$$

Consider the portfolio $\Pi$ including four contracts $\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3$ and $\mathcal{V}_4$ such that

$$
\Pi = n_1 \mathcal{V}_1 + n_2 \mathcal{V}_2 + n_3 \mathcal{V}_3 + n_4 \mathcal{V}_4.
$$

Hence,

$$
\begin{align*}
d\Pi &= n_1 d\mathcal{V}_1 + n_2 d\mathcal{V}_2 + n_3 d\mathcal{V}_3 + n_4 d\mathcal{V}_4 \\
&= n_1 (\alpha_1 dt + \beta_1 dZ_1 + \gamma_1 dZ_2 + \Delta \mathcal{V}_1 dq) \\
&+ n_2 (\alpha_2 dt + \beta_2 dZ_1 + \gamma_2 dZ_2 + \Delta \mathcal{V}_2 dq) \\
&+ n_3 (\alpha_3 dt + \beta_3 dZ_1 + \gamma_3 dZ_2 + \Delta \mathcal{V}_3 dq) \\
&+ n_4 (\alpha_4 dt + \beta_4 dZ_1 + \gamma_4 dZ_2 + \Delta \mathcal{V}_4 dq) \\
&= (n_1 \alpha_1 + n_2 \alpha_2 + n_3 \alpha_3 + n_4 \alpha_4) dt \\
&+ (n_1 \beta_1 + n_2 \beta_2 + n_3 \beta_3 + n_4 \beta_4) dZ_1 \\
&+ (n_1 \gamma_1 + n_2 \gamma_2 + n_3 \gamma_3 + n_4 \gamma_4) dZ_2 \\
&+ (n_1 \Delta \mathcal{V}_1 + n_2 \Delta \mathcal{V}_2 + n_3 \Delta \mathcal{V}_3 + n_4 \Delta \mathcal{V}_4) dq.
\end{align*}
$$

Eliminate the random terms $dZ_1, dZ_2$ and $dq$ by setting

$$
\begin{align*}
n_1 \beta_1 + n_2 \beta_2 + n_3 \beta_3 + n_4 \beta_4 &= 0, \\
n_1 \gamma_1 + n_2 \gamma_2 + n_3 \gamma_3 + n_4 \gamma_4 &= 0, \\
n_1 \Delta \mathcal{V}_1 + n_2 \Delta \mathcal{V}_2 + n_3 \Delta \mathcal{V}_3 + n_4 \Delta \mathcal{V}_4 &= 0.
\end{align*}
$$
Thus, the portfolio $\Pi$ is risk-less. Let $r$ be the risk free rate, so that
\[
d\Pi = r\Pi \, dt. \tag{2.19}
\]

From equations (2.17), (2.18) and (2.19), we get:
\[
(n_1\alpha_1 + n_2\alpha_2 + n_3\alpha_3 + n_4\alpha_4) = (n_1\mathcal{V}_1 + n_2\mathcal{V}_2 + n_3\mathcal{V}_3 + n_4\mathcal{V}_4)r. \tag{2.20}
\]

Putting equations (2.18) and (2.20) together gives
\[
\begin{bmatrix}
\beta_1 & \beta_2 & \beta_3 & \beta_4 \\
\gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 \\
\Delta\mathcal{V}_1 & \Delta\mathcal{V}_2 & \Delta\mathcal{V}_3 & \Delta\mathcal{V}_4 \\
\alpha_1 - r\mathcal{V}_1 & \alpha_2 - r\mathcal{V}_2 & \alpha_3 - r\mathcal{V}_3 & \alpha_4 - r\mathcal{V}_4 \\
\end{bmatrix}
\begin{bmatrix}
n_1 \\
n_2 \\
n_3 \\
n_4 \\
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}. \tag{2.21}
\]

Equation (2.21) has a non-zero solution only if the rows are linearly dependent. So, there must exist $\lambda_{B_1}(S_1, S_2, t)$, $\lambda_{B_2}(S_1, S_2, t)$ and $\lambda_J(S_1, S_2, t)$ such that
\[
\begin{align*}
(\alpha_1 - r\mathcal{V}_1) &= \lambda_{B_1}\beta_1 + \lambda_{B_2}\gamma_1 - \lambda_J\Delta\mathcal{V}_1, \\
(\alpha_2 - r\mathcal{V}_2) &= \lambda_{B_1}\beta_2 + \lambda_{B_2}\gamma_2 - \lambda_J\Delta\mathcal{V}_2, \\
(\alpha_3 - r\mathcal{V}_3) &= \lambda_{B_1}\beta_3 + \lambda_{B_2}\gamma_3 - \lambda_J\Delta\mathcal{V}_3, \\
(\alpha_4 - r\mathcal{V}_4) &= \lambda_{B_1}\beta_4 + \lambda_{B_2}\gamma_4 - \lambda_J\Delta\mathcal{V}_4.
\end{align*}
\]

It can be shown that $\lambda_J \geq 0$ avoids any arbitrage opportunity. Dropping the subscripts gives
\[
(\alpha - r\mathcal{V}) = \lambda_{B_1}\beta + \lambda_{B_2}\gamma - \lambda_J\Delta\mathcal{V} \tag{2.22}
\]
and substituting $\alpha, \beta, \gamma$ and $\Delta\mathcal{V}$ leads to
\[
\mathcal{V}_t + (\mu_1 - \lambda_{B_1}\sigma_1)\mathcal{V}_{S_1}S_1 + (\mu_2 - \lambda_{B_2}\sigma_2)\mathcal{V}_{S_2}S_2 + \mathcal{V}_{S_1S_2}\rho\sigma_1\sigma_2S_1S_2
+ \frac{1}{2}\mathcal{V}_{S_1S_1}\sigma_1^2S_1^2 + \frac{1}{2}\mathcal{V}_{S_2S_2}\sigma_2^2S_2^2 - r\mathcal{V} + \lambda_J[\mathcal{V}(S_1\eta, S_2\eta, t) - \mathcal{V}(S_1, S_2, t)] = 0. \tag{2.23}
\]

Suppose $\mathcal{V}_3 = S_1$ and $\mathcal{V}_4 = S_2$ are two traded underlying assets. In this case, we have:
\[
\begin{align*}
\alpha_3 &= \mu_1S_1, & \alpha_4 &= \mu_2S_2, \\
\beta_3 &= \sigma_1S_1, & \beta_4 &= 0, \\
\gamma_3 &= 0, & \gamma_4 &= \sigma_2S_2, \\
\Delta\mathcal{V}_3 &= (\eta - 1)S_1, & \Delta\mathcal{V}_4 &= (\eta - 1)S_2.
\end{align*}
\]
Substituting equation (2.25) into equation (2.23) gives

\[
(\mu_1 S_1 - r S_1) = \lambda_{B_1} \sigma_1 S_1 + \lambda_{B_2} (0) - \lambda_f (\eta - 1) S_1,
\]

\[
(\mu_2 S_2 - r S_2) = \lambda_{B_1} (0) + \lambda_{B_2} \sigma_2 S_2 - \lambda_f (\eta - 1) S_2.
\]

(2.24)

After eliminating common factors in equation (2.24), it is easy to obtain:

\[
\mu_1 - \lambda_{B_1} \sigma_1 = r - \lambda_f (\eta - 1),
\]

\[
\mu_2 - \lambda_{B_2} \sigma_2 = r - \lambda_f (\eta - 1).
\]

(2.25)

Substituting equation (2.25) into equation (2.23) gives

\[
\mathcal{V}_t + [r - \lambda_f (\eta - 1)] \mathcal{V}_S S_1 + [r - \lambda_f (\eta - 1)] \mathcal{V}_S S_2 + \mathcal{V}_S S_2 \rho \sigma_1 \sigma_2 S_1 S_2
\]

\[
+ \frac{1}{2} \mathcal{V}_S S_1 \sigma_1^2 S_1^2 + \frac{1}{2} \mathcal{V}_S S_2 \sigma_2^2 S_2^2 - r \mathcal{V} + \sum_{i=1}^{n} \lambda^i_j [\mathcal{V}(S_1 \eta_i, S_2 \eta_i, t) - \mathcal{V}(S_1, S_2, t)] = 0.
\]

(2.26)

Assume the number of jump states is finite, which means the asset price $S$ may jump to any states $S_i$ after a jump where $i = 1, \cdots, n$. By the hedging arguments above, use $n + 3$ hedging instruments so that the diffusion and jumps will be hedged perfectly. Then equation (2.23) can be written as:

\[
\mathcal{V}_t + (\mu_1 - \lambda_{B_1} \sigma_1) \mathcal{V}_S S_1 + (\mu_2 - \lambda_{B_2} \sigma_2) \mathcal{V}_S S_2 + \mathcal{V}_S S_2 \rho \sigma_1 \sigma_2 S_1 S_2
\]

\[
+ \frac{1}{2} \mathcal{V}_S S_1 \sigma_1^2 S_1^2 + \frac{1}{2} \mathcal{V}_S S_2 \sigma_2^2 S_2^2 - r \mathcal{V} + \sum_{i=1}^{n} \lambda^i_j [\mathcal{V}(S_1 \eta_i, S_2 \eta_i, t) - \mathcal{V}(S_1, S_2, t)] = 0.
\]

(2.27)

Similarly, if the underlying assets can also be used for hedging, equation (2.26) can be transformed into:

\[
\mathcal{V}_t + [r - \sum_{i=1}^{n} \lambda^i_j (\eta_i - 1)] \mathcal{V}_S S_1 + [r - \sum_{i=1}^{n} \lambda^i_j (\eta_i - 1)] \mathcal{V}_S S_2 + \mathcal{V}_S S_2 \rho \sigma_1 \sigma_2 S_1 S_2
\]

\[
+ \frac{1}{2} \mathcal{V}_S S_1 \sigma_1^2 S_1^2 + \frac{1}{2} \mathcal{V}_S S_2 \sigma_2^2 S_2^2 - r \mathcal{V} + \sum_{i=1}^{n} \lambda^i_j [\mathcal{V}(S_1 \eta_i, S_2 \eta_i, t) - \mathcal{V}(S_1, S_2, t)] = 0.
\]

(2.28)

Let $g(\eta_i) = \frac{\lambda^i_j}{\sum_{i=1}^{n} \lambda^i_j}$ and $\lambda = \sum_{i=1}^{n} \lambda^i_j$, then equation (2.27) becomes:

\[
\mathcal{V}_t + (\mu_1 - \lambda_{B_1} \sigma_1) \mathcal{V}_S S_1 + (\mu_2 - \lambda_{B_2} \sigma_2) \mathcal{V}_S S_2 + \mathcal{V}_S S_2 \rho \sigma_1 \sigma_2 S_1 S_2
\]

\[
+ \frac{1}{2} \mathcal{V}_S S_1 \sigma_1^2 S_1^2 + \frac{1}{2} \mathcal{V}_S S_2 \sigma_2^2 S_2^2 - r \mathcal{V} + \lambda \sum_{i=1}^{n} g(\eta_i) [\mathcal{V}(S_1 \eta_i, S_2 \eta_i, t) - \mathcal{V}(S_1, S_2, t)] = 0.
\]

(2.29)
In addition, \( g(\eta) \geq 0 \) and \( \lambda \geq 0 \) because \( \lambda_j^i \geq 0 \).

Now, if we take the limit as the number of jump states goes to infinity, then \( g(\eta) \) will tend to a continuous distribution and infinite hedging instruments are required. Thus, equation (2.29) can be represented as:

\[
\nu_t + (\mu_1 - \lambda B_1 \sigma_1) \nu S_1 + (\mu_2 - \lambda B_2 \sigma_2) \nu S_2 + \nu S_1 \sigma_1 \sigma_2 S_1 S_2 \\
+ \frac{1}{2} \nu S_1 \sigma_1^2 S_1^2 + \frac{1}{2} \nu S_2 \sigma_2^2 S_2^2
\]

\[
- r \nu + \lambda \int_0^\infty [\nu(S_1 \eta, S_2 \eta, t) - \nu(S_1, S_2, t)] g(\eta) d\eta = 0.
\]

(2.30)

In the case where the underlying assets can be used to hedge the risk, we have:

\[
\nu_t + (r - \lambda E[\eta - 1]) \nu S_1 + (r - \lambda E[\eta - 1]) \nu S_2 + \nu S_1 \sigma_1 \sigma_2 S_1 S_2 \\
+ \frac{1}{2} \nu S_1 \sigma_1^2 S_1^2 + \frac{1}{2} \nu S_2 \sigma_2^2 S_2^2
\]

\[
- r \nu + \lambda \int_0^\infty [\nu(S_1 \eta, S_2 \eta, t) - \nu(S_1, S_2, t)] g(\eta) d\eta = 0.
\]

(2.31)

It is crucial to point out that \( \lambda \) and \( g(\eta) \) here are not the real mean arrival rate of Poisson jump and real probability density function of the jump magnitude because they are obtained by the hedging arguments mentioned above. Thus, \( \lambda \) and \( g(\eta) \) must be derived by calibration to the financial market data.

Define \( \tau = T - t \) and \( \kappa = E[\eta - 1] = \int_0^\infty (\eta - 1) g(\eta) d\eta \), the partial integro-differential equation (PIDE) for \( \nu(S_1, S_2, \tau) \) can be directly derived from equations (2.31):

\[
\nu_\tau = \frac{1}{2} \sigma_1^2 S_1^2 \nu S_1 s_1 + \frac{1}{2} \sigma_2^2 S_2^2 \nu S_2 s_2 + (r - \lambda \kappa) S_1 \nu S_1 + (r - \lambda \kappa) S_2 \nu S_2 \\
+ \rho \sigma_1 \sigma_2 S_1 S_2 \nu S_1 s_1 - (r + \lambda) \nu + \lambda \int_0^\infty \nu(S_1 \eta, S_2 \eta) g(\eta) d\eta
\]

(2.32)

with the initial condition given by equation (2.9), (2.10) or (2.11).

A log-transformation will lead to a PIDE with constant coefficients and a cross-correlation integral. Define \( x_1 = \log(S_1), x_2 = \log(S_2) \) and let \( v(x_1, x_2, \tau) = \nu(S_1, S_2, \tau) \). The relationships between the partial derivatives of \( v \) with respect to \( x_1, x_2 \) and partial derivatives of \( \nu \) with respect to \( S_1, S_2 \) can be seen as:

\[
\nu S_1 = \frac{v(x_1)}{e^{x_1}}, \quad \nu S_2 = \frac{v(x_2)}{e^{x_2}}.
\]

\[
\nu S_1 S_1 = \frac{v(x_1) x_1 - v(x_1)}{e^{2x_1}}, \quad \nu S_2 S_2 = \frac{v(x_2) x_2 - v(x_2)}{e^{2x_2}}, \quad \nu S_1 S_2 = \frac{v(x_1) x_2}{e^{x_1} e^{x_2}}.
\]
Let $\eta = e^y$, then $d\eta = e^y dy$. Plugging the above into equation (2.32) gives the PIDE in terms of $v$:

\[
v_\tau = \frac{1}{2} \sigma_1^2 v_{x_1 x_1} + \frac{1}{2} \sigma_2^2 v_{x_2 x_2} + (r - \lambda \kappa - \frac{1}{2} \sigma_1^2) v_{x_1} + (r - \lambda \kappa - \frac{1}{2} \sigma_2^2) v_{x_2} \\
+ \rho \sigma_1 \sigma_2 v_{x_1 x_2} - (r + \lambda) v + \lambda \left( \int_{-\infty}^{\infty} v(x_1 + y, x_2 + y, \tau) g(e^y) e^y dy \right).
\] (2.33)

Introducing $f(x) = g(e^x)e^x$ and substituting into the PIDE (2.33) gives:

\[
v_\tau = \frac{1}{2} \sigma_1^2 v_{x_1 x_1} + \frac{1}{2} \sigma_2^2 v_{x_2 x_2} + (r - \lambda \kappa - \frac{1}{2} \sigma_1^2) v_{x_1} + (r - \lambda \kappa - \frac{1}{2} \sigma_2^2) v_{x_2} \\
+ \rho \sigma_1 \sigma_2 v_{x_1 x_2} - (r + \lambda) v + \lambda \left( \int_{-\infty}^{\infty} v(x_1 + y, x_2 + y, \tau) f(y) dy \right).
\] (2.34)

Hence, with log-transformation and changes of variables, the original PIDE (2.32) may be represented as the PIDE (2.34), which contains constant coefficients and a cross-correlation integral.
Chapter 3

Fourier Space Time-stepping Method

3.1 Introduction

The Fourier space time-stepping (FST) method was first developed by Jackson, Jaimungal and Surkov (2008)[8]. This method uses the continuous Fourier transform, which is a linear operator, to map the spatial derivatives into multiplications in Fourier space.

By the properties of the Fourier transform, the PIDE derived previously in (2.23) can be converted into a linear first-order ordinary differential equation (ODE) in Fourier space. This is then straightforward to solve in closed-form.

For one-factor and two-factor jump diffusion models, the details of implementing the FST method are demonstrated by Lippa (2013)[11]. In this chapter, the FST algorithm under a shared-jump diffusion model will be discussed.

3.2 Continuous Fourier Transform

The continuous Fourier transform (CFT) maps a function in the space domain $f(x)$ into a function in the frequency domain $F(k)$. Here, $x$ and $k$ can be scalars or vectors since it is straightforward to generalize a one-dimensional continuous Fourier transform into higher dimensions.

With the definition of one-dimensional continuous Fourier transform of a function $f(x)$ being:

$$F(k) = \mathcal{F}[f(x)](k) := \int_{-\infty}^{\infty} f(x)e^{-i2\pi k x} dx, \quad (3.1)$$
the definition of one-dimensional inverse continuous Fourier transform (ICFT) of a function $F(k)$ is:

$$f(x) = F^{-1}[F(k)](x) := \int_{-\infty}^{\infty} F(k) e^{i2\pi kx} dk. \quad (3.2)$$

With the definition of two-dimensional continuous Fourier transform of a function $f(x)$ being:

$$F(k_1, k_2) = F[f(x_1, x_2)](k_1, k_2) := \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x_1, x_2) e^{-i2\pi(k_1 x_1 + k_2 x_2)} dx_1 dx_2, \quad (3.3)$$

the definition of two-dimensional inverse continuous Fourier transform of a function $F(k)$ is:

$$f(x_1, x_2) = F^{-1}[F(k_1, k_2)](x_1, x_2) := \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(k_1, k_2) e^{i2\pi(k_1 x_1 + k_2 x_2)} dk_1 dk_2. \quad (3.4)$$

There are some useful properties of Fourier transform for computations. The Fourier transform of the partial derivative of a function $v(x, \tau)$ with respect to $\tau$ can be represented as:

$$\mathcal{F}\left[\frac{\partial}{\partial \tau} v(x, \tau)\right](k) = \frac{\partial}{\partial \tau} \mathcal{F}[v(x, \tau)](k). \quad (3.5)$$

Also, the Fourier transform of the partial derivative of a function $v(x, \tau)$ with respect to $x$ can be represented as:

$$\mathcal{F}\left[\frac{\partial^n}{\partial x^n} v(x, \tau)\right](k) = (2\pi i k)^n \mathcal{F}[v(x, \tau)](k). \quad (3.6)$$

In addition, the two-dimensional Fourier transform of the the second partial derivative of a function $v(x, y, \tau)$ with respect to $x$ and $y$ can be represented as:

$$\mathcal{F}\left[\frac{\partial^2}{\partial x \partial y} v(x, y, \tau)\right](k_1, k_2) = (2\pi i k_1)(2\pi i k_2) \mathcal{F}[v(x, y, \tau)](k_1, k_2). \quad (3.7)$$

### 3.3 Discrete Fourier Transform

Generally, it is impossible to compute the exact result of continuous Fourier transform in closed-form. So, the discrete Fourier transform (DFT) is introduced to approximate CFT
for numerical solutions. This has error $O(\Delta x^2)$, where $\Delta x$ is the constant spacing in the $x$ direction. In the following, we define the DFT pairs as:

$$
\hat{F}_n = \sum_{m=0}^{N-1} f(x_m) e^{-i2\pi \frac{nm}{N}},
$$

$$
f_m = \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} e^{i2\pi \frac{nm}{N}} \hat{F}_n.
$$

If the original domain of $[-\infty, \infty]$ is truncated to the new domain $\Omega = [x_{\min}, x_{\max}]$, the continuous Fourier transform (3.1) is approximated on domain $\Omega$ as:

$$
F(k) \approx \int_{x_{\min}}^{x_{\max}} f(x) e^{-i2\pi kx} dx.
$$

(3.8)

Discretizing the real domain $[x_{\min}, x_{\max}]$ as:

$$
x_m = x_{\min} + m \cdot \Delta x
$$

(3.9)

where $m = 0, 1, \cdots, N - 1$, $\Delta x = \frac{x_{\max} - x_{\min}}{N}$ and $N$ is the total number of nodes in the real domain.

Meanwhile, the Fourier domain can also be discretized as:

$$
k_n = \frac{n}{x_{\max} - x_{\min}}
$$

(3.10)

where $n = -\frac{N}{2} + 1, \cdots, \frac{N}{2}$. By the Nyquist frequency conditions, the maximum frequency is $\pm \frac{N}{2}$. Therefore, the allowable frequencies are:

$$
\{(-\frac{N}{2} + 1), \cdots, \frac{N}{2}\}
$$

(3.11)

We can approximate this using the trapezoidal rule, so equation (3.8) becomes:

$$
F(k) \approx \int_{x_{\min}}^{x_{\max}} f(x) e^{-i2\pi kx} dx
$$

$$
\approx \sum_{m=0}^{N-1} f(x_m) e^{-i2\pi kx_m} \Delta x + O(\Delta x^2).
$$

(3.12)
In Fourier space, for \( k_{-\frac{N}{2}+1}, \cdots, k_{\frac{N}{2}} \), we use equations (3.10) and (3.12) to get:

\[
F_n = F(k_n) \approx \sum_{m=0}^{N-1} f(x_m) e^{-i2\pi k_n x_m} \Delta x
\]

\[
= \Delta x \sum_{m=0}^{N-1} f(x_m) e^{-i2\pi k_n (x_{min} + m\Delta x)}
\]

\[
= e^{-i2\pi k_n x_{min}} \Delta x \sum_{m=0}^{N-1} f(x_m) e^{-i2\pi \frac{2m}{N}}.
\]  

(3.13)

Define \( \hat{F}_n := \sum_{m=0}^{N-1} f(x_m) e^{-i2\pi \frac{nm}{N}} \). Then, equation (3.13) can be rewritten as:

\[
F_n = e^{-i2\pi k_n x_{min}} \cdot \Delta x \cdot \hat{F}_n
\]  

(3.14)

where \( \hat{F}_0, \hat{F}_1, \cdots, \hat{F}_{N-1} \) are the discrete Fourier transforms of \( f_0, f_1, \cdots, f_{N-1} \).

Conversely, the inverse continuous Fourier transform (3.2) can be approximated by:

\[
f(x) = \int_{-\infty}^{\infty} F(k) e^{i2\pi k x} \, dk
\]

\[
\approx \int_{-\frac{N}{2}}^{\frac{N}{2}} F(k) e^{i2\pi k x} \, dk
\]

\[
\approx \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} F(k_n) e^{i2\pi k_n x} \Delta k + O(\Delta k^2)
\]  

(3.15)

where \( k_n = n \cdot \Delta k \) and \( \Delta k = \frac{1}{x_{max} - x_{min}} \).
In this situation:

\[ f_m = f(x_m) \approx \sum_{n=-N/2+1}^{N/2} F(k_n) e^{i2\pi k_n x_m} \Delta k \]

\[ = \sum_{n=-N/2+1}^{N/2} F(k_n) e^{i2\pi k_n (x_{min} + m\Delta x)} \Delta k \]

\[ = \sum_{n=-N/2+1}^{N/2} \left( e^{-i2\pi k_n x_{min}} \cdot \Delta x \cdot \hat{F}_n \right) e^{i2\pi k_n (x_{min} + m\Delta x)} \Delta k \]  

\[ = \sum_{n=-N/2+1}^{N/2} \Delta x \cdot e^{i2\pi \frac{nm}{N}} \hat{F}_n \cdot \Delta k \]

\[ = \frac{1}{N} \sum_{n=-N/2+1}^{N/2} e^{i2\pi \frac{nm}{N}} \hat{F}_n \]

(3.16)

where \( f_0, f_1, \cdots, f_{N-1} \) are the inverse discrete Fourier transforms (IDFT) of \( \hat{F}_0, \hat{F}_1, \cdots, \hat{F}_{N-1} \).

### 3.4 FST Method under a Shared-jump Diffusion Model

#### 3.4.1 Fourier Transform

Given a partial integro-differential differential equation in the form of equation (2.23), with constant coefficients and a cross-correlation integral term, the option pricing problem can be reduced to solving the PIDE by the Fourier space time-stepping method. Applying the two-dimensional continuous Fourier transform \( \mathcal{F} \) on the PIDE (2.23) gives:

\[ \mathcal{F}[v_{\tau}](k_1, k_2) = \frac{1}{2} \sigma_1^2 \mathcal{F}[v_{x_1x_1}](k_1, k_2) + \frac{1}{2} \sigma_2^2 \mathcal{F}[v_{x_2x_2}](k_1, k_2) + (r - \lambda \kappa - \frac{1}{2} \sigma_1^2) \mathcal{F}[v_{x_1}](k_1, k_2) + (r - \lambda \kappa - \frac{1}{2} \sigma_2^2) \mathcal{F}[v_{x_2}](k_1, k_2) + \rho \sigma_1 \sigma_2 \mathcal{F}[v_{x_1x_2}](k_1, k_2) - (r + \lambda) \mathcal{F}[v](k_1, k_2) + \lambda \mathcal{F} \left[ \int_{-\infty}^{\infty} v(x_1 + y, x_2 + y) f(y) dy \right] (k_1, k_2) \]  

(3.17)

18
where the transform variables \( k_1, k_2 \) represent the frequencies.

In equation (3.17), except for the two-dimensional cross-correlation integral term, all other terms can be easily simplified by properties (3.5), (3.6) and (3.7). Take the two-dimensional Fourier transform of cross-correlation integral term to get:

\[
\mathcal{F}[\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} v(x_1 + y, x_2 + y) f(y) dy] (k_1, k_2)
\]

\[
= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left( \int_{-\infty}^{\infty} v(x_1 + y, x_2 + y) f(y) dy \right) e^{-2\pi i (k_1 x_1 + k_2 x_2)} dx_1 dx_2.
\]  

(3.18)

Let \( w = -y \), so that \( dy = -dw \). Plugging into (3.18) gives:

\[
\mathcal{F}[\int_{-\infty}^{\infty} v(x_1 + y, x_2 + y) f(y) dy] (k_1, k_2)
\]

\[
= \int_{-\infty}^{\infty} f(-w) \left( \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} v(x_1 - w, x_2 - w) e^{-2\pi i (k_1 x_1 + k_2 x_2)} dx_1 dx_2 \right) dw.
\]  

(3.19)

Let \( u_1 = x_1 - w, u_2 = x_2 - w \), so that \( du_1 = dx_1, du_2 = dx_2 \). Plugging into (3.19) gives:

\[
\mathcal{F}[\int_{-\infty}^{\infty} v(x_1 + y, x_2 + y) f(y) dy] (k_1, k_2)
\]

\[
= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} v(u_1, u_2) e^{-2\pi i (k_1 u_1 + k_2 u_2)} du_1 du_2 \left( \int_{-\infty}^{\infty} f(-w) dw \right)
\]

\[
= \int_{-\infty}^{\infty} e^{-2\pi i (k_1 u_1 + k_2 u_2)} f(-w) dw \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} v(u_1, u_2) e^{-2\pi i (k_1 u_1 + k_2 u_2)} du_1 du_2
\]

\[
= \mathcal{F}[f(-x)] (k_1 + k_2) \mathcal{F}[v(u_1, u_2)] (k_1, k_2)
\]

\[
= \mathcal{F}[f(x)] (k_1 + k_2) \mathcal{F}[v](k_1, k_2)
\]  

(3.20)

where \( \bar{z} \) denotes the complex conjugate of \( z \).

By the properties of Fourier Transform (3.5), (3.6), (3.7) and (3.20), equation (3.17) can be simplified to:

\[
\frac{\partial}{\partial \tau} \mathcal{F}[v](k_1, k_2) = \frac{1}{2} \sigma_1^2 (2\pi i k_1)^2 \mathcal{F}[v](k_1, k_2) + \frac{1}{2} \sigma_2^2 (2\pi i k_2)^2 \mathcal{F}[v](k_1, k_2)
\]

\[
+ (r - \lambda \kappa - \frac{1}{2} \sigma_1^2)(2\pi i k_1) \mathcal{F}[v](k_1, k_2) + (r - \lambda \kappa - \frac{1}{2} \sigma_2^2)(2\pi i k_2) \mathcal{F}[v](k_1, k_2)
\]

\[
+ \rho \sigma_1 \sigma_2 (2\pi i k_1)(2\pi i k_2) \mathcal{F}[v](k_1, k_2) - (r + \lambda) \mathcal{F}[v](k_1, k_2)
\]

\[
+ \lambda \mathcal{F}[f(x)] (k_1 + k_2) \mathcal{F}[v](k_1, k_2).
\]

(3.21)
Rearranging the terms in (3.21) gives:

\[
\frac{\partial}{\partial \tau} \mathcal{F}[v](k_1, k_2) = \mathcal{F}[v](k_1, k_2) \left( -\frac{1}{2} \sigma_1^2 (2\pi k_1)^2 - \frac{1}{2} \sigma_2^2 (2\pi k_2)^2 
+ (r - \lambda \kappa - \frac{1}{2} \sigma_1^2)(2\pi i k_1) + (r - \lambda \kappa - \frac{1}{2} \sigma_2^2)(2\pi i k_2)
- \rho \sigma_1 \sigma_2 (2\pi k_1)(2\pi k_2) - (r + \lambda) + \lambda \mathcal{F}[f(x)](k_1 + k_2) \right). 
\]

(3.22)

Define \( V(k_1, k_2) := \mathcal{F}[v](k_1, k_2), F(k_1 + k_2) := \mathcal{F}[f(x)](k_1 + k_2) \) so that the derivative with respect to \( \tau \) can be written as \( V_{\tau}(k_1, k_2) := \frac{\partial}{\partial \tau} \mathcal{F}[v](k_1, k_2) \). In addition, define the characteristic exponent \( \Psi(k_1, k_2) \) as:

\[
\Psi(k_1, k_2) = -\frac{1}{2} \sigma_1^2 (2\pi k_1)^2 - \frac{1}{2} \sigma_2^2 (2\pi k_2)^2 
+ (r - \lambda \kappa - \frac{1}{2} \sigma_1^2)(2\pi i k_1) + (r - \lambda \kappa - \frac{1}{2} \sigma_2^2)(2\pi i k_2)
- \rho \sigma_1 \sigma_2 (2\pi k_1)(2\pi k_2) - (r + \lambda) + \lambda \mathcal{F}[f(x)](k_1 + k_2). 
\]

(3.23)

With these new notations, equation (3.22) can be rewritten as:

\[
0 = V_{\tau}(k_1, k_2) - V(k_1, k_2) \cdot \Psi(k_1, k_2). 
\]

(3.24)

Hence, by using a change of variables and the properties of the Fourier transforms, the original PIDE (2.23) is converted into a linear ordinary differential equation (ODE) in \( \tau \) given by (3.24). This ODE can be solved in closed-form in Fourier space and the option price in real space can be transformed back from Fourier space by taking the inverse Fourier transform.

### 3.4.2 Solving the Ordinary Differential Equation

By introducing the integrating factor \( e^{-\Psi(k_1, k_2)\tau} \), the ODE (3.24) can be easily solved:

\[
0 = e^{-\Psi(k_1, k_2)\tau} V_{\tau}(k_1, k_2) - e^{-\Psi(k_1, k_2)\tau} V(k_1, k_2) \cdot \Psi(k_1, k_2)
= \frac{\partial}{\partial \tau} \left( V(k_1, k_2) e^{-\Psi(k_1, k_2)\tau} \right). 
\]

(3.25)

Integrating both sides of (3.25) with respect to \( \tau \):

\[
V(k_1, k_2) = C \cdot e^{\Psi(k_1, k_2)\tau}
\]
where \( C \) is constant.

Let \( V^u(k_1, k_2) := \mathcal{F}[v(x_1, x_2, \tau)](k_1, k_2) \) denote the Fourier transform of the option price at time \( \tau \). Then for any \( 0 \leq \tau \leq \tau_u \leq T \), given the Fourier transform of the option price at time \( \tau_l \), \( V^l \), the Fourier transform of the option price at time \( \tau_u \) can be computed by:

\[
V^u(k_1, k_2) = V^l(k_1, k_2) \cdot e^{\Psi(k_1, k_2)(\tau_u - \tau)}.
\] (3.26)

The inverse Fourier transform can be applied on (3.26) to recover the option value:

\[
v(x_1, x_2, \tau_u) = \mathcal{F}^{-1}[V^l(k_1, k_2) \cdot e^{\Psi(k_1, k_2)(\tau_u - \tau)}](x_1, x_2).
\] (3.27)

The equations (3.26) and (3.27) provide the main idea on how to time-step in Fourier space and recover the option prices in real space, using a continuous Fourier transform and its inverse.

### 3.4.3 Fourier Space Time-stepping

Note that equations (3.26) and (3.27) use the CFT and ICFT. In practice, DFT and IDFT are used to approximate CFT and ICFT to compute numerical solutions:

\[
V^u(k_{1m}, k_{2n}) \approx e^{-i2\pi(k_{1m} + k_{2n})x_{\min}} \cdot \Delta x_1 \Delta x_2 \cdot \hat{V}^u(n_1, n_2)
\] (3.28)

and

\[
V^l(k_{1m}, k_{2n}) \approx e^{-i2\pi(k_{1m} + k_{2n})x_{\min}} \cdot \Delta x_1 \Delta x_2 \cdot \hat{V}^l(n_1, n_2).
\] (3.29)

Therefore, equation (3.26) can be approximated by using (3.28) and (3.29), which after eliminating factors of \( e^{-i2\pi(k_{1m} + k_{2n})x_{\min}} \cdot \Delta x_1 \Delta x_2 \) gives:

\[
\hat{V}^u(n_1, n_2) = \hat{V}^l(n_1, n_2) \cdot e^{\Psi(k_1, k_2)(\tau_u - \tau_l)}.
\] (3.30)

The inverse discrete Fourier transform (IDFT) can be used to transform the option price in real space back, for any \( m_1, m_2 = 0, 1, \ldots, N - 1 \):

\[
IDFT[\hat{V}^u(n_1, n_2)](x_{1m_1}, x_{2m_2}) = IDFT[\hat{V}^l(n_1, n_2) \cdot e^{\Psi(k_1, k_2)(\tau_u - \tau_l)}](x_{1m_1}, x_{2m_2})
\] (3.31)

\[
v(x_{1m_1}, x_{2m_2}, \tau_u) = IDFT[e^{\Psi(k_1, k_2)(\tau_u - \tau_l)} \cdot DFT[v(x_{1m_1}, x_{2m_2}, \tau_l)]]
\]

Here, the computed value \( v(x_{1m_1}, x_{2m_2}, \tau_u) \) represents the option price when the underlying prices are \((S_1, S_2) = (e^{x_{1m_1}}, e^{x_{2m_2}}) \) at time \( \tau_u \).
3.4.4 Illustration of Method

To price the options under the shared-jump model, the FST method can be implemented to solve the PIDE given by equation (2.23) numerically.

The underlying prices domain \((S_1, S_2)\) is discretized by defining nodes in both of the \((S_1, S_2)\) directions denoted as \([S_{10}, S_{11}, \ldots, S_{1\text{max}}] \times [S_{20}, S_{21}, \ldots, S_{2\text{max}}]\), which are equally spaced in log space \(\log(S_1), \log(S_2)\).

For European options, there are no conditions imposed on the option until the expiry date. Hence, the time domain can be discretized as \(\tau^0 = 0, \tau^1 = T\) so that \(\Delta \tau = T\). By using the FST method, European options can be priced with only one time-step, from \(\tau^0 = 0\) to \(\tau^1 = T\).

For American options, the time domain should be discretized as \([\tau^0 = 0, \tau^1, \ldots, \tau^M = T]\) with \(\Delta \tau = \frac{T}{M}\) in order to take the early exercise boundary into consideration. When using the FST method, the computed option price needs to be updated during each time-step by checking the possibility of early exercise.
Chapter 4

Numerical Results

4.1 One-factor Jump Diffusion Cases

Under the one-factor jump diffusion model, the options pricing PIDE is given by equation (2.2):

\[ V_\tau = \frac{\sigma^2 S^2}{2} V_{SS} + (r - \lambda \kappa)SV_S - (r + \lambda)V + \left( \lambda \int_0^\infty V(S\eta)g(\eta)d\eta \right). \]

After doing a log transformation and Fourier transform, the characteristic exponent can be obtained as:

\[ \Psi(k) = -\frac{\sigma^2}{2} (2\pi k)^2 + (r - \lambda \kappa - \frac{\sigma^2}{2})(2\pi ik) - (r + \lambda) + \lambda \tilde{F}(k) \quad (4.1) \]

where \( F(k) \) is defined as the Fourier transform of the jump density function \( f(x) \) and the derivation of \( F(k) \) can be found in (Appendix A). In addition, \( \bar{z} \) denotes the complex conjugate of \( z \).

Precisely speaking, for the Merton jump density, we have

\[ f(y) = \frac{1}{\sqrt{2\pi\gamma}} e^{-\frac{1}{2}(\frac{y-\mu}{\gamma})^2}, \quad F(k) = e^{-2(\pi k \mu + (\pi k \gamma)^2)}. \quad (4.2) \]

For the Kou jump density, we get

\[ f(y) = p\eta_1 e^{-\eta_1 y} \cdot 1_{\{y \geq 0\}} + (1 - p)\eta_2 e^{\eta_2 y} \cdot 1_{\{y \leq 0\}}, \quad F(k) = \frac{p}{1 + 2\pi ik(\frac{1}{\eta_1})} + \frac{1 - p}{1 - 2\pi ik(\frac{1}{\eta_2})}. \quad (4.3) \]
4.1.1 Pricing Results

Table (4.1) shows the parameters for a European put under a one-factor jump diffusion model with the Merton jump density. The pricing results are listed in Table (4.2).

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>100</td>
</tr>
<tr>
<td>$K$</td>
<td>110</td>
</tr>
<tr>
<td>$r$</td>
<td>0.05</td>
</tr>
<tr>
<td>$q^1$</td>
<td>0.02</td>
</tr>
<tr>
<td>$T$</td>
<td>10</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>0.15</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>0.1</td>
</tr>
<tr>
<td>$\mu$</td>
<td>-1.08</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Table 4.1: Parameters for a European put under the one-factor Merton jump diffusion model

<table>
<thead>
<tr>
<th>Refinement</th>
<th>Nodes</th>
<th>Price (^2)</th>
<th>Change</th>
<th>Ratio (^3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>512</td>
<td>18.00485074</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1024</td>
<td>18.00393356</td>
<td>0.00091718</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2048</td>
<td>18.00370453</td>
<td>0.00022903</td>
<td>4.00462736</td>
</tr>
<tr>
<td>3</td>
<td>4096</td>
<td>18.00364730</td>
<td>0.00005722</td>
<td>4.00229568</td>
</tr>
<tr>
<td>4</td>
<td>8192</td>
<td>18.00363300</td>
<td>0.00001430</td>
<td>4.00114271</td>
</tr>
<tr>
<td>5</td>
<td>16384</td>
<td>18.00362943</td>
<td>0.00000358</td>
<td>4.00056890</td>
</tr>
</tbody>
</table>

Table 4.2: Pricing results of a European put under the one-factor Merton jump diffusion model: $[x_{\text{min}}, x_{\text{max}}] = [-7.5, 7.5]$

As shown in Tables (4.1) and (4.2), second order convergence is obtained for pricing a European put under the one-factor Merton jump diffusion model by FST method.

\(^1\)Note that $q$ is the continuous dividend rate, which shrinks risk-free rate $r$.

\(^2\)Reference price of 18.00362936 is given by Surkov (2009)[17].

\(^3\)Denote the option prices as $V_1, V_2, \cdots, V_6$, then the changes are defined as $V_1 - V_2, V_2 - V_3, \cdots, V_5 - V_6$, and ratios are defined as $\frac{V_2 - V_3}{V_2 - V_4}, \frac{V_3 - V_4}{V_3 - V_5}, \cdots, \frac{V_5 - V_6}{V_4 - V_5}$. Therefore, the binary logarithm of a ratio shows the numerical order of convergence.
Table (4.3) shows the parameters for a European call under a one-factor jump diffusion model with the Kou jump density. The pricing results are shown in Table (4.4).

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>100</td>
</tr>
<tr>
<td>$K$</td>
<td>110</td>
</tr>
<tr>
<td>$r$</td>
<td>0</td>
</tr>
<tr>
<td>$T$</td>
<td>1</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>0.2</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>0.2</td>
</tr>
<tr>
<td>$p$</td>
<td>0.5</td>
</tr>
<tr>
<td>$\eta_1$</td>
<td>3</td>
</tr>
<tr>
<td>$\eta_2$</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 4.3: Parameters for a European call under the one-factor Kou jump diffusion model

<table>
<thead>
<tr>
<th>Refinement</th>
<th>Nodes</th>
<th>Price$^4$</th>
<th>Change</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>512</td>
<td>7.30281862</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1024</td>
<td>7.28243068</td>
<td>0.02038794</td>
<td>25.08114610</td>
</tr>
<tr>
<td>2</td>
<td>2048</td>
<td>7.28161780</td>
<td>0.00081288</td>
<td>0.64181120</td>
</tr>
<tr>
<td>3</td>
<td>4096</td>
<td>7.28035126</td>
<td>0.00126654</td>
<td>4.00048234</td>
</tr>
<tr>
<td>4</td>
<td>8192</td>
<td>7.27995551</td>
<td>0.00007916</td>
<td>3.99940610</td>
</tr>
<tr>
<td>5</td>
<td>16384</td>
<td>7.27993383</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.4: Pricing results of a European call under the one-factor Kou jump diffusion model: $[x_{\min}, x_{\max}] = [-7.5, 7.5]$

As shown in Tables (4.3) and (4.4), second order convergence is obtained as well for pricing a European call under the one-factor Kou jump diffusion model by FST method.

4.1.2 Wrap-around Error

When implementing the FST method, a periodic domain is assumed implicitly, while the domain of the log underlying prices is actually aperiodic. Hence, the periodic assumption

$^4$Reference price of 7.27993383 is given by Surkov (2009).
causes option values on the ends of the grid to be wrapped around to the other side of the grid. This produces spurious solutions near both left and right ends.

The simple cure to remedy the wrap-around error is zero padding. For the domain of the log underlying prices, zeros are added and the Fourier transform is done on a grid with the size twice as the original length. At the same time, in the Fourier domain, the number of nodes is doubled as well. Then during the FST process, after applying the inverse Fourier transform, the added zeros are removed.

In detail, let $\mathcal{V}(S_i, \tau^m)$ denote the option value at node $S_i$, $i = 0, 1, \cdots, N - 1$, at time $\tau^m$. So, without padding, the Fourier transform is applied to the vector $\mathcal{V}^m$:

$$\mathcal{V}^m = [\mathcal{V}(S_0, \tau^m), \mathcal{V}(S_1, \tau^m), \cdots, \mathcal{V}(S_{N-1}, \tau^m)]$$

where $N$ is the total number of nodes in the $S$ direction. When using zero padding, the new vector $\hat{\mathcal{V}}^m$ is constructed by adding zero nodes to both ends of $\mathcal{V}^m$ in the $S$ direction:

$$\hat{\mathcal{V}}^m = \left[0, \cdots, 0, \mathcal{V}(S_0, \tau^m), \mathcal{V}(S_1, \tau^m), \cdots, \mathcal{V}(S_{N-1}, \tau^m), 0, \cdots, 0\right].$$

However, zero padding is not accurate enough since there may still have a huge gap between zeros and the values on the end, which leads to spurious results. Therefore, constant padding is introduced in this part.

For constant padding, rather than adding zeros to the original vector $\mathcal{V}^m$, the values on the very left and right ends are used to expand the domain. Define constants $c_l^m = \mathcal{V}(S_0, \tau^m)$ and $c_r^m = \mathcal{V}(S_{N-1}, \tau^m)$, then the brand new vector $\tilde{\mathcal{V}}^m$ is constructed as follows:

$$\tilde{\mathcal{V}}^m = [\frac{N}{2} \underbrace{c_l^m, \cdots, c_l^m}_N, \underbrace{\mathcal{V}(S_0, \tau^m), \mathcal{V}(S_1, \tau^m), \cdots, \mathcal{V}(S_{N-1}, \tau^m)}_N, \frac{N}{2} \underbrace{c_r^m, \cdots, c_r^m}_N].$$

Take the European options as an example. In the log underlying price domain, the differences between the zero padding and constant padding are presented in Figure (4.1) and Figure (4.2).
Figure 4.1: European call option pricing using zero padding and constant padding

Figure 4.2: European put option pricing using zero padding and constant padding
In practice, it is of interest to focus on the option value when the underlying price tends to be zero rather than extremely large. Since the payoff function of European call option is always zero near the left end, it is almost the same for zero padding and constant padding. However, constant padding is significantly different from zero padding for the European put option.

The parameters for a European put under the one-factor jump diffusion model with the Merton jump density are displayed in Table (4.5).

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>100</td>
</tr>
<tr>
<td>$K$</td>
<td>100</td>
</tr>
<tr>
<td>$r$</td>
<td>0.1</td>
</tr>
<tr>
<td>$q$</td>
<td>0.02</td>
</tr>
<tr>
<td>$T$</td>
<td>1</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>0.2</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>0.1</td>
</tr>
<tr>
<td>$\mu$</td>
<td>-0.5</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Table 4.5: Parameters for wrap-around analysis: a European put option

In this case, the domain of the log underlying prices is defined as $Se^x$ rather than $e^x$ in order to make the grid centered at $S = K = 100$. So, when using the zero padding and constant padding, the original domain $\Omega = [x_{\min}, x_{\max}] = [-7.5, 7.5]$ is expanded into $\tilde{\Omega} = [x'_{\min}, x'_{\max}] = [-15, 15]$. In addition, the total number of nodes in $\Omega$ is $N = 16384$. Therefore, $\Omega$ contains $2N = 32768$ nodes in total.
<table>
<thead>
<tr>
<th>S</th>
<th>No padding</th>
<th>Zero padding</th>
<th>Constant padding</th>
<th>Closed-form solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>87.18014225</td>
<td>87.17857200</td>
<td>90.38700804</td>
<td>90.38572194</td>
</tr>
<tr>
<td>1</td>
<td>89.50246250</td>
<td>89.50246152</td>
<td>89.50354351</td>
<td>89.50354313</td>
</tr>
<tr>
<td>10</td>
<td>80.68175406</td>
<td>80.68175401</td>
<td>80.68175411</td>
<td>80.68175514</td>
</tr>
<tr>
<td>100</td>
<td>5.94853212</td>
<td>5.94853211</td>
<td>5.94853211</td>
<td>5.94851381</td>
</tr>
<tr>
<td>1000</td>
<td>0.00211219</td>
<td>0.00211219</td>
<td>0.00211219</td>
<td>0.00211218</td>
</tr>
<tr>
<td>10000</td>
<td>0.00000024</td>
<td>0.00000023</td>
<td>0.00000023</td>
<td>0.00000023</td>
</tr>
<tr>
<td>100000</td>
<td>0.68766360</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
</tbody>
</table>

Table 4.6: Effect of wrap-around error on the value of a European put option: zero padding and constant padding

As shown in Table (4.6), the option price tends to be incorrect on both ends without padding. Even though zero padding can reduce the effect of wrap-around on the right end, it cannot fix the problem on the left end. Last but not least, constant padding performs best among all three methods for eliminating the wrap-around error for both ends.

Besides zero padding and constant padding, asymptotic padding is another promising choice to extend the domain of log underlying prices and reduce the wrap-around error. Take a European call option as an example. Since the log transformation gives $x = \log \left( \frac{S}{K} \right)$ and $S = Ke^x$, the payoff function of a European call option can be written as:

$$V(x, 0) = V(S, 0)$$
$$= \max(S - K, 0)$$
$$= \max(Ke^x - K, 0)$$
$$= \max(K(e^x - 1), 0).$$

By the Taylor expansions for the exponential function at $x = x_0$:

$$e^x = e^{x_0} + \frac{e^{x_0}}{1!}(x - x_0) + \frac{e^{x_0}}{2!}(x - x_0)^2 + \frac{e^{x_0}}{3!}(x - x_0)^3 + \cdots$$
$$= e^{x_0}\left(1 + \frac{(x - x_0)}{1!} + \frac{(x - x_0)^2}{2!} + \frac{(x - x_0)^3}{3!} + \cdots \right),$$

(4.4)

the asymptotic extension of log underlying prices domain for the right end can be easily derived by taking $x_0 = x_{max}$: $K\left(e^{x_{max}}\left(1 + \frac{(x - x_{max})}{1!} + \frac{(x - x_{max})^2}{2!} + \frac{(x - x_{max})^3}{3!} + \cdots \right) - 1\right)$, where $K$ is the strike price and $x_{max}$ is the right end point of the log prices domain. As for the left end, it is straightforward to simply use zero padding since the value of a European call option is always zero when $S \leq K$.
Furthermore, zero padding can be seen as using zero term from asymptotic padding while constant padding can be viewed as using only first term from asymptotic padding. If the first two terms are used, then linear trend of log underlying prices on the right side will be captured by asymptotic padding. Similarly, if the first three terms are used, then both of linear and quadratic properties can be reflected. Figure (4.3) displays the differences between asymptotic padding using one (constant), two (linear) and three (quadratic) terms.

![Figure 4.3: European call option pricing using asymptotic padding](image)

The parameters for a European call under the one-factor jump diffusion model with the Merton jump density are shown in Table (4.7).
Parameter | Value
---|---
$S$ | $100$
$K$ | $100$
$r$ | $0.05$
$q$ | $0.01$
$T$ | $1$
$\sigma$ | $0.4$
$\lambda$ | $0.2$
$\mu$ | $-0.45$
$\gamma$ | $0.5$

Table 4.7: Parameters for wrap-around analysis: a European call option

In this example, the original domain $\Omega = [x_{\text{min}}, x_{\text{max}}] = [-7.5, 7.5]$ is expanded into $\tilde{\Omega} = [x'_{\text{min}}, x'_{\text{max}}] = [-15, 15]$. The total number of nodes in $\Omega$ is $N = 8192$. Therefore, $\tilde{\Omega}$ contains $2N = 16384$ nodes in total. The results for asymptotic padding can be found in Table (4.8).

<table>
<thead>
<tr>
<th>$S$</th>
<th>Constant padding</th>
<th>Linear padding</th>
<th>Quadratic padding</th>
<th>Closed-form solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.1$</td>
<td>$0.00000043$</td>
<td>$0.00000352$</td>
<td>$0.00001474$</td>
<td>$0.00000000$</td>
</tr>
<tr>
<td>$1$</td>
<td>$0.00000000$</td>
<td>$0.00000000$</td>
<td>$0.00000000$</td>
<td>$0.00000000$</td>
</tr>
<tr>
<td>$10$</td>
<td>$0.0003847$</td>
<td>$0.0003847$</td>
<td>$0.0003847$</td>
<td>$0.0003847$</td>
</tr>
<tr>
<td>$100$</td>
<td>$18.86737568$</td>
<td>$18.86737568$</td>
<td>$18.86737568$</td>
<td>$18.86730167$</td>
</tr>
<tr>
<td>$1000$</td>
<td>$885.10319389$</td>
<td>$885.10319389$</td>
<td>$885.10319389$</td>
<td>$885.10310367$</td>
</tr>
<tr>
<td>$10000$</td>
<td>$9706.86630576$</td>
<td>$9706.86748324$</td>
<td>$9706.86765999$</td>
<td>$9706.86380538$</td>
</tr>
<tr>
<td>$100000$</td>
<td>$95250.91437482$</td>
<td>$97462.61113516$</td>
<td>$97848.56616981$</td>
<td>$97924.74438823$</td>
</tr>
</tbody>
</table>

Table 4.8: Effect of wrap-around error on the value of a European call option: asymptotic padding

It can be seen from Table (4.8) that there exists a tradeoff between the wrap-around error on the left and right ends. The more terms used in asymptotic padding, the computed option values are more accurate on the right end while less accurate on the left end.

Last but not least, the extension of log underlying prices domain is not necessary to be symmetric. Since the Fourier transform assumes a periodic domain implicitly, the extension
of domain in current period may be shifted to next period. For example, when using zero padding, the vector \( V^m \) is extended into a new vector \( \hat{V}^m \):

\[
\hat{V}^m = \left[ 0, \cdots, 0, V(S_0, \tau^m), V(S_1, \tau^m), \cdots, V(S_{N-1}, \tau^m), 0, \cdots, 0 \right].
\]

It is equivalent to shifting all the zeros on the left end to the right end:

\[
\hat{V}^m = [V(S_0, \tau^m), V(S_1, \tau^m), \cdots, V(S_{N-1}, \tau^m), 0, \cdots, 0].
\]

### 4.2 Two-factor Jump Diffusion Cases

As before, under the two-factor jump diffusion model, the options pricing PIDE is given by equation (2.8):

\[
\mathcal{V}_r = \frac{\sigma^2 S_1^2}{2} \mathcal{V}_{S_1 S_1} + \frac{\sigma^2 S_2^2}{2} \mathcal{V}_{S_2 S_2} + (r - \lambda_1 \kappa_1) \mathcal{V}_{S_1} + (r - \lambda_2 \kappa_2) \mathcal{V}_{S_2} + \rho \sigma_1 \sigma_2 S_1 S_2 \mathcal{V}_{S_1 S_2}
\]

\[
- (r + \lambda_1 + \lambda_2) \mathcal{V} + \left( \lambda_1 \int_0^\infty \mathcal{V}(S_1 \eta_1) g(\eta_1) d\eta_1 \right) + \left( \lambda_2 \int_0^\infty \mathcal{V}(S_2 \eta_2) g(\eta_2) d\eta_2 \right).
\]

After log transformation and Fourier transform, the characteristic exponent can be derived as follows:

\[
\Psi(k_1, k_2) = - \frac{\sigma_1^2}{2} (2\pi k_1)^2 - \frac{\sigma_2^2}{2} (2\pi k_2)^2 + (r - \lambda_1 \kappa_1 - \frac{\sigma_1^2}{2})(2\pi i k_1) + (r - \lambda_2 \kappa_2 - \frac{\sigma_2^2}{2})(2\pi i k_2)
\]

\[
+ \rho \sigma_1 \sigma_2 (2\pi k_1)(2\pi k_2) - (r + \lambda_1 + \lambda_2) + \lambda_1 \tilde{F}_1(k_1) + \lambda_2 \tilde{F}_2(k_2).
\]

(4.5)

The independent jump density is given by equations (4.2) and (4.3).

### 4.2.1 Pricing Results

Table (4.9) gives the parameters for a European spread call under the two-factor jump diffusion model with the Merton jump density. The pricing results are presented in Table (4.10).
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1, S_2$</td>
<td>100</td>
</tr>
<tr>
<td>$\sigma_1$</td>
<td>0.1</td>
</tr>
<tr>
<td>$\sigma_2$</td>
<td>0.2</td>
</tr>
<tr>
<td>$\lambda_1$</td>
<td>0.25</td>
</tr>
<tr>
<td>$\lambda_2$</td>
<td>0.5</td>
</tr>
<tr>
<td>$\mu_1$</td>
<td>0.13</td>
</tr>
<tr>
<td>$\mu_2$</td>
<td>0.11</td>
</tr>
<tr>
<td>$\gamma_1$</td>
<td>0.37</td>
</tr>
<tr>
<td>$\gamma_2$</td>
<td>0.41</td>
</tr>
<tr>
<td>$q_1, q_2$</td>
<td>0.05</td>
</tr>
<tr>
<td>$B_1, B_2$</td>
<td>1</td>
</tr>
<tr>
<td>$T$</td>
<td>1</td>
</tr>
<tr>
<td>$K$</td>
<td>2</td>
</tr>
<tr>
<td>$r$</td>
<td>0.1</td>
</tr>
<tr>
<td>$\rho$</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 4.9: Parameters for a European spread call under the two-factor Merton jump diffusion model

<table>
<thead>
<tr>
<th>Refinement</th>
<th>Nodes</th>
<th>Price$^5$</th>
<th>Change</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$512^2$</td>
<td>13.74164280</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$1024^2$</td>
<td>13.72000852</td>
<td>0.02163429</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$2048^2$</td>
<td>13.71519276</td>
<td>0.00481576</td>
<td>4.49239252</td>
</tr>
<tr>
<td>3</td>
<td>$4096^2$</td>
<td>13.71392828</td>
<td>0.00126448</td>
<td>3.80850192</td>
</tr>
<tr>
<td>4</td>
<td>$8192^2$</td>
<td>13.71363235</td>
<td>0.00029593</td>
<td>4.27284272</td>
</tr>
</tbody>
</table>

Table 4.10: Pricing results of a European spread call under the two-factor Merton jump diffusion model: $[x_{\text{min}}, x_{\text{max}}]^2 = [-7.5, 7.5]^2$

As shown in Tables (4.9) and (4.10), the order of convergence is approximately 2 in space for pricing a European spread call under the two-factor Merton jump diffusion model by the FST method.

$^5$Reference price of 13.714948858 is given by Surkov (2009).
4.2.2 Wrap-around Error

From one-dimensional to two-dimensional cases, wrap-around error will also occur when implementing the FST method. Continuously with the idea of constant padding in one-factor case, two-dimensional constant padding can be introduced as follows.

Let $\psi_{i,j}^m := \psi(S_1, S_2, \tau^m)$ denote the option value at node $(S_1, S_2)$, where $i, j = 0, 1, \cdots, N - 1$ at time $\tau^m$. Without padding, the two-dimensional Fourier transform is applied to the $N \times N$ matrix $\psi^m$:

$$\psi^m = \begin{bmatrix} \psi_{0,0}^m & \cdots & \psi_{N-1,0}^m \\ \vdots & \ddots & \vdots \\ \psi_{0,N-1}^m & \cdots & \psi_{N-1,N-1}^m \end{bmatrix}.$$ 

In order to use constant padding in the two-dimensional case, the original matrix $\psi^m$ can be expanded into a new $2N \times 2N$ matrix $\tilde{\psi}^m$:

$$\tilde{\psi}^m = \begin{bmatrix} \mathcal{M}_1^m & \mathcal{M}_2^m & \mathcal{M}_3^m \\ \mathcal{M}_4^m & \psi^m & \mathcal{M}_5^m \\ \mathcal{M}_6^m & \mathcal{M}_7^m & \mathcal{M}_8^m \end{bmatrix}$$

where $\mathcal{M}_1^m, \mathcal{M}_3^m, \mathcal{M}_6^m$ and $\mathcal{M}_8^m$ are all $\frac{N}{2} \times \frac{N}{2}$ constant matrices:

$$\mathcal{M}_1^m = \begin{bmatrix} \psi_{0,0}^m & \cdots & \psi_{0,0}^m \\ \vdots & \ddots & \vdots \\ \psi_{0,0}^m & \cdots & \psi_{0,0}^m \end{bmatrix}, \quad \mathcal{M}_3^m = \begin{bmatrix} \psi_{N-1,0}^m & \cdots & \psi_{N-1,0}^m \\ \vdots & \ddots & \vdots \\ \psi_{N-1,0}^m & \cdots & \psi_{N-1,0}^m \end{bmatrix},$$

$$\mathcal{M}_6^m = \begin{bmatrix} \psi_{0,N-1}^m & \cdots & \psi_{0,N-1}^m \\ \vdots & \ddots & \vdots \\ \psi_{0,N-1}^m & \cdots & \psi_{0,N-1}^m \end{bmatrix}, \quad \mathcal{M}_8^m = \begin{bmatrix} \psi_{N-1,N-1}^m & \cdots & \psi_{N-1,N-1}^m \\ \vdots & \ddots & \vdots \\ \psi_{N-1,N-1}^m & \cdots & \psi_{N-1,N-1}^m \end{bmatrix}.$$
$\mathcal{M}_4^m$ and $\mathcal{M}_5^m$ are $N \times \frac{N}{2}$ matrices with same values for each column:

$$\mathcal{M}_4^m = \begin{bmatrix} V_{0,0}^m & \cdots & V_{0,0}^m \\ \vdots & \ddots & \vdots \\ V_{0,N-1}^m & \cdots & V_{0,N-1}^m \end{bmatrix}, \mathcal{M}_5^m = \begin{bmatrix} V_{N-1,0}^m & \cdots & V_{N-1,0}^m \\ \vdots & \ddots & \vdots \\ V_{N-1,N-1}^m & \cdots & V_{N-1,N-1}^m \end{bmatrix}.$$

Intuitively, the idea for constant padding can be described by Figure (4.4):

![Intuitive idea for constant padding in two dimensions](image)

Figure 4.4: Intuitive idea for constant padding in two dimensions

Using the European spread call example shown above, the parameters are listed in Table (4.9). The original domain $\Omega = [x_{\min}, x_{\max}]^2 = [-7.5, 7.5]^2$ is expanded into $\tilde{\Omega} = [x'_{\min}, x'_{\max}]^2 = [-15, 15]^2$. In addition, the total number of nodes in $\Omega$ is $N^2 = 1024^2$. Therefore, $\tilde{\Omega}$ contains $(2N)^2 = 2048^2$ nodes in total.

The effect of the wrap-around error and the reduction in wrap-around error by using constant padding on the European spread call are displayed in Table (4.11) and Table (4.12).
### Table 4.11: Effect of wrap-around error on the value of a European spread call option: *no padding*

<table>
<thead>
<tr>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$x$</th>
<th>0.1</th>
<th>1</th>
<th>10</th>
<th>100</th>
<th>1000</th>
<th>10000</th>
<th>100000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>-6.9078</td>
<td>5003.73181862</td>
<td>0.05921847</td>
<td>7.3490359</td>
<td>91.2059832</td>
<td>928.82572488</td>
<td>9303.91456738</td>
<td>80823.40168517</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-4.6052</td>
<td>5128.71781581</td>
<td>0.04508410</td>
<td>6.75280164</td>
<td>92.3655481</td>
<td>948.49374193</td>
<td>9508.64636900</td>
<td>82598.37083190</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>-2.3026</td>
<td>5128.40582923</td>
<td>0.03699906</td>
<td>0.93411200</td>
<td>83.80336239</td>
<td>939.93259791</td>
<td>9500.08689466</td>
<td>82590.27332970</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>5125.27577522</td>
<td>0.03686791</td>
<td>0.00220805</td>
<td>13.72001200</td>
<td>854.32012750</td>
<td>9141.47742373</td>
<td>82509.49357610</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>2.3026</td>
<td>5093.97524082</td>
<td>0.03665502</td>
<td>0.00000079</td>
<td>0.02411654</td>
<td>143.05136774</td>
<td>8558.35440025</td>
<td>81701.70615469</td>
<td></td>
</tr>
<tr>
<td>10000</td>
<td>4.6052</td>
<td>4780.97012512</td>
<td>0.03452676</td>
<td>0.00000005</td>
<td>0.00000000</td>
<td>1435.34375033</td>
<td>82598.33708319</td>
<td>73623.88083482</td>
<td></td>
</tr>
<tr>
<td>100000</td>
<td>6.9078</td>
<td>1763.14404765</td>
<td>0.01408759</td>
<td>0.00000151</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>98.37997582</td>
<td>8032.19200756</td>
<td></td>
</tr>
</tbody>
</table>

### Table 4.12: Effect of wrap-around error on the value of a European spread call option: *constant padding*

<table>
<thead>
<tr>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$x$</th>
<th>0.1</th>
<th>1</th>
<th>10</th>
<th>100</th>
<th>1000</th>
<th>10000</th>
<th>100000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>-6.9078</td>
<td>0.000000151</td>
<td>0.02374630</td>
<td>7.6076427</td>
<td>93.22048326</td>
<td>949.34861339</td>
<td>9510.3290373</td>
<td>91519.18921942</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-4.6052</td>
<td>0.000000040</td>
<td>0.00815625</td>
<td>6.7529563</td>
<td>92.36457173</td>
<td>948.49270146</td>
<td>9509.47319481</td>
<td>81518.33307490</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>-2.3026</td>
<td>0.000000000</td>
<td>0.00010992</td>
<td>0.9339475</td>
<td>83.80337167</td>
<td>939.93141440</td>
<td>9500.9124075</td>
<td>81509.77202444</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>0.000000000</td>
<td>0.000100007</td>
<td>0.00220956</td>
<td>13.72000852</td>
<td>854.31899368</td>
<td>9415.2944524</td>
<td>81424.15069093</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>2.3026</td>
<td>0.000000000</td>
<td>0.00000007</td>
<td>0.00000075</td>
<td>0.00011624</td>
<td>143.04821613</td>
<td>8559.1753924</td>
<td>90568.03092627</td>
<td></td>
</tr>
<tr>
<td>10000</td>
<td>4.6052</td>
<td>0.000000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>0.00000759</td>
<td>1435.34375033</td>
<td>82598.33708319</td>
<td>82006.83260731</td>
<td></td>
</tr>
<tr>
<td>100000</td>
<td>6.9078</td>
<td>0.000000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>98.37997582</td>
<td>8032.19200756</td>
<td></td>
</tr>
</tbody>
</table>

As shown in Table (4.11) and Table (4.12), the option values on the edges of the matrix seem to be incorrect without padding. After using constant padding, however, these values tend to be reasonable.
4.3 Shared-jump Diffusion Cases

As mentioned before in Section 3.4, the characteristic exponent under the shared-jump diffusion model is given by equation (3.23):

\[
\Psi(k_1, k_2) = - \frac{1}{2} \sigma_1^2 (2\pi k_1)^2 - \frac{1}{2} \sigma_2^2 (2\pi k_2)^2 \\
+ (r - \lambda \kappa - \frac{1}{2} \sigma_1^2)(2\pi i k_1) + (r - \lambda \kappa - \frac{1}{2} \sigma_2^2)(2\pi i k_2) \\
- \rho \sigma_1 \sigma_2 (2\pi k_1)(2\pi k_2) - (r + \lambda) + \lambda \mathcal{F}[f(x)](k_1 + k_2).
\]

In addition, the jump density is given by equation (4.3).

4.3.1 Pricing Results

The parameters for a European spread call under the shared-jump diffusion model with Kou jump density are shown in Table (4.13) and the pricing results are presented in Table (4.14).

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1, S_2$</td>
<td>100</td>
</tr>
<tr>
<td>$\sigma_1$</td>
<td>0.1</td>
</tr>
<tr>
<td>$\sigma_2$</td>
<td>0.2</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>1</td>
</tr>
<tr>
<td>$p$</td>
<td>0.4</td>
</tr>
<tr>
<td>$\eta_1$</td>
<td>3</td>
</tr>
<tr>
<td>$\eta_2$</td>
<td>2</td>
</tr>
<tr>
<td>$B_1, B_2$</td>
<td>1</td>
</tr>
<tr>
<td>$T$</td>
<td>1</td>
</tr>
<tr>
<td>$K$</td>
<td>2</td>
</tr>
<tr>
<td>$r$</td>
<td>0.1</td>
</tr>
<tr>
<td>$\rho$</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 4.13: Parameters for a European spread call under the shared-jump diffusion model
Table 4.14: Pricing results of a European spread call under the shared-jump diffusion model: \textit{FST method}, \([x_{\min}, x_{\max}]^2 = [-7.5, 7.5]^2\)

<table>
<thead>
<tr>
<th>Refinement</th>
<th>Nodes</th>
<th>Price</th>
<th>Change</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>512\textsuperscript{2}</td>
<td>6.16739294</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1024\textsuperscript{2}</td>
<td>6.12648119</td>
<td>0.04091174</td>
<td>4.489778638</td>
</tr>
<tr>
<td>2</td>
<td>2048\textsuperscript{2}</td>
<td>6.11736900</td>
<td>0.00911220</td>
<td>3.833485132</td>
</tr>
<tr>
<td>3</td>
<td>4096\textsuperscript{2}</td>
<td>6.11499200</td>
<td>0.00237700</td>
<td>3.833485132</td>
</tr>
<tr>
<td>4</td>
<td>8192\textsuperscript{2}</td>
<td>6.11441482</td>
<td>0.00057717</td>
<td>4.118356562</td>
</tr>
</tbody>
</table>

Tables (4.13) and (4.14) demonstrate that second order convergence is attained for pricing a European spread call under the shared-jump diffusion model by the FST method.

### 4.3.2 Monte Carlo Results

In order to check the results by FST method, Monte Carlo simulation is used to price the same European spread call under the shared-jump diffusion model.

The Monte Carlo approach for option pricing was firstly introduced by Boyle (1977)[4]. Monte Carlo method simulates the process generating the returns on the underlying asset and invokes the risk neutrality assumption to derive the value of the option.

Note that using forward Euler to approximate the SDEs has \(O(\Delta t)\) truncation error while Monte Carlo simulation has \(O(\frac{1}{\sqrt{M}})\) error, where \(\Delta t\) is the timestep and \(M\) is the total number of Monte Carlo paths. Thus, there are two sources of error in Monte Carlo approach: time-stepping error and sampling error:

\[
\text{error} = O\left(\max(\Delta t, \frac{1}{\sqrt{M}})\right)
\]  

Equation (4.6) shows that it is crucial to balance time-stepping error and sampling error when using Monte Carlo. In order to make these two errors the same order, \(M\) should be chosen as \(O\left(\frac{1}{(\Delta t)^2}\right)\).

The details about the methodology of Monte Carlo simulation under the shared-jump diffusion model can be found in (Appendix C). The results derived by Monte Carlo simulations are provided in Table (4.15).
<table>
<thead>
<tr>
<th>Simulations</th>
<th>Timesteps</th>
<th>Price</th>
<th>Stdev</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>100000</td>
<td>1600</td>
<td>6.10757277</td>
<td>0.04689684</td>
<td>[6.0157, 6.1995]</td>
</tr>
<tr>
<td>400000</td>
<td>3200</td>
<td>6.11528466</td>
<td>0.02473321</td>
<td>[6.0668, 6.1638]</td>
</tr>
<tr>
<td>1600000</td>
<td>6400</td>
<td>6.12621791</td>
<td>0.01166751</td>
<td>[6.1033, 6.1491]</td>
</tr>
<tr>
<td>6400000</td>
<td>128000</td>
<td>6.11290968</td>
<td>0.00583618</td>
<td>[6.1015, 6.1243]</td>
</tr>
<tr>
<td>25600000</td>
<td>25600000</td>
<td>6.11145233</td>
<td>0.00298162</td>
<td>[6.1056, 6.1173]</td>
</tr>
</tbody>
</table>

Table 4.15: Pricing results of a European spread call under the shared-jump diffusion model: Monte Carlo simulation

By comparing the results in Table (4.14) and Table (4.15), the FST method (6.11441482) is consistent with Monte Carlo simulations.

### 4.3.3 Wrap-around Error

When using constant padding to reduce the wrap-around error under the shared-jump diffusion model, another issue is about jump density grid. The equation (3.23) gives the characteristic exponent:

\[
\Psi(k_1, k_2) = -\frac{1}{2}\sigma_1^2(2\pi k_1)^2 - \frac{1}{2}\sigma_2^2(2\pi k_2)^2 \\
+ (r - \lambda \kappa - \frac{1}{2}\sigma_1^2)(2\pi ik_1) + (r - \lambda \kappa - \frac{1}{2}\sigma_2^2)(2\pi ik_2) \\
- \rho \sigma_1 \sigma_2 (2\pi k_1)(2\pi k_2) - (r + \lambda) + \lambda \mathcal{F}[f(x)](k_1 + k_2).
\]

In the frequency domain, \(k_1, k_2 \in \left\{ \frac{-N+1}{x_{\text{max}}-x_{\text{min}}}, \ldots, \frac{N}{x_{\text{max}}-x_{\text{min}}} \right\} \) by equations (3.10) and (3.11). Since the Fourier transform of jump density is evaluated at \((k_1 + k_2)\), the range of its grid is doubled naturally when evaluating \( \mathcal{F}[f(x)](k_1 + k_2) \) under the shared-jump diffusion model.

Especially, when using the constant padding, suppose the log underlying domain \(\Omega = [x_{\text{min}}, x_{\text{max}}]^2\) is expanded into \(\bar{\Omega} = [x'_{\text{min}}, x'_{\text{max}}]^2 = [2x_{\text{min}}, 2x_{\text{max}}]^2\), the grid for the Fourier transform of jump density has to be quadrupled in order to evaluate \( \mathcal{F}[f(x)](k_1 + k_2) \). Once \(\Psi(k_1, k_2)\) is computed, however, only a doubled domain is required.

\[6\] The 95% confidence interval (CI) is constructed as \([\bar{x} - 1.96\frac{\sigma}{\sqrt{n}}, \bar{x} + 1.96\frac{\sigma}{\sqrt{n}}]\), where \(\bar{x}\) denotes the sample mean, \(\sigma\) denotes the sample standard deviation and \(n\) represents the sample size.
Take the European spread put as an example, with the parameters listed in Table (4.16). In addition, the original domain $\Omega = [x_{\min}, x_{\max}]^2 = [-15, 15]^2$ is expanded into $\tilde{\Omega} = [x'_{\min}, x'_{\max}]^2 = [2x_{\min}, 2x_{\max}]^2 = [-30, 30]^2$. Also, the total number of nodes in $\Omega$ is $N^2 = 4096^2$. Therefore, $\tilde{\Omega}$ contains $(2N)^2 = 8192^2$ nodes in total.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$, $S_2$</td>
<td>100</td>
</tr>
<tr>
<td>$\sigma_1$</td>
<td>0.15</td>
</tr>
<tr>
<td>$\sigma_2$</td>
<td>0.2</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>0.1</td>
</tr>
<tr>
<td>$p$</td>
<td>0.3</td>
</tr>
<tr>
<td>$\eta_1$</td>
<td>2.0</td>
</tr>
<tr>
<td>$\eta_2$</td>
<td>1.5</td>
</tr>
<tr>
<td>$B_1$, $B_2$</td>
<td>1</td>
</tr>
<tr>
<td>$T$</td>
<td>1</td>
</tr>
<tr>
<td>$K$</td>
<td>2</td>
</tr>
<tr>
<td>$r$</td>
<td>0.15</td>
</tr>
<tr>
<td>$\rho$</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Table 4.16: Parameters for wrap-around analysis: a European spread put option

The effect of the wrap-around error and the reduction in wrap-around error by using constant padding on a European spread put under the shared-jump model are displayed in Table (4.17) and Table (4.18).
Table 4.17: Effect of wrap-around error on the value of a European spread put option: no padding

<table>
<thead>
<tr>
<th>$S_1$</th>
<th>$S_2$</th>
<th>0.1</th>
<th>1</th>
<th>10</th>
<th>100</th>
<th>1000</th>
<th>10000</th>
<th>100000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>-6.9078</td>
<td>17.47053985</td>
<td>75.23962310</td>
<td>74.69524146</td>
<td>74.69237973</td>
<td>74.69224819</td>
<td>74.69223570</td>
</tr>
<tr>
<td>0.1</td>
<td>-6.9078</td>
<td>4.97531710</td>
<td>2.27490574</td>
<td>2.61650451</td>
<td>2.62906396</td>
<td>2.62196232</td>
<td>2.62196013</td>
<td>2.62198299</td>
</tr>
<tr>
<td>1</td>
<td>-4.6052</td>
<td>11.71229797</td>
<td>10.80384886</td>
<td>1.95910322</td>
<td>0.09157702</td>
<td>0.09177131</td>
<td>0.09179222</td>
<td>0.09202230</td>
</tr>
<tr>
<td>10</td>
<td>-2.3026</td>
<td>101.62461135</td>
<td>100.72457649</td>
<td>91.72431900</td>
<td>8.43379086</td>
<td>0.00322242</td>
<td>0.00347426</td>
<td>0.00577647</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>1000.62461135</td>
<td>1000.72457649</td>
<td>991.72431900</td>
<td>9901.75775362</td>
<td>9901.75855724</td>
<td>9901.75845603</td>
<td>9901.75485603</td>
</tr>
<tr>
<td>1000</td>
<td>4.6052</td>
<td>10000.62461135</td>
<td>10000.72457649</td>
<td>9991.72431900</td>
<td>9991.75775362</td>
<td>9991.75855724</td>
<td>9991.75845603</td>
<td>9991.75485603</td>
</tr>
<tr>
<td>10000</td>
<td>6.9078</td>
<td>99999.62461135</td>
<td>99999.72457649</td>
<td>99999.72431900</td>
<td>99999.75775362</td>
<td>99999.75855724</td>
<td>99999.75845603</td>
<td>99999.75485603</td>
</tr>
</tbody>
</table>

Table 4.18: Effect of wrap-around error on the value of a European spread put option: constant padding

<table>
<thead>
<tr>
<th>$S_1$</th>
<th>$S_2$</th>
<th>0.1</th>
<th>1</th>
<th>10</th>
<th>100</th>
<th>1000</th>
<th>10000</th>
<th>100000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>-6.9078</td>
<td>1.72141966</td>
<td>0.83609664</td>
<td>0.00408077</td>
<td>0.00014001</td>
<td>0.00000494</td>
<td>0.00000018</td>
</tr>
<tr>
<td>0.1</td>
<td>-6.9078</td>
<td>2.62142170</td>
<td>1.72176427</td>
<td>1.00465273</td>
<td>0.00014199</td>
<td>0.00000494</td>
<td>0.00000018</td>
<td>0.00000000</td>
</tr>
<tr>
<td>1</td>
<td>-4.6052</td>
<td>11.62147615</td>
<td>10.72147038</td>
<td>1.93974960</td>
<td>0.00016483</td>
<td>0.00000501</td>
<td>0.00000017</td>
<td>-0.00000001</td>
</tr>
<tr>
<td>10</td>
<td>-2.3026</td>
<td>101.62141429</td>
<td>100.72140853</td>
<td>91.72135410</td>
<td>8.43311492</td>
<td>0.00000580</td>
<td>0.00000018</td>
<td>0.00000001</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>1000.62141429</td>
<td>1000.72140853</td>
<td>991.72135410</td>
<td>9901.75775362</td>
<td>9901.75855724</td>
<td>9901.75845603</td>
<td>9901.75485603</td>
</tr>
<tr>
<td>1000</td>
<td>4.6052</td>
<td>10000.62141429</td>
<td>10000.72140853</td>
<td>9991.72135410</td>
<td>9991.75775362</td>
<td>9991.75855724</td>
<td>9991.75845603</td>
<td>9991.75485603</td>
</tr>
<tr>
<td>10000</td>
<td>6.9078</td>
<td>100000.62141429</td>
<td>100000.72140853</td>
<td>99999.72135410</td>
<td>99999.75775362</td>
<td>99999.75855724</td>
<td>99999.75845603</td>
<td>99999.75485603</td>
</tr>
</tbody>
</table>

Table (4.17) and Table (4.18) demonstrate that the option values on the edges of the matrix become reasonable after using the constant padding.
It is known that two key factors have influence on the accuracy of constant padding. One is the total number of nodes while the other is the length of the domain. In order to have a better understanding of the accuracy of constant padding, the convergence of the option value at \((S_1, S_2) = (100, 100)\) under two different cases are shown in Table (4.19) and Table (4.20).

On one hand, the domain \(\Omega = [x_{\text{min}}, x_{\text{max}}]^2 = [-15, 15]^2\) remains unchanged and the total number of nodes \(N^2\) is chosen as: 512\(^2\), 1024\(^2\), 2048\(^2\), 4096\(^2\) and 8192\(^2\) for each case. So, \(\Delta x = \frac{x_{\text{max}} - x_{\text{min}}}{N}\) tends to be smaller as \(N\) becomes larger. The convergence of the option value at \((S_1, S_2) = (100, 100)\) is displayed in Table (4.19).

<table>
<thead>
<tr>
<th>Nodes (N^2)</th>
<th>Price</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>512(^2)</td>
<td>8.44537854</td>
<td></td>
</tr>
<tr>
<td>1024(^2)</td>
<td>8.43882006</td>
<td>0.00655847</td>
</tr>
<tr>
<td>2048(^2)</td>
<td>8.43248399</td>
<td>0.00633607</td>
</tr>
<tr>
<td>4096(^2)</td>
<td>8.43311492</td>
<td>-0.00063093</td>
</tr>
<tr>
<td>8192(^2)</td>
<td>8.43310941</td>
<td>0.00000551</td>
</tr>
</tbody>
</table>

Table 4.19: Convergence table of a European spread put under the shared-jump diffusion model using constant padding: \(\text{constant } \Omega, \text{ various } \Delta x\)

On the other hand, \(\Delta x = 0.0073\) remains constant and the domain \(\Omega = [x_{\text{min}}, x_{\text{max}}]^2\) is selected to be: \([-11, 11]^2\), \([-12, 12]^2\), \([-13, 13]^2\), \([-14, 14]^2\) and \([-15, 15]^2\) for each case. The convergence of the option value at \((S_1, S_2) = (100, 100)\) is displayed in Table (4.20).

<table>
<thead>
<tr>
<th>Domain ([-x_{\text{min}}, x_{\text{max}}]^2)</th>
<th>Price</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>([-11, 11]^2]</td>
<td>8.43310421</td>
<td>-0.00000655</td>
</tr>
<tr>
<td>([-12, 12]^2]</td>
<td>8.43311077</td>
<td>-0.00000233</td>
</tr>
<tr>
<td>([-13, 13]^2]</td>
<td>8.43311310</td>
<td>-0.00000171</td>
</tr>
<tr>
<td>([-14, 14]^2]</td>
<td>8.43311480</td>
<td>-0.00000012</td>
</tr>
<tr>
<td>([-15, 15]^2]</td>
<td>8.43311492</td>
<td>-0.00000012</td>
</tr>
</tbody>
</table>

Table 4.20: Convergence table of a European spread put under the shared-jump diffusion model using constant padding: \(\text{constant } \Delta x, \text{ various } \Omega\)
Chapter 5

Empirical Data Analysis

5.1 Data Exploration

In order to justify the validity of the shared-jump diffusion model, the daily and monthly total return data of the following three main financial markets’ index, S&P 500\textsuperscript{1}, Eurostox\textsuperscript{2} and FTSE\textsuperscript{3}, were extracted from Thomson-Reuters Eikon\textsuperscript{4}. All returns are from December 31, 1991 to December 31, 2017, including dividends and distributions.

In Figure (5.1), the same trend of prices can be observed for S&P 500, Eurostox and FTSE, especially when dramatic fluctuations happened. All the prices are rescaled to start from 1000 in USD, dating from December 31, 1991 to December 31, 2017.

From the plot of log returns Figure (5.2), S&P 500, Eurostox and FTSE tend to share almost the same fluctuations at same time, which reflects the contagion effects on global financial markets.

Figures (5.3), (5.4) and (5.5) compare the daily and monthly observed densities of log returns of S&P 500, Eurostox and FTSE with the standard normal density. Note that all the log returns are rescaled to zero mean, unit standard deviation. It is clear to see that there exist some improbable log returns compared with standard normal distribution. Hence, these cases can be determined as jumps rather than Brownian motion changes.

\begin{enumerate}
\item S&P 500 is an American stock market index based on the market capitalizations of 500 large companies having common stock listed on the NYSE or NASDAQ.
\item Eurostox is a stock index of fifty largest and most liquid Eurozone stocks.
\item FTSE is a share index of the 100 companies listed on the London Stock Exchange with the highest market capitalization.
\item Thomson-Reuters Eikon is a set of financial analysis tools. \url{https://eikon.thomsonreuters.com/}
\end{enumerate}
Figure 5.1: Prices of S&P 500, Eurostox and FTSE

Figure 5.2: Log returns of S&P 500, Eurostox and FTSE
Figure 5.3: Scaled observed density and standard normal density of S&P 500 log returns. 
*Left: daily, Right: monthly*

Figure 5.4: Scaled observed density and standard normal density of Eurostox log returns. 
*Left: daily, Right: monthly*
Figure 5.5: Scaled observed density and standard normal density of FTSE log returns. 
*Left: daily, Right: monthly*

Furthermore, the descriptive statistics for daily and monthly log returns of S&P 500, Eurostox and FTSE are displayed in Tables (5.1) and (5.2). The skewness tells that log returns follow the asymmetric distribution and kurtosis shows that higher frequency series tend to behave less like a normal distribution while lower frequency series behave more like a normal distribution.

<table>
<thead>
<tr>
<th>Statistics</th>
<th>S&amp;P 500</th>
<th>Eurostox</th>
<th>FTSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>-0.0003</td>
<td>-0.0001</td>
<td>-0.0001</td>
</tr>
<tr>
<td>Median</td>
<td>-0.0005</td>
<td>-0.0004</td>
<td>-0.0004</td>
</tr>
<tr>
<td>Stdev</td>
<td>0.0111</td>
<td>0.0148</td>
<td>0.0126</td>
</tr>
<tr>
<td>Skewness</td>
<td>0.2498</td>
<td>0.1397</td>
<td>0.2821</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>9.4865</td>
<td>6.1047</td>
<td>8.2900</td>
</tr>
</tbody>
</table>

Table 5.1: Descriptive statistics for daily log returns of S&P 500, Eurostox and FTSE
### Table 5.2: Descriptive statistics for monthly log returns of S&P 500, Eurostox and FTSE

<table>
<thead>
<tr>
<th>Statistics</th>
<th>S&amp;P 500</th>
<th>Eurostox</th>
<th>FTSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>-0.0060</td>
<td>-0.0037</td>
<td>-0.0026</td>
</tr>
<tr>
<td>Median</td>
<td>-0.0104</td>
<td>-0.0074</td>
<td>-0.0043</td>
</tr>
<tr>
<td>Stdev</td>
<td>0.0407</td>
<td>0.0593</td>
<td>0.0457</td>
</tr>
<tr>
<td>Skewness</td>
<td>0.8896</td>
<td>0.7558</td>
<td>0.5832</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>2.1198</td>
<td>1.7267</td>
<td>1.8225</td>
</tr>
</tbody>
</table>

#### 5.2 Empirical Estimates

With the intention of estimating appropriate parameters for the jump diffusion model, the daily and monthly total return series of the three indices from December 31, 1991 to December 31, 2017 are used. The methodology of parameter estimation is systematically mentioned in Dang and Forsyth (2016)[6] and Forsyth and Vetzal (2017)[7].

Consider discrete series of index prices $S(t_i) = S_i, i = 1, 2, \cdots, N + 1$, with equal time intervals $\Delta t = t_{i+1} - t_i, \forall i$ and $T = N\Delta t$. Define the log returns $\Delta X_i$ as:

$$\Delta X_i = \log \left( \frac{S_{i+1}}{S_i} \right)$$  \hspace{1cm} (5.1)

and the detrended log returns $\Delta \hat{X}_i$ as:

$$\Delta \hat{X}_i = \Delta X_i - \hat{m} \Delta t$$

$$\hat{m} = \frac{\log(S_{N+1}) - \log(S_1)}{T}.$$  \hspace{1cm} (5.2)

The important feature of a jump diffusion model is that it allows modelling of infrequent large jumps in underlying prices. The thresholding technique described by Mancini (2009)[13] is used to filter out the infrequent large jumps.

Suppose the estimate for the diffusive volatility component $\hat{\sigma}$ is given, then a jump can be detected by Shimizu (2013)[16] if

$$|\Delta \hat{X}_i| > \alpha \hat{\sigma} \frac{\sqrt{\Delta t}}{(\Delta t)^{\beta}}$$  \hspace{1cm} (5.3)

where $\alpha$ and $\beta$ are tuning parameters. The intuition behind equation (5.3) can be easily explained. For example, if $\alpha = 3$ and $\beta \ll 1$, then a return will be viewed as a jump if it is
larger than a 3 standard deviation Brownian motion change, which would be improbable. Therefore, it should be considered as a jump.

Consequently, the jump detection indicators $1_{i}^{up}$ and $1_{i}^{dwn}$ are defined as follows:

$$1_{i}^{up} = \begin{cases} 1, & \text{if } \Delta \hat{X}_i > \alpha \hat{\sigma} \sqrt{\Delta t} \\ 0, & \text{otherwise} \end{cases}$$

(5.4)

and

$$1_{i}^{dwn} = \begin{cases} 1, & \text{if } \Delta \hat{X}_i < -\alpha \hat{\sigma} \sqrt{\Delta t} \\ 0, & \text{otherwise} \end{cases}.$$  

(5.5)

Criteria (5.4) and (5.5) allows one to separate downward from upward jumps.

Define

$$\sum_{i=1}^{N} 1_{i}^{up} = N^{up}, \sum_{i=1}^{N} 1_{i}^{dwn} = N^{dwn}, N^{jmps} = N^{up} + N^{dwn}; \sum_{i=1}^{N} \left(1 - 1_{i}^{up} - 1_{i}^{dwn}\right) = N^{gbm}$$

where $N^{jmps}$ denotes the total number of jumps detected and $N^{gbm}$ denotes the number of geometric Brownian motion increments. Then the estimate of the diffusive volatility is:

$$\hat{\sigma}^2 = \frac{1}{\Delta t} \text{var}\left(\{\Delta \hat{X}_i \mid (1_{i}^{up} + 1_{i}^{dwn} = 0)\}\right).$$

(5.6)

Note that equations (5.4), (5.5) and (5.6) constitute an implicit equation for $\hat{\sigma}^2$, which must be solved by an iterative method by Clewlow and Strickland (2002)[5].

Given the estimate of the diffusive volatility $\hat{\sigma}^2$, other jump parameters can be estimated by:

$$\lambda = \frac{N^{jmps}}{T}; \quad p = \frac{N^{up}}{N^{jmps}}; \quad \eta_1 = \frac{1}{\text{mean}\left(\{\Delta \hat{X}_i \mid (1_{i}^{up} = 1)\}\right)}; \quad \eta_2 = \frac{-1}{\text{mean}\left(\{\Delta \hat{X}_i \mid (1_{i}^{dwn} = 1)\}\right)}.$$  

(5.7)

Once estimates for $\sigma, \lambda, p, \eta_1, \eta_2$ are fixed, the drift term is easy to be estimated as follows. From equation (2.1), let $X = \log(S)$:

$$dX = \left(\mu - \lambda \kappa - \frac{\sigma^2}{2}\right)dt + \sigma dZ + \left(\log(\eta)\right) dq.$$  

(5.8)
Taking expectations of both sides of (5.8) and assuming that only one jump takes place in \([t, t + dt]\) gives:

\[
E[dX] = \left(\mu - \lambda \kappa - \frac{\sigma^2}{2}\right)dt + \lambda E[\log(\eta)]dt. \tag{5.9}
\]

Recall \(\kappa = E[\eta - 1]\). Writing equation (5.9) in discrete time leads to:

\[
\frac{\text{mean}(\Delta X_i)}{\Delta t} = \left(\mu - \lambda \kappa - \frac{\sigma^2}{2}\right) + \lambda \left(\frac{p \eta_1}{\eta_2} - (1 - p)\right). \tag{5.10}
\]

Therefore, given \(\sigma, \lambda, p, \eta_1, \eta_2\), the drift term \(\mu\) can be solved by equation (5.10).

The parameters estimated from real data are listed in Tables (5.3) and (5.4):

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\alpha)</td>
<td>5</td>
</tr>
<tr>
<td>(T)</td>
<td>26</td>
</tr>
<tr>
<td>(N)</td>
<td>6776</td>
</tr>
<tr>
<td>(\Delta t)</td>
<td>0.003837072</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
<th>S&amp;P 500</th>
<th>Eurostoxx</th>
<th>FTSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\hat{m})</td>
<td>0.071457034</td>
<td>0.04406454</td>
<td>0.030916764</td>
</tr>
<tr>
<td>(\hat{\sigma})</td>
<td>0.165005933</td>
<td>0.225618585</td>
<td>0.189453899</td>
</tr>
<tr>
<td>(N^{\text{jmps}})</td>
<td>25</td>
<td>20</td>
<td>19</td>
</tr>
<tr>
<td>(N^{\text{up}})</td>
<td>10</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>0.961538462</td>
<td>0.769230769</td>
<td>0.730769231</td>
</tr>
<tr>
<td>(p)</td>
<td>0.4</td>
<td>0.5</td>
<td>0.315789474</td>
</tr>
<tr>
<td>(\eta_1)</td>
<td>14.46086149</td>
<td>11.01653107</td>
<td>11.07519672</td>
</tr>
<tr>
<td>(\eta_2)</td>
<td>14.3162148</td>
<td>11.59146755</td>
<td>12.45671547</td>
</tr>
<tr>
<td>(\kappa)</td>
<td>-0.09945839</td>
<td>0.010208051</td>
<td>-0.019502028</td>
</tr>
<tr>
<td>(\mu)</td>
<td>0.089985375</td>
<td>0.055513894</td>
<td>0.043757312</td>
</tr>
</tbody>
</table>

Table 5.3: Parameters estimated from daily data

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\alpha)</td>
<td>3</td>
</tr>
<tr>
<td>(T)</td>
<td>26</td>
</tr>
<tr>
<td>(N)</td>
<td>312</td>
</tr>
<tr>
<td>(\Delta t)</td>
<td>0.083333333</td>
</tr>
</tbody>
</table>
### Table 5.4: Parameters estimated from monthly data

<table>
<thead>
<tr>
<th>Parameters</th>
<th>S&amp;P 500</th>
<th>Eurostox</th>
<th>FTSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{m}$</td>
<td>0.072230128</td>
<td>0.044368741</td>
<td>0.031416643</td>
</tr>
<tr>
<td>$\hat{\sigma}$</td>
<td>0.128230026</td>
<td>0.192026453</td>
<td>0.145231375</td>
</tr>
<tr>
<td>$N_{jumps}$</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>$N_{up}$</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>0.153846154</td>
<td>0.153846154</td>
<td>0.153846154</td>
</tr>
<tr>
<td>$p$</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
</tr>
<tr>
<td>$\eta_1$</td>
<td>-</td>
<td>-</td>
<td>7.822503629</td>
</tr>
<tr>
<td>$\eta_2$</td>
<td>6.663146548</td>
<td>4.843224528</td>
<td>5.840098476</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>-</td>
<td>-</td>
<td>-0.073004105</td>
</tr>
<tr>
<td>$\mu$</td>
<td>-</td>
<td>-</td>
<td>0.045071965</td>
</tr>
</tbody>
</table>

Note that for monthly data, there are no upward jumps detected by choosing $\alpha = 3$ for both S&P 500 and Eurostox. For details, the specific dates of jumps are shown in Table (5.5):

<table>
<thead>
<tr>
<th>indices</th>
<th>dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eurostox</td>
<td>2002-07 ↓ 2002-09 ↓ 2008-10 ↓ 2009-01 ↓</td>
</tr>
<tr>
<td>FTSE</td>
<td>2008-09 ↓ 2008-10 ↓ 2009-05 ↑ 2012-05 ↓</td>
</tr>
</tbody>
</table>

Table 5.5: Specific dates of jumps detected from monthly data

In the period of October 2008, which is known as the global financial crisis, all these three indices went through a huge downward jump. As for September 2002, both of S&P 500 and Eurostox suffered a big drop, which reflects the impact of the dot-com bubble. All the evidence here shows the significance of developing the shared-jump diffusion model.
In this essay, a shared-jump diffusion model is developed to analyze financial shocks and contagion effects among global financial markets. The framework of options pricing under this new model by the Fourier Space Time-stepping (FST) method is presented. The FST method uses properties of the Fourier transform to convert the PIDE into a linear first-order ODE. As the numerical results shown, second order convergence is obtained by using the FST method under all three jump diffusion models. The option value solved by the FST method under the shared-jump diffusion model is consistent with Monte Carlo simulations, which reflects the accuracy and efficiency of the FST algorithm.

Furthermore, constant padding is introduced to remedy the wrap-around error for both of one- and two-dimensional option pricing cases. The methodology of expanding the original domain is elaborated in detail. Special treatment for the shared-jump diffusion model is also discussed. Stock index data in recent decades from US, Europe and Britain are gathered to conduct empirical data analysis. It can be shown that all three indices went through downward jumps during the dot-com bubble in 2002 and the global financial crisis in 2008.

As discussed in this essay, further research can include:

- Extending the FST method to price American and Asian options under the shared-jump diffusion model
- Developing more useful and efficient techniques to reduce the wrap-around error for single- and multi-asset cases
- Deriving a systematic framework for parameter estimation under the shared-jump diffusion model
APPENDICES
Appendix A

Fourier Transforms of Distributions

A.1 Fourier Transform of Normal Distribution

The density function of normal distribution can be written as

$$f(y) = \frac{1}{\sqrt{2\pi\gamma}} e^{-\frac{1}{2}(y-\mu)^2}.$$ 

So, the Fourier transform of $f(y)$ is defined as

$$F(k) := \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\gamma}} e^{-\frac{1}{2}(y-\mu)^2} e^{-2\pi i ky} dy.$$ 

Let $x = y - \mu$, then $dx = dy$, so that

$$F(k) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\gamma}} e^{-\frac{1}{2}(x)^2} e^{-2\pi i k(x+\mu)} dx$$

$$= \frac{e^{-2\pi i k\mu}}{\sqrt{2\pi\gamma}} \int_{-\infty}^{\infty} e^{-\frac{1}{2}(x)^2} e^{-2\pi i kx} dx.$$ 

By Euler’s formula, $e^{-i\theta} = \cos(\theta) - i\sin(\theta)$ gives

$$F(k) = \frac{e^{-2\pi i k\mu}}{\sqrt{2\pi\gamma}} \int_{-\infty}^{\infty} e^{-\frac{1}{2}(x)^2} \left(\cos(2\pi kx) - i\sin(2\pi kx)\right) dx$$

$$= \frac{e^{-2\pi i k\mu}}{\sqrt{2\pi\gamma}} \left( \int_{-\infty}^{\infty} e^{-\frac{1}{2}(x)^2} \cos(2\pi kx) dx - i \int_{-\infty}^{\infty} e^{-\frac{1}{2}(x)^2} \sin(2\pi kx) dx \right).$$

53
It is known that \(\sin(\theta)\) is an odd function while \(\cos(\theta)\) is an even function, so that 
\[
\int_{-\infty}^{\infty} e^{-\frac{1}{2} \left( \frac{x}{\gamma} \right)^2} \sin(2\pi k x)\, dx = 0.
\]
Therefore, simplifying the equation above gives

\[
F(k) = e^{-2\pi i k \mu} \cdot \int_{0}^{\infty} e^{-\frac{1}{2} \left( \frac{x}{\gamma} \right)^2} \cos(2\pi k x)\, dx.
\] (A.1)

By the integral formula, 
\[
\int_{0}^{\infty} e^{-at^2} \cos(2st)\, dt = \frac{1}{2} \sqrt{\frac{\pi}{a}} e^{-\frac{s^2}{a}},
\]
(A.1) can be expressed as

\[
F(k) = \frac{e^{-2\pi i k \mu}}{\sqrt{2\pi \gamma}} \cdot 2 \cdot \int_{0}^{\infty} e^{-ax^2} \cos(2sx)\, dx
\]

where \(a = \frac{1}{2\gamma}\) and \(s = \pi k\). Using this formula obtains

\[
F(k) = \frac{e^{-2\pi i k \mu}}{\sqrt{2\pi \gamma}} \cdot \int_{0}^{\infty} e^{-ax^2} \cos(2s x)\, dx
\]

\[
= e^{-2\pi i k \mu} \cdot \frac{1}{\sqrt{2\pi a}} \sqrt{\frac{\pi}{a}} e^{-\frac{s^2}{a}}
\]

\[
= e^{-2\pi i k \mu} \cdot \frac{1}{\sqrt{2\gamma}} \sqrt{2\gamma} e^{-\frac{(\pi k)^2}{2\gamma^2}}
\]

\[
= e^{-2\pi i k \mu} \cdot e^{-2(\pi k \gamma)^2}
\]

\[
= e^{-2(\pi i k \mu + (\pi k \gamma)^2)}.
\]

Hence, \(F(k) = e^{-2(\pi i k \mu + (\pi k \gamma)^2)}\) and \(\bar{F}(k) = e^{2(\pi i k \mu - (\pi k \gamma)^2)}\).
A.2 Fourier Transform of Double Exponential Distribution

The density function of double exponential distribution is given by

\[ f(y) = p\eta_1 e^{-y\eta_1} \cdot 1_{\{y \geq 0\}} + (1 - p)\eta_2 e^{y\eta_2} \cdot 1_{\{y \leq 0\}}. \]

So, the Fourier transform of \( f(y) \) is defined as

\[ F(k) := \int_{-\infty}^{\infty} \left( p\eta_1 e^{-y\eta_1} \cdot 1_{\{y \geq 0\}} + (1 - p)\eta_2 e^{y\eta_2} \cdot 1_{\{y \leq 0\}} \right) e^{-2\pi iky} \, dy. \]

It is easy to simplify the equation above as follows

\[
F(k) = \int_{-\infty}^{\infty} p\eta_1 e^{-y\eta_1} \cdot e^{-2\pi iky} \, dy + \int_{-\infty}^{\infty} (1 - p)\eta_2 e^{y\eta_2} \cdot e^{-2\pi iky} \, dy
\]

\[
= p\eta_1 \int_{0}^{\infty} e^{-y(\eta_1 + 2\pi ik)} \, dy + (1 - p)\eta_2 \int_{-\infty}^{0} e^{y(\eta_2 - 2\pi ik)} \, dy
\]

\[
= p\eta_1 \left[ \frac{-1}{\eta_1 + 2\pi ik} e^{-y(\eta_1 + 2\pi ik)} \right]_{0}^{\infty} + (1 - p)\eta_2 \left[ \frac{1}{\eta_2 - 2\pi ik} e^{y(\eta_2 - 2\pi ik)} \right]_{-\infty}^{0}
\]

\[
= p\eta_1 \left( \frac{1}{\eta_1 + 2\pi ik} \right) + (1 - p)\eta_2 \left( \frac{1}{\eta_2 - 2\pi ik} \right)
\]

\[
= \frac{p}{1 + 2\pi ik(\frac{1}{\eta_1})} + \frac{1 - p}{1 - 2\pi ik(\frac{1}{\eta_2})}.
\]

Hence, \( F(k) = \frac{p}{1 + 2\pi ik(\frac{1}{\eta_1})} + \frac{1 - p}{1 - 2\pi ik(\frac{1}{\eta_2})} \) and \( \tilde{F}(k) = \frac{p}{1 + 2\pi ik(\frac{1}{\eta_1})} + \frac{1 - p}{1 + 2\pi ik(\frac{1}{\eta_2})} \).
Appendix B

Algorithms for European Options

B.1 FST under One-factor Jump Diffusion Model

Data: $S, K, r, q, T, N, \sigma, \lambda, \mu, \gamma$
Result: $V$

$x \leftarrow x_{\text{min}} + (0, 1, \cdots, N - 1) \frac{x_{\text{max}} - x_{\text{min}}}{N}$;
$k \leftarrow (0, 1, \cdots, \frac{N}{2}, -\frac{N}{2} + 1, \cdots, -1) \frac{1}{(x_{\text{max}} - x_{\text{min}})}$;
$p \leftarrow \max(Se^x - K, 0)$(call); $\max(K - Se^x, 0)$(put);
$\kappa \leftarrow e^{(\mu + \frac{1}{2} \gamma^2)} - 1$;
for $j \leftarrow 1$ to $N$ do
$\Psi_j \leftarrow \frac{-\sigma^2}{2}(2\pi k_j)^2 + (r - q - \lambda \kappa - \frac{\sigma^2}{2})(2\pi i k_j) - (r + \lambda) + \lambda e^{2(\pi i k_j \mu - (\pi k_j \gamma)^2)}$
end
$v \leftarrow p$;
$v \leftarrow IDFT[DFT[v] \cdot e^{\Psi_T}]$;
$V \leftarrow$ interpolation of $v$ at $x = 0$;

Algorithm 1: FST under One-factor Jump Diffusion Model: Merton jump density
Data: $S, K, r, q, T, N, \sigma, \lambda, p, \eta_1, \eta_2$

Result: $V$

$x \leftarrow x_{\text{min}} + (0, 1, \cdots, N-1) \frac{x_{\text{max}} - x_{\text{min}}}{N}$;

$k \leftarrow (0, 1, \cdots, \frac{N}{2}, -\frac{N}{2} + 1, \cdots, -1) \frac{1}{(x_{\text{max}} - x_{\text{min}})}$;

$p \leftarrow \max(Se^x - K, 0)\text{ (call)}; \max(K - Se^x, 0)\text{ (put)}$;

$\kappa \leftarrow p \frac{\eta_1}{\eta_1 - 1} + (1 - p) \frac{\eta_2}{\eta_2 + 1} - 1$;

for $j \leftarrow 1$ to $N$ do

$\Psi_j \leftarrow -\frac{\sigma^2}{2} (2\pi k_j)^2 + (r - q - \lambda \kappa - \frac{\sigma^2}{2})(2\pi i k_j) - (r + \lambda) + \lambda \left( \frac{p}{1 - 2\pi i k_j} \frac{1}{\eta_1} + \frac{1 - p}{1 + 2\pi i k_j} \frac{1}{\eta_2} \right)$

end

$v \leftarrow p$;

$v \leftarrow \text{IDFT}[\text{DFT}[v] \cdot e^{\Psi T}]$;

$V \leftarrow \text{interpolation of } v \text{ at } x = 0$;

**Algorithm 2:** FST under One-factor Jump Diffusion Model: *Kou jump density*
B.2 FST under Two-factor Jump Diffusion Model

Data: \(S, K, B_1, B_2, r, q_1, q_2, T, N, \sigma_1, \sigma_2, \rho, \lambda_1, \lambda_2, \mu_1, \mu_2, \gamma_1, \gamma_2\)

Result: \(V\)

\[
x_1, x_2 \leftarrow x_{\min} + (0, 1, \cdots, N - 1) \frac{x_{\max} - x_{\min}}{N};
\]

\[
k_1, k_2 \leftarrow (0, 1, \cdots, \frac{N}{2}, -\frac{N}{2} + 1, \cdots, -1) \frac{1}{(x_{\max} - x_{\min})};
\]

\[
\kappa_1 \leftarrow e^{(\mu_1 + \frac{1}{2} \gamma_1^2)} - 1;
\]

\[
\kappa_2 \leftarrow e^{(\mu_2 + \frac{1}{2} \gamma_2^2)} - 1;
\]

for \(l \leftarrow 1\) to \(N\) do

\[
\text{for } j \leftarrow 1\text{ to } N\text{ do}
\]

\[
p_{lj} \leftarrow \max(B_2S e^{x_2j} - B_1S e^{x_1j} - K, 0) \text{ (spread call)};
\]

\[
\max(K - B_2S e^{x_2j} + B_1S e^{x_1j}, 0) \text{ (spread put)};
\]

end

end

for \(l \leftarrow 1\) to \(N\) do

\[
\text{for } j \leftarrow 1\text{ to } N\text{ do}
\]

\[
\Psi_{lj} \leftarrow -\frac{\sigma_1^2}{2}(2\pi k_1) + \frac{\sigma_2^2}{2}(2\pi k_2)^2 + (r - q_1 - \lambda_1 \kappa_1 - \frac{\sigma_1^2}{2})(2\pi i k_1) + (r - q_2 - \lambda_2 \kappa_2 - \frac{\sigma_2^2}{2})(2\pi i k_2) + \rho \sigma_1 \sigma_2 (2\pi k_1)(2\pi k_2) - (r + \lambda_1 + \lambda_2) + \lambda_1 e^{2(\pi i k_1 \mu_1) - (\pi k_1 \gamma_1)^2} + \lambda_2 e^{2(\pi i k_2 \mu_2 - (\pi k_2 \gamma_2)^2)}
\]

end

end

\(v \leftarrow p;\)

\(v \leftarrow IDFT[DFT[v] \cdot e^{\Psi T}];\)

\(V \leftarrow \text{interpolation of } v \text{ at } x_1 = 0, x_2 = 0;\)

**Algorithm 3:** FST under Two-factor Jump Diffusion Model: *Merton jump density*
B.3 FST under Shared-jump Diffusion Model

Data: \( S, K, B_1, B_2, r, T, N, \sigma_1, \sigma_2, \rho, \lambda, p, \eta_1, \eta_2 \)

Result: \( V \)

\[
x_1, x_2 \leftarrow x_{\text{min}} + (0, 1, \cdots, N-1) \frac{x_{\text{max}} - x_{\text{min}}}{N};
\]

\[
k_1, k_2 \leftarrow (0, 1, \cdots, \frac{N}{2}, -\frac{N}{2} + 1, \cdots, -1) \frac{1}{x_{\text{max}} - x_{\text{min}}};
\]

\[
\kappa \leftarrow p \frac{\eta_1}{\eta_1 - 1} + (1 - p) \frac{\eta_2}{\eta_2 + 1} - 1;
\]

\[
\text{for } l \leftarrow 1 \text{ to } N \text{ do}
\]

\[
\text{for } j \leftarrow 1 \text{ to } N \text{ do}
\]

\[
p_{lj} \leftarrow \max(B_2 S e^{x_2 j} - B_1 S e^{x_1 j} - K, 0) \text{ (spread call)};
\]

\[
\max(K - B_2 S e^{x_2 j} + B_1 S e^{x_1 j}, 0) \text{ (spread put)};
\]

\[
\text{end}
\]

\[
\text{end}
\]

\[
\text{for } l \leftarrow 1 \text{ to } N \text{ do}
\]

\[
\text{for } j \leftarrow 1 \text{ to } N \text{ do}
\]

\[
\Psi_{lj} \leftarrow -\frac{\sigma_1^2}{2} (2\pi k_1)^2 - \frac{\sigma_2^2}{2} (2\pi k_2)^2 + (r - \lambda \kappa - \frac{\sigma_1^2}{2})(2\pi ik_1) + (r - \lambda \kappa - \frac{\sigma_2^2}{2})(2\pi ik_2) +
\]

\[
\rho \sigma_1 \sigma_2 (2\pi k_1)(2\pi k_2) - (r + \lambda) + \lambda \left( \frac{p}{1 - 2\pi i (k_1 + k_2)} + \frac{1-p}{1 + 2\pi i (k_1 + k_2)} \right);
\]

\[
\text{end}
\]

\[
\text{end}
\]

\[
v \leftarrow p;
\]

\[
v \leftarrow \text{IDFT}[\text{DFT}[v] \cdot e^{\Psi T}];
\]

\[
V \leftarrow \text{interpolation of } v \text{ at } x_1 = 0, x_2 = 0;
\]

Algorithm 4: FST under Shared-jump Diffusion Model: Kou jump density
Appendix C

Monte Carlo Approach

C.1 Methodology

Consider the following stochastic differential equations (SDEs) in the risk neutral world:

\[
\frac{dS_1}{S_1} = (r - \lambda^Q \kappa^Q) dt + \sigma_1 dZ_1 + (\eta^Q - 1) dq, \tag{C.1}
\]

\[
\frac{dS_2}{S_2} = (r - \lambda^Q \kappa^Q) dt + \sigma_2 dZ_2 + (\eta^Q - 1) dq, \tag{C.2}
\]

\[
dZ_1 dZ_2 = \rho dt, \tag{C.3}
\]

where \(\lambda^Q, \kappa^Q\) and \(\eta^Q\) are all risk adjusted parameters under the \(Q\)-measure.

After a log transformation \(X_t = \log(S_t)\), equations (C.1) and (C.2) can be rewritten as:

\[
dx_{1,t} = \left( r - \frac{\sigma_1^2}{2} - \lambda^Q \kappa^Q \right) dt + \sigma_1 dZ_1 + \log(\eta^Q_t) dq, \tag{C.4}
\]

\[
dx_{2,t} = \left( r - \frac{\sigma_2^2}{2} - \lambda^Q \kappa^Q \right) dt + \sigma_2 dZ_2 + \log(\eta^Q_t) dq \tag{C.5}
\]

where \(dq = \begin{cases} 
0, & \text{with probability } 1 - \lambda^Q dt, \\
1, & \text{with probability } \lambda^Q dt.
\end{cases}\)}
Under the shared-jump diffusion model, \( \log(\eta) \) follows a double exponential distribution. So, let \( y = \log(\eta) \), then the density of \( y \), \( f(y) \) is:

\[
f(y) = p\eta e^{-\eta y} \cdot 1_{\{y \geq 0\}} + (1 - p)\eta e^{\eta y} \cdot 1_{\{y \leq 0\}}.
\]

Also, \( \kappa = E[\eta - 1] = p\frac{\eta_1}{n_1} + (1 - p)\frac{\eta_2}{n_2 + 1} - 1 \). In addition, \( dq \) is assumed to be independent with \( \eta_t \). The condition \( \lambda \Delta t \ll 1 \) guarantees that the probability of having more than one jump in \([t, t + \Delta t]\) is negligible.

Suppose no jump occurs in \([t, t + \Delta t]\), by forward Euler, the equations (C.4) and (C.5) give:

\[
X_{1,t+\Delta t} = X_{1,t} + (r - \frac{\sigma_1^2}{2} - \lambda Q \kappa)\Delta t + \sigma_1 \phi_{1,t} \sqrt{\Delta t},
\]

\[
X_{2,t+\Delta t} = X_{2,t} + (r - \frac{\sigma_2^2}{2} - \lambda Q \kappa)\Delta t + \sigma_2 \phi_{2,t} \sqrt{\Delta t},
\]

where \( \Delta t \) is the finite timestep and \( \phi_{1,t} \) and \( \phi_{2,t} \) are two random numbers from standard normal distribution. The correlation between \( \phi_{1,t} \) and \( \phi_{2,t} \) is \( \rho \). If there exists a jump in \([t, t + \Delta t]\), by forward Euler, the equations (C.4) and (C.5) give:

\[
X_{1,t+\Delta t} = X_{1,t} + (r - \frac{\sigma_1^2}{2} - \lambda Q \kappa)\Delta t + \sigma_1 \phi_{1,t} \sqrt{\Delta t} + y_t,
\]

\[
X_{2,t+\Delta t} = X_{2,t} + (r - \frac{\sigma_2^2}{2} - \lambda Q \kappa)\Delta t + \sigma_2 \phi_{2,t} \sqrt{\Delta t} + y_t
\]

where \( y_t \) is a random number from the double exponential distribution.

Therefore, the realized paths for \( S_1 \) and \( S_2 \) can be simulated using the method introduced above. After \( N \) timesteps, with \( T = N\Delta t \), the option value can be derived given the payoff function: \( V = Payoff(S_{1,T}, S_{2,T}) \). Suppose the total number of trials is \( M \) and denote the payoff after the \( m^{th} \) trial as \( Payoff(m) \), where \( m = 1, \ldots, M \). So, the no-arbitrage value of the option is:

\[
Value = e^{-rT} E^Q[Payoff] \\
\approx e^{-rT} \frac{1}{M} \sum_{m=1}^{M} Payoff(m).
\]

Furthermore, the sampling error can be estimated via a statistical approach. With the estimated mean of the sample

\[
\hat{\mu} = e^{-rT} \frac{1}{M} \sum_{m=1}^{M} Payoff(m)
\]
and the standard deviation of the estimate

\[
\omega = \sqrt{\frac{1}{M - 1} \sum_{m=1}^{M} (e^{-rT \text{Payoff}(m) - \hat{\mu}})^2}, \quad (C.12)
\]

the 95% confidence interval for the actual value of the option \( V \) is:

\[
\hat{\mu} - \frac{1.96 \omega}{\sqrt{M}} \leq V \leq \hat{\mu} + \frac{1.96 \omega}{\sqrt{M}}, \quad (C.13)
\]
C.2 Algorithm for Monte Carlo Simulation

Data: \( S, K, B_1, B_2, r, T, \sigma_1, \sigma_2, \rho, \lambda, p, \eta_1, \eta_2, N, M \)

Result: \( V, sd \)

\[
\Delta t \leftarrow \frac{T}{N};
\]

\[
\kappa \leftarrow p\eta_1^{\frac{\eta_1}{\eta_1-1}} + (1-p)\frac{\eta_2}{\eta_2+1} - 1;
\]

\[
R \leftarrow \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix};
\]

\[
X_{1,old}, X_{2,old} \leftarrow \log(S);
\]

\[
L \leftarrow \text{Cholesky Decomposition}(R);
\]

for \( i \leftarrow 1 \) to \( N \) do

\[
\text{JumpCheck} \leftarrow (\text{rand}(M, 1) \leq \lambda \Delta t);
\]

\[
\text{RandomDraw} \leftarrow \text{rand}(M, 1);
\]

\[
\text{JumpSize} \leftarrow (\text{RandomDraw} \leq p) \cdot \text{exprnd}(\frac{1}{\eta_1}, M, 1) - (\text{RandomDraw} > p) \cdot \text{exprnd}(\frac{1}{\eta_2}, M, 1);
\]

\[
\text{JumpSize} \leftarrow \text{JumpSize} \cdot \text{JumpCheck};
\]

\[
\Phi \leftarrow \text{randn}(M, 2)L;
\]

\[
X_{1,new} \leftarrow X_{1,old} + \left( r - \frac{\sigma_1^2}{2} - \lambda \kappa \right) \Delta t + \sigma_1 \sqrt{\Delta t} \cdot \Phi(:, 1) + \text{JumpSize};
\]

\[
X_{1,old} \leftarrow X_{1,new};
\]

\[
X_{2,new} \leftarrow X_{2,old} + \left( r - \frac{\sigma_2^2}{2} - \lambda \kappa \right) \Delta t + \sigma_2 \sqrt{\Delta t} \cdot \Phi(:, 2) + \text{JumpSize};
\]

\[
X_{2,old} \leftarrow X_{2,new};
\]

end

\[
(S_1, S_2) \leftarrow (e^{X_{1,new}}, e^{X_{2,new}});
\]

\[
V \leftarrow \text{mean}(e^{-rT} \cdot \text{Payoff}(S_1, S_2, K, B_1, B_2));
\]

\[
sd \leftarrow \text{std}(e^{-rT} \cdot \text{Payoff}(S_1, S_2, K, B_1, B_2));
\]

**Algorithm 5:** Monte Carlo under Shared-jump Diffusion Model: *Kou jump density*
References


