# Commodity Options Pricing by the Fourier Space Time-stepping Method

by

Chunhao Shan

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

Supervisors: Prof. Peter Forsyth & Prof. George Labahn

Waterloo, Ontario, Canada, 2014

## Abstract

We study the framework of the Fourier space time-stepping (FST) method for pricing contingent claims in financial markets. By using the semi-Lagrangian method, the FST method is extended to the pricing of commodity contingent claims. We study the commodity one-factor and two-factor models and the related semi-Lagrangian schemes for these models. Numerical results of pricing commodity European and American options are presented.

## Acknowledgements

I would like to thank my supervisors, Prof. Peter Forsyth and Prof. George Labahn, for their guidance, patience and support throughout the year. I would also like to thank my family and friends.

## Dedication

This is dedicated to my parents.

# Table of Contents

**List of Tables** viii

**List of Figures** xii

**List of Algorithms** xiii

**1 Introduction** 1

**2 Mathematical Models** 4

2.1 Black-Scholes-Merton Model . . . . . . . . . . . . . . . . . . . . . . . . . . 4

2.2 Commodity Price Models . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

2.2.1 One-factor Model . . . . . . . . . . . . . . . . . . . . . . . . . . 6

2.2.2 Two-factor Model . . . . . . . . . . . . . . . . . . . . . . . . . . 6

**3 Fourier Space Time-stepping Method** 9

3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

3.1.1 Continuous Fourier Transform . . . . . . . . . . . . . . . . . . . 9

3.1.2 Discrete Fourier Transform . . . . . . . . . . . . . . . . . . . . . 10

3.2 FST Algorithm for Options under the BSM Model . . . . . . . . . . . . . . 13

3.2.1 Fully Implicit and Crank-Nicolson . . . . . . . . . . . . . . . . . 16

3.2.2 FST for European Option under the BSM Model . . . . . . . . . . 19

3.2.3 FST for American Option under the BSM Model . . . . . . . . . . 21

**4 Semi-Lagrangian Method** **23**
4.1 Advection-diffusion Equation with Constant Velocity . . . . . . . . . . . . 23
4.2 Advection-diffusion Equation with Non-constant Velocity . . . . . . . . . . 24
4.3 Application to the Commodity Price Models . . . . . . . . . . . . . . . . . 27
4.4 Interpolation Methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
4.5 Error of the Semi-Lagrangian Method . . . . . . . . . . . . . . . . . . . . . 31

**5 Numerical Results** **32**
5.1 Commodity Options under a One-factor Model . . . . . . . . . . . . . . . . 32
5.1.1 FST Algorithm for Options under a One-factor Model . . . . . . . . 32
5.1.2 Monte Carlo for European Option under a One-factor Model . . . . 36
5.1.3 FST for European Option under a One-factor Model . . . . . . . . 36
5.1.4 FST for American Option under a One-factor Model . . . . . . . . 40
5.2 Commodity Options under a Two-factor Model . . . . . . . . . . . . . . . 43
5.2.1 FST Algorithm for Options under a Two-factor Model . . . . . . . 43
5.2.2 Monte Carlo for European Option under a Two-factor Model . . . . 47
5.2.3 FST for European Option under a Two-factor Model . . . . . . . . 47
5.2.4 FST for American Option under a Two-factor Model . . . . . . . . 51

**6 Conclusions** **53**

**APPENDICES** **55**

**A Wrap-around Error** **56**

**References** **59**

# List of Tables

3.1 Single asset option parameters under BSM model. . . . . 20

3.2 Single asset European put under BSM model, FST with fully implicit, closed form solution = 3.75341839, first order convergence, including both space and time errors. . . . . 20

3.3 Single asset European put under BSM model, FST with Crank-Nicolson, closed form solution = 3.75341839, second order convergence, including both space and time errors. . . . . 20

3.4 Single asset American put under BSM model, FST with fully implicit, first order convergence, including both space and time errors. . . . . 21

3.5 Single asset American put under BSM model, FST with Crank-Nicolson, first order convergence, including both space and time errors. . . . . 22

3.6 Single asset American put under BSM model, finite difference with fully implicit (not vectorized), first order convergence, including both space and time errors. . . . . 22

3.7 Single asset American put under BSM model, finite difference with Crank-Nicolson (not vectorized), first order convergence, including both space and time errors. . . . . 22

5.1 Single asset commodity option parameters under one-factor model. . . . . 36

5.2 Single asset commodity European call under one-factor model, Monte Carlo. 36

5.3 Single asset commodity European call under one-factor model, fully implicit with linear interpolation, Monte Carlo solution = 0.87940762 (standard error = 0.00435362), first order convergence, including both space and time errors. 37

5.4 Single asset commodity European call under one-factor model, fully implicit with piecewise quadratic interpolation (not vectorized), Monte Carlo solution = 0.87940762 (standard error = 0.00435362), first order convergence, including both space and time errors. . . . . . . . . . . . . . . . . . . . . . 37

5.5 Single asset commodity European call under one-factor model, fully implicit with cubic spline interpolation, Monte Carlo solution = 0.87940762 (standard error = 0.00435362), first order convergence, including both space and time errors. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37

5.6 Single asset commodity European call under one-factor model, Crank-Nicolson with linear interpolation, Monte Carlo solution = 0.87940762 (standard error = 0.00435362), first order convergence, including both space and time errors. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38

5.7 Single asset commodity European call under one-factor model, Crank-Nicolson with piecewise quadratic interpolation (not vectorized), Monte Carlo solution = 0.87940762 (standard error = 0.00435362), second order convergence, including both space and time errors. . . . . . . . . . . . . . . . . . . . . . 38

5.8 Single asset commodity European call under one-factor model, Crank-Nicolson with cubic spline interpolation, Monte Carlo solution = 0.87940762 (standard error = 0.00435362), second order convergence, including both space and time errors. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

5.9 Single asset commodity American call under one-factor model, fully implicit with linear interpolation, first order convergence, including both space and time errors. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40

5.10 Single asset commodity American call under one-factor model, fully implicit with piecewise quadratic interpolation (not vectorized), first order convergence, including both space and time errors. . . . . . . . . . . . . . . . . . 40

5.11 Single asset commodity American call under one-factor model, fully implicit with cubic spline interpolation, first order convergence, including both space and time errors. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41

5.12 Single asset commodity American call under one-factor model, Crank-Nicolson with linear interpolation, first order convergence, including both space and time errors. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42

5.13 Single asset commodity American call under one-factor model, Crank-Nicolson with piecewise quadratic interpolation (not vectorized), first order convergence, including both space and time errors. . . . . . . . . . . . . . . . . . 42

5.14 Single asset commodity American call under one-factor model, Crank-Nicolson with cubic spline interpolation, first order convergence, including both space and time errors. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42

5.15 Single asset commodity option parameters under two-factor model. . . . . 47

5.16 Single asset commodity European put under two-factor model, Monte Carlo. 47

5.17 Single asset commodity European put under two-factor model, fully implicit with linear interpolation, Monte Carlo solution = 1.10497142 (standard error = 0.00477254), first order convergence, including both space and time errors. 48

5.18 Single asset commodity European put under two-factor model, fully implicit with piecewise quadratic interpolation (not vectorized), Monte Carlo solution = 1.10497142 (standard error = 0.00477254), first order convergence, including both space and time errors. . . . . . . . . . . . . . . . . . . . . 48

5.19 Single asset commodity European put under two-factor model, fully implicit with cubic spline interpolation, Monte Carlo solution = 1.10497142 (standard error = 0.00477254), first order convergence, including both space and time errors. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48

5.20 Single asset commodity European put under two-factor model, Crank-Nicolson with linear interpolation, Monte Carlo solution = 1.10497142 (standard error = 0.00477254), first order convergence, including both space and time errors. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49

5.21 Single asset commodity European put under two-factor model, Crank-Nicolson with piecewise quadratic interpolation (not vectorized), Monte Carlo solution = 1.10497142 (standard error = 0.00477254), second order convergence, including both space and time errors. . . . . . . . . . . . . . . . . . . . . 49

5.22 Single asset commodity European put under two-factor model, Crank-Nicolson with cubic spline interpolation, Monte Carlo solution = 1.10497142 (standard error = 0.00477254), second order convergence, including both space and time errors. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50

5.23 Single asset commodity American put under two-factor model, fully implicit with linear interpolation, first order convergence, including both space and time errors. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51

5.24 Single asset commodity American put under two-factor model, fully implicit with piecewise quadratic interpolation (not vectorized), first order convergence, including both space and time errors. . . . . . . . . . . . . . . . . 51

5.25 Single asset commodity American put under two-factor model, fully implicit with cubic spline interpolation, first order convergence, including both space and time errors. . . . . 51

5.26 Single asset commodity American put under two-factor model, Crank-Nicolson with linear interpolation, first order convergence, including both space and time errors. . . . . 52

5.27 Single asset commodity American put under two-factor model, Crank-Nicolson with piecewise quadratic interpolation (not vectorized), first order convergence, including both space and time errors. . . . . 52

5.28 Single asset commodity American put under two-factor model, Crank-Nicolson with cubic spline interpolation, first order convergence, including both space and time errors. . . . . 52

A.1 Single asset option parameters under BSM model. . . . . 58

A.2 Single asset European call under BSM model, wrap-around error. . . . . 58

# List of Figures

3.1 Approximation for the discrete Fourier transform. . . . . . . . . . . . . . . . 12
3.2 Approximation for the inverse discrete Fourier transform. . . . . . . . . . . 13

4.1 Semi-Lagrangian scheme. . . . . . . . . . . . . . . . . . . . . . . . . . . 26
4.2 Linear interpolation and piecewise quadratic interpolation. . . . . . . . . . 30

# List of Algorithms

1 FST with fully implicit for single asset option under BSM model. . . . . . . 18

2 FST with Crank-Nicolson for single asset option under BSM model. . . . . . 19

3 FST with fully implicit for single asset commodity option under one-factor model. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34

4 FST with Crank-Nicolson for single asset commodity option under one-factor model. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

5 FST with fully implicit for single asset commodity option under two-factor model. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

6 FST with Crank-Nicolson for single asset commodity option under two-factor model. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46

7 FST with fully implicit for single asset option under BSM model (with zero padding). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57

# Chapter 1

# Introduction

In finance, a derivative is a type of contract whose price is dependent on and derived from a particular underlying asset or a basket of underlying assets. The most common underlying assets include stocks, bonds, commodities, currencies, interest rates and market indexes. A derivative is generally used to insure against the risk of price movement, which is usually called hedging. It can also be used in many other areas in financial markets, even for speculative purposes.

Options, futures contracts, forward contracts and swaps are the most common types of derivatives. An option is a type of contract which gives the owner the right, but not the obligation, to buy or sell an underlying asset or instrument at a specified strike price on or before a specified date (maturity). An option that can only be exercised on expiration is called a European option. The type of option which gives the owner the right to exercise on any trading day before expiry is called an American option. A futures contract is a contract between two parties to buy or sell a specified asset of standardized quantity and quality for a price agreed upon today (futures price) with delivery and payment occurring at a specified future date, the delivery date.

As various types of financial derivatives play more significant roles in financial markets, accurate and efficient methods to value derivative contracts are in high demand. The Black-Scholes model was first proposed by Black and Scholes (1973) [1]. In this model, they assume the underlying asset price follows a geometric Brownian motion, which describes the stochastic behaviour of the underlying asset. They then determine the price as the solution of a partial differential equation (PDE). Based on this Black-Scholes partial differential equation, there are various numerical methods to solve the valuation of the option, such as finite difference, finite volume and finite element methods.

The Fourier space time-stepping method was first developed by Jackson, Jaimungal and Surkov (2008) [6]. This method uses the Fourier transform to numerically solve the partial differential equation. The continuous Fourier transform is a linear operator which maps spatial derivatives into multiplications in Fourier space. Due to convenient properties of the Fourier transform, we can convert the valuation in real space to Fourier space, where the partial differential equation will become an ordinary differential equation (ODE). We then have various straightforward methods to solve the ODE. Meanwhile, it is easy to add constraints or other conditions between two timesteps to allow for using the Fourier transform method to price American or other path dependent options.

We know that many financial derivatives, such as options, are dependent on the underlying assets of the stock prices, which can be modelled as a geometric Brownian motion. However, there are also many contingent claims based on commodity prices. Straightforwardly, by assuming interest rates and convenience yields[1] are constant, we can extend our pricing methods for common stock options[2] to derivatives based on commodities. However, this assumption is not very suitable for commodity prices since it implies that the volatility of future prices[3] is equal to the volatility of spot prices[4]. Intuitively, in a commodity market, when prices are relatively high, supply will increase since higher cost producers of the commodity will enter the market. This will put a downward pressure on prices. Conversely, when prices are relatively low, supply will decrease since some of the higher cost producers will exit the market. This will put an upward pressure on prices [13]. The impact of relative prices on the supply of the commodity will induce mean reversion in commodity prices. Gibson (1990) [4], Ross (1997) [11], and Schwartz (2000) [13] developed models that describe mean reversion properties of commodity prices and derived the corresponding partial differential equations for valuation of the futures and options based on commodity prices.

[1]Convenience yield can be seen as the flow of services accruing to the holder of the spot commodity but not to the owner of a futures contract [13].

[2]Options whose underlying assets are stock prices.

[3]The market price under the condition that an commodity asset can be bought or sold for delivery in the future.

[4]The current market price under the condition that an commodity asset is bought or sold for immediate payment and delivery.

In the commodity price models, including one-factor and two-factor models, an Ornstein-Uhlenbeck process is introduced to describe the mean-reverting behaviour of commodity prices. This induces a non-constant drift term in the pricing partial differential equation. In order to solve the option pricing PDE by the Fourier space time-stepping method, we can use the semi-Lagrangian method to deal with the non-constant velocity term in the advection-diffusion equation [16]. The purpose of this paper is to combine the semi-Lagrangian method and the Fourier space time-stepping method to compute the values of some typical options based on commodity prices.

This paper is structured as follows. In Chapter 2, we introduce the Black-Scholes model and two models for commodity prices. Chapter 3 presents the framework of the Fourier space time-stepping method and its application to some typical stock options. In Chapter 4, we will provide the introduction of the semi-Lagrangian method and its application to commodity options valuation. Chapter 5 gives the numerical examples for one-factor and two-factor models. Chapter 6 lists the conclusions.

# Chapter 2

# Mathematical Models

## 2.1 Black-Scholes-Merton Model

In the Black-Scholes-Merton (BSM) model [1], the price of an underlying asset is assumed to follow a geometric Brownian motion. The stochastic differential equation for the underlying asset price $S$ can be written as:

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

where:

| | |
|---|---|
| $\mu$ | is the drift rate, |
| $\sigma$ | is the volatility, |
| $dZ$ | is the increment of a standard Wiener process. |

Moreover, by no-arbitrage arguments, we can write the dynamics of the process under the equivalent martingale measure[1] as:

$$\frac{dS}{S} = rdt + \sigma dZ^*, \tag{2.2}$$

[1]Equivalent martingale measure, also called risk-neutral measure, is used in the pricing of financial derivatives due to the fundamental theorem of asset pricing. Under this measure, the current value of financial assets is equal to their expected payoffs in the future discounted at the risk-free rate.

where:

| | |
|---|---|
| $r$ | is the risk-free rate, |
| $dZ^*$ | is the increment of a standard Wiener process under the equivalent martingale measure. |

Let $\mathcal{V}(S,\tau)$ be the value of a European option where $\tau$ is the time to maturity. Assuming the payoff of the option at maturity $\mathcal{V}(S,0)$ is known, we can derive a partial differential equation to solve the price of the option by Ito's Lemma. This is the Black-Scholes equation:

$$\mathcal{V}_\tau = \frac{1}{2}\sigma^2 S^2 \mathcal{V}_{SS} + rS\mathcal{V}_S - r\mathcal{V}, \tag{2.3}$$

with the initial condition (for a call):

$$\mathcal{V}(S,0) = \max(S-K,0), \tag{2.4}$$

or (for a put):

$$\mathcal{V}(S,0) = \max(K-S,0). \tag{2.5}$$

Here $K$ is the strike price.

## 2.2 Commodity Price Models

Commodity price models are used to describe the stochastic behaviour of the commodity spot price, which is the current market price under the condition that a commodity asset is bought or sold for immediate payment and delivery. Commodity prices are assumed to be mean reverting. A straightforward approach is to use a mean-reverting Ornstein-Uhlenbeck process to model the commodity spot price, which is a one-factor model. However, this model has an implicit assumption that the convenience yield has a certain relation with the spot price. The convenience yield can be seen as the flow of services accruing to the holder of the spot commodity but not to the owner of a futures contract [13]. If we let the convenience yield also follow a mean-reverting Ornstein-Uhlenbeck process that is correlated with the spot price, this leads to a two-factor model. Moreover, by assuming the interest rate is non-constant, we then have a three-factor model. In this paper, we will focus on one-factor and two-factor models.

### 2.2.1 One-factor Model

We first assume that the interest rate $r$ is constant, and the logarithm of the spot price $S$ of the commodity follows a mean-reverting Ornstein-Uhlenbeck process:

$$\frac{dS}{S} = \kappa(\mu - \log S)dt + \sigma dZ, \tag{2.6}$$

where:

| | |
|---|---|
| $\kappa$ | is the magnitude of the speed of adjustment, $\kappa > 0$, |
| $\mu$ | is the drift rate, |
| $\sigma$ | is the volatility, |
| $dZ$ | is the increment of a standard Wiener process. |

By no-arbitrage arguments, we can write the dynamics of the process under the equivalent martingale measure as:

$$\frac{dS}{S} = \kappa(\mu - \lambda - \log S)dt + \sigma dZ^*, \tag{2.7}$$

where:

| | |
|---|---|
| $\lambda$ | is the market price of risk, |
| $dZ^*$ | is the increment of a standard Wiener process under the equivalent martingale measure. |

Let $\mathcal{V}(S, \tau)$ be the value of a commodity option price where $\tau$ is the time to maturity. Then by Ito's Lemma, we can derive a partial differential equation to solve the option price:

$$\mathcal{V}_\tau = \frac{1}{2}\sigma^2 S^2 \mathcal{V}_{SS} + \kappa(\mu - \lambda - \log S)S\mathcal{V}_S, \tag{2.8}$$

with initial condition given in equation (2.4) or (2.5).

### 2.2.2 Two-factor Model

Unlike the one-factor model, a two-factor model was developed by Gibson and Schwartz (1990) [4]. The first factor is the spot price of the commodity $S$ and the second is the instantaneous convenience yield $\delta$. These factors are assumed to follow a joint Ornstein-Uhlenbeck process:

$$\begin{aligned} \frac{dS}{S} &= (\mu - \delta)dt + \sigma_1 dZ_1, \\ d\delta &= \kappa(\alpha - \delta)dt + \sigma_2 dZ_2, \end{aligned} \tag{2.9}$$

where the increments of two standard Wiener processes are correlated with:

$$dZ_1 \cdot dZ_2 = \rho dt, \tag{2.10}$$

where:

| | |
|---|---|
| $\kappa$ | is the magnitude of the speed of adjustment, $\kappa > 0$ |
| $\mu$ | is the drift rate, |
| $\alpha$ | is the long run log price, |
| $\sigma_1$ | is the volatility of the commodity spot price $S$, |
| $\sigma_2$ | is the volatility of the instantaneous convenience yield $\delta$, |
| $\rho$ | is the correlation of two standard Wiener processes. |
| $dZ_1, dZ_2$ | are the increments of two standard Wiener processes. |

Note that, if we let the instantaneous convenience yield $\delta$ be a deterministic function:

$$\delta = \kappa \log S, \tag{2.11}$$

then this two-factor model will be reduced to the one-factor model. Moreover, by no-arbitrage arguments, we can write the dynamics of the process under the equivalent martingale measure as:

$$\begin{aligned} \frac{dS}{S} &= (r - \delta)dt + \sigma_1 dZ_1^*, \\ d\delta &= [\kappa(\alpha - \delta) - \lambda]dt + \sigma_2 dZ_2^*, \\ dZ_1^* \cdot dZ_2^* &= \rho dt, \end{aligned} \tag{2.12}$$

where:

| | |
|---|---|
| $r$ | is the risk-free rate, |
| $\lambda$ | is the market price of convenience yield risk, |
| $dZ_1^*, dZ_2^*$ | are the increments of two standard Wiener processes under the equivalent martingale measure. |

Let $\mathcal{V}(S, \delta, \tau)$ be the value of a commodity option price where $\tau$ is the time to maturity. Then by Ito's Lemma, we can derive a partial differential equation to solve the option price:

$$\mathcal{V}_\tau = \frac{1}{2}\sigma_1^2 S^2 \mathcal{V}_{SS} + \sigma_1\sigma_2\rho S\mathcal{V}_{S\delta} + \frac{1}{2}\sigma_2^2\mathcal{V}_{\delta\delta} + (r-\delta)S\mathcal{V}_S + \kappa[(\alpha-\delta)-\lambda]\mathcal{V}_\delta, \tag{2.13}$$

with initial condition given in equation (2.4) or (2.5).

# Chapter 3

# Fourier Space Time-stepping Method

## 3.1 Introduction

### 3.1.1 Continuous Fourier Transform

A function in the space domain $f(x)$ can be transformed to a function in the frequency domain $F(k)$ by using the continuous Fourier transform (CFT). The one-dimensional continuous Fourier transform of a function $f(x)$ is defined as:

$$F(k) = \mathcal{F}\left[f(x)\right](k) := \int_{-\infty}^{\infty} f(x) e^{-i2\pi kx} dx. \tag{3.1}$$

The one-dimensional inverse Fourier transform of a function $F(k)$ is defined as:

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

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

$$\mathcal{F}\left[\frac{\partial}{\partial \tau} v(x,\tau)\right](k) = \frac{\partial}{\partial \tau} \mathcal{F}\left[v(x,\tau)\right](k). \tag{3.3}$$

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

$$\mathcal{F}\left[\frac{\partial^n}{\partial x^n}v(x,\tau)\right](k) = (2\pi i k)^n \mathcal{F}\left[v(x,\tau)\right](k). \tag{3.4}$$

### 3.1.2 Discrete Fourier Transform

In general, we cannot usually compute the exact value of the continuous Fourier transform. For numerical computations, we need to approximate this by the discrete Fourier transform (DFT). We discretize the domain in real space as:

$$x_m = x_{\min} + m \cdot \Delta x, \tag{3.5}$$

where $m = 0, 1, \cdots, N-1, \Delta x = \frac{\hat{x}}{N}$ and $\hat{x} = x_{\max} - x_{\min}$. Note here $N$ is the number of nodes in real space. Then the CFT can be approximated by:

$$F(k) \approx \int_{x_{\min}}^{x_{\max}} e^{-i2\pi kx} f(x)dx. \tag{3.6}$$

We discretize the domain in Fourier space as:

$$k_n = \frac{n}{\hat{x}}, \tag{3.7}$$

where $n = -\frac{N}{2}+1, \cdots, \frac{N}{2}$. Then the maximum frequency is $\pm\frac{N}{2}$, which is the Nyquist condition. Hence the allowable frequencies are:

$$\left\{\left(-\frac{N}{2}+1\right), \cdots, \frac{N}{2}\right\}. \tag{3.8}$$

Consequently, we have:

$$\begin{aligned} F(k) &\approx \int_{x_{\min}}^{x_{\max}} e^{-i2\pi kx} f(x)dx \\ &\approx \sum_{m=0}^{N-1} e^{-i2\pi k x_m} f(x_m)\Delta x + O(\Delta x^2). \end{aligned} \tag{3.9}$$

That is, we are approximating the integral via trapezoidal rule sums. The error of this approximation will be $O(\Delta x^2)$. For $k_{-\frac{N}{2}+1}, \cdots, k_{\frac{N}{2}}$, we then have:

$$\begin{aligned} F_n = F(k_n) &\approx \sum_{m=0}^{N-1} e^{-i2\pi k_n x_m} f(x_m)\Delta x \\ &= \Delta x \sum_{m=0}^{N-1} e^{-i2\pi \frac{n}{\hat{x}}(x_{\min} + m\frac{\hat{x}}{N})} f(x_m) \\ &= \Delta x e^{-i2\pi \frac{n x_{\min}}{\hat{x}}} \sum_{m=0}^{N-1} e^{-i2\pi \frac{nm}{N}} f(x_m) \\ &= \hat{x} e^{-i2\pi \frac{n x_{\min}}{\hat{x}}} \frac{1}{N} \sum_{m=0}^{N-1} e^{-i2\pi \frac{nm}{N}} f(x_m). \end{aligned} \tag{3.10}$$

If we define:

$$\alpha_n := \hat{x} e^{-i2\pi \frac{n x_{\min}}{\hat{x}}}, \tag{3.11}$$

we then have:

$$F_n = F(k_n) \approx \alpha_n \hat{F}_n, \tag{3.12}$$

where $\hat{F}_0, \hat{F}_1, \cdots, \hat{F}_{N-1}$ are the discrete Fourier transforms of $f_0, f_1, \cdots, f_{N-1}$. Figure (3.1) shows the approximation for the DFT. For the inverse continuous Fourier transform (ICFT), we can approximate this by:

$$\begin{aligned} f(x) &= \int_{-\infty}^{\infty} e^{i2\pi kx} F(k) dk \\ &\approx \int_{-\frac{N}{2}}^{\frac{N}{2}} e^{i2\pi kx} F(k) dk \\ &\approx \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} e^{i2\pi k_n x} F(k_n) \Delta k + O(\Delta k^2), \end{aligned} \tag{3.13}$$

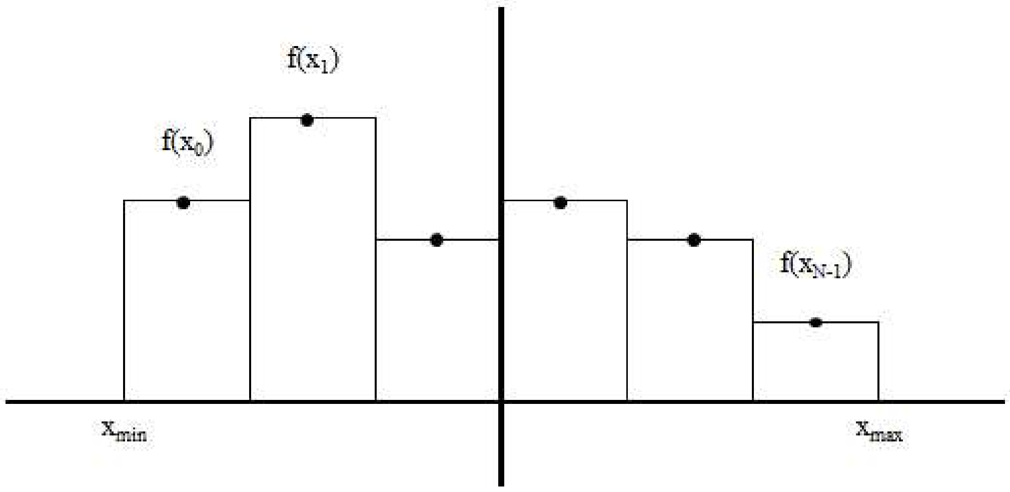

Figure 3.1: Approximation for the discrete Fourier transform.

where $k_n = n \cdot \Delta k$ and $\Delta k = \frac{1}{\hat{x}}$. In this case:

$$\begin{aligned}
f_m = f(x_m) &\approx \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} e^{i2\pi k_n x_m} F(k_n) \frac{1}{\hat{x}} \\
&= \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} e^{i2\pi k_n (x_{\min}+m\Delta x)} F(k_n) \frac{1}{\hat{x}} \\
&= \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} \frac{e^{i2\pi \frac{n}{\hat{x}} x_{\min}}}{\hat{x}} e^{i2\pi \frac{nm}{N}} F(k_n) \qquad (3.14) \\
&= \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} \frac{1}{\alpha_n} e^{i2\pi \frac{nm}{N}} F(k_n) \\
&= \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} e^{i2\pi \frac{mn}{N}} \hat{F}_n.
\end{aligned}$$

Here $f_0, f_1, \cdots, f_{N-1}$ are the inverse discrete Fourier transforms of $\hat{F}_{-\frac{N}{2}+1}, \hat{F}_{-\frac{N}{2}+2} \cdots, \hat{F}_{\frac{N}{2}}$.

Note that the function $\alpha_n$ has cancelled in equation (3.14) and we need only deal with $\hat{F}_n$. Figure (3.2) shows the approximation for IDFT.

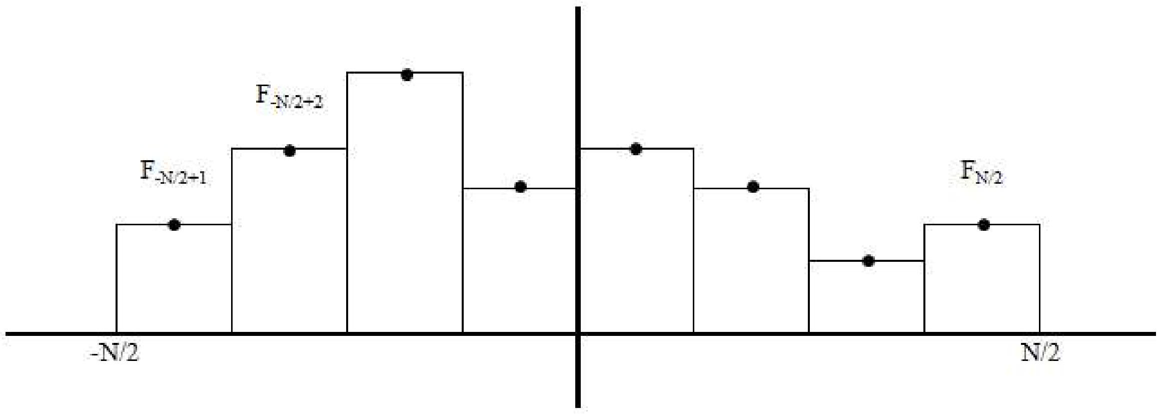

Figure 3.2: Approximation for the inverse discrete Fourier transform.

## 3.2 FST Algorithm for Options under the BSM Model

Given a partial differential equation in the form of equation (2.3), we can price the option via solving the partial differential equation by the Fourier space time-stepping method. First, we need to apply a logarithm transform to the asset price $S$, so we define:

$$\begin{aligned} x &:= \log S, \\ v(x,\tau) &:= \mathcal{V}(S,\tau). \end{aligned} \tag{3.15}$$

Then we can rewrite equation (2.3) of the BSM model as a partial differential equation that describes $v(x,\tau)$:

$$v_\tau = \frac{1}{2}\sigma^2 v_{xx} + \left(r - \frac{1}{2}\sigma^2\right) v_x - rv. \tag{3.16}$$

We apply the continuous Fourier transform $\mathcal{F}$, a linear operator, to the partial differential equation (3.16):

$$\mathcal{F}\left[v_\tau\right](k) = \mathcal{F}\left[\frac{1}{2}\sigma^2 v_{xx}\right](k) + \mathcal{F}\left[\left(r - \frac{1}{2}\sigma^2\right) v_x\right](k) - \mathcal{F}\left[rv\right](k), \tag{3.17}$$

where the transform variable $k$ represents the frequency. By applying properties of the Fourier transform (3.3) and (3.4), we can simplify equation (3.17) as following:

$$\frac{\partial}{\partial \tau}\mathcal{F}\left[v\right](k) = \frac{1}{2}\sigma^2(2\pi i k)^2\mathcal{F}\left[v\right](k) + \left(r - \frac{1}{2}\sigma^2\right)(2\pi i k)\mathcal{F}\left[v\right](k) - r\mathcal{F}\left[v\right](k). \tag{3.18}$$

To simplify equation (3.18), we define:

$$V(k) := \mathcal{F}\left[v\right](k). \tag{3.19}$$

Then equation (3.18) can be written as:

$$V_\tau(k) = \Psi(k) \cdot V(k), \tag{3.20}$$

where the function $\Psi(k)$ is defined by:

$$\Psi(k) := \frac{1}{2}\sigma^2(2\pi i k)^2 + \left(r - \frac{1}{2}\sigma^2\right)(2\pi i k) - r. \tag{3.21}$$

Note that the function $\Psi(k)$ is called the characteristic exponent. If we use a linear operator $\mathcal{L}$ to represent the right hand side of equation (3.16) as:

$$\mathcal{L}v = \frac{1}{2}\sigma^2 v_{xx} + \left(r - \frac{1}{2}\sigma^2\right) v_x - rv, \tag{3.22}$$

then $\Psi(k)$ can be defined by:

$$\mathcal{F}[\mathcal{L}v](k) = \Psi(k) \cdot V(k). \tag{3.23}$$

Hence, by applying the Fourier transform and some of its properties, we can convert the partial differential equation (3.16) in real space into an ordinary differential equation (3.20) in Fourier space. For the BSM model (2.3), the ordinary differential equation (3.20) can be solved analytically:

$$V(k,\tau+\Delta\tau)=e^{\Delta\tau\cdot\Psi(k)}\cdot V(k,\tau). \tag{3.24}$$

Here we use the notation $V(k,\tau+\Delta\tau)$ for the value in Fourier space at time $\tau+\Delta\tau$. Then the CFT of $v(x,\tau)$ can be approximated by DFT as:

$$\begin{aligned}\mathcal{F}[v(x,\tau)](k)&\approx\alpha_n\cdot\hat{V}(k_n,\tau)\\&=\alpha_n\cdot DFT[v(x_m,\tau)],\end{aligned} \tag{3.25}$$

where:

$$\begin{aligned}x_m&=x_{\min}+\frac{m}{x_{\max}-x_{\min}}\ ;\ (m=0,1,\cdots,N-1),\\k_n&=\frac{n}{x_{\max}-x_{\min}}\ ;\ \left(n=-\frac{N}{2}+1,\cdots,\frac{N}{2}\right).\end{aligned} \tag{3.26}$$

Hence the discrete form of equation (3.24) is:

$$\begin{aligned}v(x_m,\tau+\Delta\tau)&=IDFT[\alpha_n^{-1}\cdot V(k_n,\tau+\Delta\tau)]\\&=IDFT[\alpha_n^{-1}\cdot e^{\Delta\tau\cdot\Psi(\frac{n}{\hat{x}})}\cdot\alpha_n\cdot DFT[v(x_m,\tau)]]\\&=IDFT[e^{\Delta\tau\cdot\Psi(\frac{n}{\hat{x}})}\cdot DFT[v(x_m,\tau)]]\\&=IDFT[e^{\Delta\tau\cdot\Psi(k_n)}\cdot DFT[v(x_m,\tau)]].\end{aligned} \tag{3.27}$$

Note by convention, when we carry out the $DFT[v(x_m,\tau)],(m=0,1,\cdots,N-1)$, the algorithm generates:

$$\hat{V}(k_n,\tau)\ ;\ (n=0,1,\cdots,N-1), \tag{3.28}$$

where:

$$k_n=0,\frac{1}{\hat{x}},\cdots,\frac{\frac{N}{2}}{\hat{x}},\frac{-\frac{N}{2}+1}{\hat{x}},\cdots,\frac{-1}{\hat{x}}. \tag{3.29}$$

Hence, to form:

$$e^{\Delta\tau\cdot\Psi(k_n)}\hat{V}(k_n,\tau)\ ;\ (n=0,1,\cdots,N-1), \tag{3.30}$$

one is actually doing:

$$e^{\Delta\tau\cdot\Psi(k'_n)}\hat{V}(k'_n,\tau)\ ;\ (n=0,1,\cdots,N-1), \tag{3.31}$$

where $\hat{V}(k'_0,\tau),\hat{V}(k'_1,\tau),\cdots,\hat{V}(k'_{N-1},\tau)$ are the output of the DFT and where:

$$k'_n=\begin{cases}\frac{n}{\hat{x}}; & n=0,\cdots,\frac{N}{2},\\ \frac{n-N}{\hat{x}}; & n=\frac{N}{2}+1,\cdots,N-1.\end{cases} \tag{3.32}$$

Hence, for a single asset European option under the BSM model, since the ODE can be solved analytically, the option value can be obtained within one timestep. We first apply the Fourier transform, then solve the ODE in Fourier space, and then apply the inverse Fourier transform to convert the solution into real space. As for American options, we need to divide the total time into several time intervals, apply the FST method for one timestep, add optimal exercise conditions and move to the next timestep.

### 3.2.1 Fully Implicit and Crank-Nicolson

In order to extend the FST method to commodity models, we will approximate the time derivative by the values in two timesteps. Recall the partial differential equation that describes $v(x,\tau)$ under the BSM model:

$$v_\tau=\frac{1}{2}\sigma^2v_{xx}+\left(r-\frac{1}{2}\sigma^2\right)v_x-rv. \tag{3.33}$$

To simplify, we use a linear operator $\mathcal{L}$ to represent the right side:

$$\mathcal{L}v:=\frac{1}{2}\sigma^2v_{xx}+\left(r-\frac{1}{2}\sigma^2\right)v_x-rv. \tag{3.34}$$

Then we use the fully implicit scheme to approximate the derivative $v_\tau$ as:

$$[v_\tau]_i^{n+1}=\frac{v_i^{n+1}-v_i^n}{\Delta\tau}=[\mathcal{L}v]_i^{n+1}. \tag{3.35}$$

We then apply the Fourier transform to equation (3.35) and get:

$$\frac{V^{n+1}(k) - V^n(k)}{\Delta\tau} = \Psi(k) \cdot V^{n+1}(k). \tag{3.36}$$

Here we use the notation $V^n(k)$ for the value in Fourier space at timestep $\tau = \tau^n$. Note here $\Psi(k)$ is the characteristic exponent defined the term $\mathcal{L}$ i.e. $\mathcal{F}[\mathcal{L}v](k) = \Psi(k) \cdot V(k)$. Hence, the value at the next timestep $v^{n+1}(x)$ can be expressed as:

$$v^{n+1}(x) = \mathcal{F}^{-1}\left[\frac{\mathcal{F}[v^n(x)]}{1 - \Delta\tau \cdot \Psi(k)}\right]. \tag{3.37}$$

Since the fully implicit is a scheme of first order convergence in time, we can also use the Crank-Nicolson scheme to obtain second order convergence. Here we approximate the derivative $v_\tau$ as:

$$[v_\tau]_i^{n+1} = \frac{v_i^{n+1} - v_i^n}{\Delta\tau} = \frac{1}{2}\left([\mathcal{L}v]_i^{n+1} + [\mathcal{L}v]_i^n\right). \tag{3.38}$$

Following the same process as the fully implicit scheme, we apply the Fourier transform to equation (3.38):

$$\frac{V^{n+1}(k) - V^n(k)}{\Delta\tau} = \frac{1}{2}\left[\Psi(k) \cdot V^{n+1}(k) + \Psi(k) \cdot V^n(k)\right]. \tag{3.39}$$

Hence, the value at the next timestep $v^{n+1}(x)$ can be expressed as:

$$v^{n+1}(x) = \mathcal{F}^{-1}\left[\frac{\mathcal{F}[v^n(x)] \cdot (2 + \Delta\tau \cdot \Psi(k))}{2 - \Delta\tau \cdot \Psi(k)}\right]. \tag{3.40}$$

Note that to obtain second order convergence for Crank-Nicolson, we need to apply fully implicit for the first two timesteps. This is known as Rannacher smoothing [10]. Algorithms (1) and (2) show the processes of using the FST method with the the fully implicit and Crank-Nicolson methods to price a single asset option under the BSM model.

**Algorithm 1:** FST with fully implicit for single asset option under BSM model.

```
Data: $S, K, r, T, \sigma, N, m$
Result: $V$
$\mathbf{x} \leftarrow (x_{\min}, x_{\min} + \frac{x_{\max}-x_{\min}}{N}, \cdots, x_{\min} + (N-1)\frac{x_{\max}-x_{\min}}{N})$;
$\mathbf{k} \leftarrow \frac{1}{x_{\max}-x_{\min}} \cdot (0, 1, \cdots, \frac{N}{2}, -\frac{N}{2}+1, \cdots, -1)$;
$\mathbf{p} \leftarrow \max(S \cdot \exp(\mathbf{x}) - K, 0)$ (call); $\max(K - S \cdot \exp(\mathbf{x}), 0)$ (put);
for $j \leftarrow 1$ to $N$ do
    $\mathbf{\Psi}_j \leftarrow \frac{1}{2}\sigma^2(2\pi i \mathbf{k}_j)^2 + \left(r - \frac{1}{2}\sigma^2\right)(2\pi i \mathbf{k}_j) - r$;
end
$\Delta\tau \leftarrow T/m$; $\mathbf{v} \leftarrow \mathbf{p}$;
for $t \leftarrow 1$ to $m$ do
    $\bar{\mathbf{v}} \leftarrow DFT[\mathbf{v}]$;
    for $j \leftarrow 1$ to $N$ do
        $\bar{\mathbf{v}}_j \leftarrow \bar{\mathbf{v}}_j/(1 - \Delta\tau \cdot \mathbf{\Psi}_j)$;
    end
    $\mathbf{v} \leftarrow IDFT[\bar{\mathbf{v}}]$;
    if American option then
        $\mathbf{v} \leftarrow \max(\mathbf{v}, \mathbf{p})$;
    end
end
$V \leftarrow$ interpolation result of $\mathbf{v}$ at $x = 0$;
```

**Algorithm 2:** FST with Crank-Nicolson for single asset option under BSM model.

```
Data: $S, K, r, T, \sigma, N, m$
Result: $V$
$\mathbf{x} \leftarrow (x_{\text{min}}, x_{\text{min}} + \frac{x_{\text{max}}-x_{\text{min}}}{N}, \cdots, x_{\text{min}} + (N-1)\frac{x_{\text{max}}-x_{\text{min}}}{N})$;
$\mathbf{k} \leftarrow \frac{1}{x_{\text{max}}-x_{\text{min}}} \cdot (0, 1, \cdots, \frac{N}{2}, -\frac{N}{2}+1, \cdots, -1)$;
$\mathbf{p} \leftarrow \max(S \cdot \exp(\mathbf{x}) - K, 0)$ (call); $\max(K - S \cdot \exp(\mathbf{x}), 0)$ (put);
for $j \leftarrow 1$ to $N$ do
    $\mathbf{\Psi}_j \leftarrow \frac{1}{2}\sigma^2(2\pi i \mathbf{k}_j)^2 + \left(r - \frac{1}{2}\sigma^2\right)(2\pi i \mathbf{k}_j) - r$;
end
$\Delta\tau \leftarrow T/m$; $\mathbf{v} \leftarrow \mathbf{p}$;
for $t \leftarrow 1$ to $2$ do
    $\bar{\mathbf{v}} \leftarrow DFT[\mathbf{v}]$;
    for $j \leftarrow 1$ to $N$ do
        $\bar{\mathbf{v}}_j \leftarrow \bar{\mathbf{v}}_j/(1 - \Delta\tau \cdot \mathbf{\Psi}_j)$;
    end
    $\mathbf{v} \leftarrow IDFT[\bar{\mathbf{v}}]$;
    if American option then
        $\mathbf{v} \leftarrow \max(\mathbf{v}, \mathbf{p})$;
    end
end
for $t \leftarrow 3$ to $m$ do
    $\bar{\mathbf{v}} \leftarrow DFT[\mathbf{v}]$;
    for $j \leftarrow 1$ to $N$ do
        $\bar{\mathbf{v}}_j \leftarrow \bar{\mathbf{v}}_j \cdot (2 + \Delta\tau \cdot \mathbf{\Psi}_j)/(2 - \Delta\tau \cdot \mathbf{\Psi}_j)$;
    end
    $\mathbf{v} \leftarrow IDFT[\bar{\mathbf{v}}]$;
    if American option then
        $\mathbf{v} \leftarrow \max(\mathbf{v}, \mathbf{p})$;
    end
end
$V \leftarrow$ interpolation result of $\mathbf{v}$ at $x = 0$;
```

### 3.2.2 FST for European Option under the BSM Model

For a European option with parameters in Table (3.1), the FST method with fully implicit gives first order convergence, given by the results in Table (3.2). As for Crank-Nicolson with Rannacher smoothing, Table (3.3) shows second order convergence. Both are expected since fully implicit is a first order method while Crank-Nicolson is a second order method.

| Parameter | Value |
|---|---|
| $S$ | 100.0 |
| $K$ | 100.0 |
| $r$ | 0.1 |
| $T$ | 1.0 |
| $\sigma$ | 0.2 |

Table 3.1: Single asset option parameters under BSM model.

| Nodes | Timesteps | Value | Change | Ratio[1] | Time(s) |
|---|---|---|---|---|---|
| 1024 | 16 | 3.71934495 | | | 0.0005 |
| 2048 | 32 | 3.73481358 | -0.01546863 | | 0.0038 |
| 4096 | 64 | 3.74373391 | -0.00892033 | 1.73 | 0.0149 |
| 8192 | 128 | 3.74848185 | -0.00474794 | 1.88 | 0.0717 |
| 16384 | 256 | 3.75092669 | -0.00244485 | 1.94 | 0.2269 |

Table 3.2: Single asset European put under BSM model, FST with fully implicit, closed form solution = 3.75341839, first order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| 1024 | 16 | 3.75548948 | | | 0.0005 |
| 2048 | 32 | 3.75390429 | 0.00158519 | | 0.0038 |
| 4096 | 64 | 3.75353537 | 0.00036892 | 4.30 | 0.0151 |
| 8192 | 128 | 3.75344703 | 0.00008833 | 4.18 | 0.0742 |
| 16384 | 256 | 3.75342547 | 0.00002156 | 4.10 | 0.2227 |

Table 3.3: Single asset European put under BSM model, FST with Crank-Nicolson, closed form solution = 3.75341839, second order convergence, including both space and time errors.

[1]If we denote values as $V_1, V_2, \cdots, V_5$, then changes are defined as $V_1 - V_2, \cdots, V_4 - V_5$, and ratios are defined as $\frac{V_1-V_2}{V_2-V_3}, \cdots, \frac{V_3-V_4}{V_4-V_5}$. The binary logarithm of a ratio shows the order of convergence. That is, a ratio of 2 corresponds to first order convergence and a ratio of 4 shows second order convergence.

### 3.2.3 FST for American Option under the BSM Model

For an American option under the BSM model, we can apply the FST method for each timestep, add optimal exercise conditions and move to the next timestep. Algorithms (1) and (2) also show the processes to price an American option under the BSM model by the FST method. The FST method with fully implicit or Crank-Nicolson obtains first order convergence, given by the results in Tables (3.4) and (3.5). The reason that Crank-Nicolson obtains only first order convergence is that the optimal exercise conditions for an American option are added explicitly.

Then we use a finite difference method with the same grid and number of timesteps as the FST method to check and compare the results of the FST method. Table (3.6) shows the finite difference method with fully implicit and Table (3.7) shows the finite difference method with Crank-Nicolson and Rannacher smoothing. Note here the FST method is carried out by using an equally spaced grid of log prices, while the finite difference method is carried out by using an unequally spaced grid of prices, which is obtained by the equally spaced grid of log prices. We add the optimal exercise conditions explicitly for the finite difference method, consistent with the FST method. We can see that with the same number of nodes and timesteps, the solution of the FST method is very close to the solution of the finite difference method.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| 512 | 16 | 4.68092256 | | | 0.0006 |
| 1024 | 32 | 4.73755974 | -0.05663718 | | 0.0022 |
| 2048 | 64 | 4.77319767 | -0.03563793 | 1.59 | 0.0078 |
| 4096 | 128 | 4.79359198 | -0.02039431 | 1.75 | 0.0310 |
| 8192 | 256 | 4.80455089 | -0.01095891 | 1.86 | 0.1529 |

Table 3.4: Single asset American put under BSM model, FST with fully implicit, first order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| 512 | 16 | 4.77748571 | | | 0.0006 |
| 1024 | 32 | 4.78867421 | -0.01118850 | | 0.0021 |
| 2048 | 64 | 4.80081512 | -0.01214091 | 0.92 | 0.0077 |
| 4096 | 128 | 4.80806759 | -0.00725247 | 1.67 | 0.0322 |
| 8192 | 256 | 4.81204960 | -0.00398201 | 1.82 | 0.1533 |

Table 3.5: Single asset American put under BSM model, FST with Crank-Nicolson, first order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s)[2] |
|---|---|---|---|---|---|
| 512 | 16 | 4.66630596 | | | 0.12 |
| 1024 | 32 | 4.73376737 | -0.06746140 | | 0.81 |
| 2048 | 64 | 4.77221789 | -0.03845052 | 1.75 | 8.57 |
| 4096 | 128 | 4.79334001 | -0.02112212 | 1.82 | 84.02 |
| 8192 | 256 | 4.80448634 | -0.01114634 | 1.89 | 865.62 |

Table 3.6: Single asset American put under BSM model, finite difference with fully implicit (not vectorized), first order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| 512 | 16 | 4.76208530 | | | 0.08 |
| 1024 | 32 | 4.78478130 | -0.02269601 | | 0.88 |
| 2048 | 64 | 4.79981199 | -0.01503069 | 1.51 | 11.38 |
| 4096 | 128 | 4.80781121 | -0.00799922 | 1.88 | 100.13 |
| 8192 | 256 | 4.81198397 | -0.00417276 | 1.92 | 962.55 |

Table 3.7: Single asset American put under BSM model, finite difference with Crank-Nicolson (not vectorized), first order convergence, including both space and time errors.

[2]Note that the code for the finite difference method has not been vectorized. Since a vectorized code has better performance in CPU time, it is not reliable to use CPU time to compare the finite difference method with the FST method.

# Chapter 4

# Semi-Lagrangian Method

When we try to use the FST method to solve a partial differential equation derived from the commodity one-factor or two-factor model, we cannot obtain an ordinary differential equation in Fourier space by directly applying the Fourier transform. This is due to the non-constant velocity term in the partial differential equation. Hence, we consider to use the semi-Lagrangian method to deal with this problem. The semi-Lagrangian method is a numerical technique for solving partial differential equations describing the advection-diffusion processes.

## 4.1 Advection-diffusion Equation with Constant Velocity

To introduce the semi-Lagrangian method in a simple context, let us consider a one-dimensional linear advection-diffusion equation of $v(x,\tau)$ with a constant coefficient $\mu$:

$$\frac{\partial v}{\partial \tau} + \mu \cdot \frac{\partial v}{\partial x} = 0, \tag{4.1}$$

with a known initial condition at timestep $\tau = \tau^0$:

$$v(x, \tau^0) = v^0(x). \tag{4.2}$$

By solving equation (4.1) using the method of characteristics, we will have the following solution:

$$v(x,\tau) = v^0(x - \mu\tau). \tag{4.3}$$

Now we consider to solve equation (4.1) on a finite mesh. If we discretize this equation on a grid with space $\Delta x$ and timestep $\Delta\tau$, we can denote the nodes of the grid as:

$$v_i^n = v(x_i, \tau^n), \tag{4.4}$$

where $x_i = i \cdot \Delta x$, $\tau^n = n \cdot \Delta\tau$. If we know the value $v^n(x)$ at timestep $\tau = \tau^n$, we can find the solution $v^{n+1}(x)$ at the next timestep $\tau = \tau^{n+1}$. From equation (4.3), we have:

$$\begin{aligned} v_i^{n+1} = v(x_i, \tau^{n+1}) &= v^0(x_i - \mu\tau^{n+1}) \\ &= v^0(x_i - \mu\Delta\tau - \mu\tau^n) \\ &= v(x_i - \mu\Delta\tau, \tau^n). \end{aligned} \tag{4.5}$$

Hence we can use equation (4.5) to find the value $v^{n+1}(x)$ at timestep $\tau = \tau^{n+1}$ as long as we know the solution $v^n(x)$ at the previous timestep $\tau = \tau^n$. If $v(x_i - \mu\Delta\tau, \tau^n)$ happens to be at a mesh node at timestep $\tau = \tau^n$, we then can let $v(x_i, \tau^{n+1})$ be equal to the value at the node $v(x_i - \mu\Delta\tau, \tau^n)$. But in most cases, this situation will not be satisfied. Hence, in order to get the value $v(x_i, \tau^{n+1})$, we need to use interpolation methods to approximate the value of $v(x_i - \mu\Delta\tau, \tau^n)$ by some nodes at timestep $\tau = \tau^n$. Conversely, we can also get the the value $v^n(x)$ at timestep $\tau = \tau^n$ if we know the solution $v^{n+1}(x)$ at timestep $\tau = \tau^{n+1}$:

$$\begin{aligned} v_i^n = v(x_i, \tau^n) &= v^0(x_i - \mu\tau^n) \\ &= v^0(x_i + \mu\Delta\tau - \mu\tau^{n+1}) \\ &= v(x_i + \mu\Delta\tau, \tau^{n+1}). \end{aligned} \tag{4.6}$$

## 4.2 Advection-diffusion Equation with Non-constant Velocity

Now consider a more general case of advection-diffusion equations with a non-constant velocity $\mu(x,\tau)$:

$$\frac{\partial v}{\partial \tau} + \mu(x,\tau) \cdot \frac{\partial v}{\partial x} = 0, \tag{4.7}$$

with a known initial condition at timestep $\tau = \tau^0$:

$$v(x, \tau^0) = v^0(x). \tag{4.8}$$

To solve equation (4.7) using the method of characteristics, we first need to define a function $X(x, s; \tau), s \in [0, T]$ to be the solution to the following ordinary differential equation:

$$\begin{aligned} \frac{D}{D\tau} X(x, s; \tau) &= \mu(X(x, s; \tau), \tau), \\ X(x, s; s) &= x. \end{aligned} \tag{4.9}$$

Then we consider the Lagrangian derivative $\frac{D}{D\tau} v(X(x, s; \tau), \tau)$, which describes the rate of change in time of $v$ subjected to a space-and-time-dependent velocity field $X(x, s; \tau)$. With the chain rule, we have:

$$\begin{aligned} \frac{D}{D\tau} v(X(x, s; \tau), \tau) &= \left[ \frac{\partial}{\partial \tau} + \frac{D}{D\tau} X(x, s; \tau) \cdot \frac{\partial}{\partial x} \right] v(X(x, s; \tau), \tau) \\ &= \left[ \frac{\partial}{\partial \tau} + \mu(X(x, s; \tau), \tau) \cdot \frac{\partial}{\partial x} \right] v(X(x, s; \tau), \tau) \\ &= 0, \end{aligned} \tag{4.10}$$

and we have:

$$v(X(x, s; \tau), \tau) = \textit{constant}. \tag{4.11}$$

In order to use this for numerical computations, we need to solve equation (4.7) on a finite grid. Thus for the value at timestep $\tau = \tau^{n+1}$, we first define a function $X(x, \tau^{n+1}; \tau)$ to be the solution to the following ODE:

$$\begin{aligned} \frac{D}{D\tau} X(x, \tau^{n+1}; \tau) &= \mu(X(x, \tau^{n+1}; \tau), \tau), \\ X(x, \tau^{n+1}; \tau^{n+1}) &= x. \end{aligned} \tag{4.12}$$

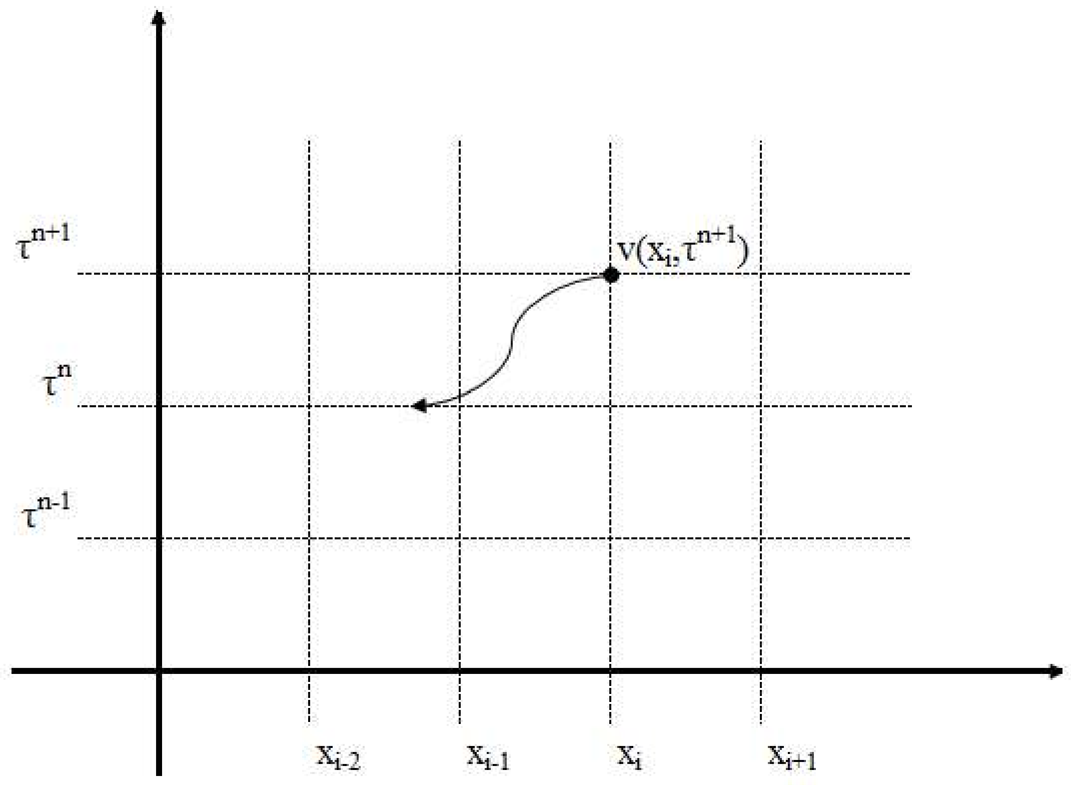

Figure 4.1: Semi-Lagrangian scheme.

By the above discussion, we know $v(X(x, \tau^{n+1}; \tau), \tau) = constant$, we have:

$$\begin{aligned} v(x_i, \tau^{n+1}) &= v(X(x_i, \tau^{n+1}; \tau^{n+1}), \tau^{n+1}) \\ &= v(X(x_i, \tau^{n+1}; \tau^n), \tau^n). \end{aligned} \tag{4.13}$$

Following the same process, for the value at timestep $\tau = \tau^n$, we also have:

$$\begin{aligned} v(x_i, \tau^n) &= v(X(x_i, \tau^n; \tau^n), \tau^n) \\ &= v(X(x_i, \tau^n; \tau^{n+1}), \tau^{n+1}). \end{aligned} \tag{4.14}$$

Figure (4.1) shows the characteristic curve of the semi-Lagrangian scheme. We can use equation (4.13) to find the value $v^{n+1}(x)$ at timestep $\tau = \tau^{n+1}$ as long as we know the solution $v^n(x)$ at timestep $\tau = \tau^n$. Hence, for each timestep $\tau = \tau^n$, we first solve the ordinary differential equation (4.9) to determine the value of $X(x_i, \tau^{n+1}; \tau^n)$. Then we use interpolation methods to approximate this value by some nodes at timestep $\tau = \tau^n$. The result will be the solution $v(x_i, \tau^{n+1})$ at timestep $\tau = \tau^{n+1}$.

## 4.3 Application to the Commodity Price Models

Since some commodity price models, including one-factor and two-factor models, have stochastic drift terms, the partial differential equations for pricing commodity-based options will typically be advection-diffusion equations. We can apply the semi-Lagrangian method to deal with the stochastic drift term, which will be the non-constant velocity term in the advection-diffusion equation. After that, it is convenient to use the Fourier space time-stepping method to obtain the value of the option. Suppose we need to price an option $v(x, \tau)$ with the pricing partial differential equation:

$$\frac{Dv}{D\tau} = \frac{\partial v}{\partial \tau} + \mu(x, \tau) \cdot \frac{\partial v}{\partial x} = \mathcal{L}v, \tag{4.15}$$

where the notation $Dv/D\tau$ is for the Lagrangian derivative, and $\mathcal{L}$ is a linear operator for the rest terms of the partial differential equation. To apply the semi-Lagrangian method, we first define a function $X(x, s; \tau)$ to be the solution to the ordinary differential equation (4.9). Then we use the fully implicit method to deal with the Lagrangian derivative $Dv/D\tau$ by:

$$\left[\frac{Dv}{D\tau}\right]_i^{n+1} = \frac{v_i^{n+1} - v_{i^*}^n}{\Delta\tau} = [\mathcal{L}v]_i^{n+1}. \tag{4.16}$$

From the semi-Lagrangian discussion above, we can use equation (4.14)

$$v_{i^*}^n = v(x_{i^*}, \tau^n) = v(X(x_{i^*}, \tau^n; \tau^{n+1}), \tau^{n+1}),$$

to rewrite equation (4.16) as:

$$\frac{v_i^{n+1} - v(X(x_{i^*}, \tau^n; \tau^{n+1}), \tau^{n+1})}{\Delta\tau} = [\mathcal{L}v]_i^{n+1}. \tag{4.17}$$

To apply the Fourier space time-stepping method, we first define the Fourier transform of $v(X(x_{i^*}, \tau^n; \tau^{n+1}), \tau^{n+1})$ as:

$$\widehat{V^n}(k) := \mathcal{F}\left[v(X(x_{i^*}, \tau^n; \tau^{n+1}), \tau^{n+1})\right](k). \tag{4.18}$$

We apply the Fourier transform to equation (4.17) and obtain the following equation:

$$\frac{V^{n+1}(k) - \widehat{V^n}(k)}{\Delta\tau} = \Psi(k) \cdot V^{n+1}(k), \tag{4.19}$$

where $\Psi(k)$ is the characteristic exponent defined the term $\mathcal{L}$, i.e. $\mathcal{F}[\mathcal{L}v](k) = \Psi(k) \cdot V(k)$. Finally, by simplifying equation (4.19), we have:

$$V^{n+1}(k) = \frac{\widehat{V^n}(k)}{1 - \Delta\tau \cdot \Psi(k)}. \tag{4.20}$$

Since the value $v(X(x_{i^*}, \tau^n; \tau^{n+1}), \tau^{n+1})$ at timestep $\tau = \tau^{n+1}$ can be approximated by the value $v^n(x)$ at timestep $\tau = \tau^n$ via interpolation methods, we can use the following process to solve the value at the next timestep:

$$v^n(x) \xrightarrow{\textit{interpolation}} v(X(x_{i^*}, \tau^n; \tau^{n+1}), \tau^{n+1}) \xrightarrow{\textit{DFT}} \widehat{V^n}(k) \xrightarrow{(4.20)} V^{n+1}(k) \xrightarrow{\textit{IDFT}} v^{n+1}(x). \tag{4.21}$$

Moreover, we can also use the Crank-Nicolson method to deal with the Lagrangian derivative $Dv/D\tau$, then instead of equation (4.16), we have:

$$\left[\frac{Dv}{D\tau}\right]_i^{n+1} = \frac{v_i^{n+1} - v_{i^*}^n}{\Delta\tau} = \frac{1}{2}\left([\mathcal{L}v]_i^{n+1} + [\mathcal{L}v]_{i^*}^n\right). \tag{4.22}$$

From the semi-Lagrangian discussion above, we can use equation (4.14)

$$v_{i^*}^n = v(x_{i^*}, \tau^n) = v(X(x_{i^*}, \tau^n; \tau^{n+1}), \tau^{n+1}),$$

to rewrite equation (4.22) as:

$$\frac{v_i^{n+1} - v(X(x_{i^*}, \tau^n; \tau^{n+1}), \tau^{n+1})}{\Delta\tau} = \frac{1}{2}\left([\mathcal{L}v]_i^{n+1} + [\mathcal{L}v]_{i^*}^n\right). \tag{4.23}$$

For the term $[\mathcal{L}v]_{i^*}^n$, we need to first apply $\mathcal{L}$ to get the value of $[\mathcal{L}v]_i^n$, and then do an interpolation step to get the value of $[\mathcal{L}v]_{i^*}^n$. Also, we define:

$$\begin{aligned}\widehat{V^n}(k) &:= \mathcal{F}\left[v(X(x_{i^*},\tau^n;\tau^{n+1}),\tau^{n+1})\right](k),\\ \widehat{\mathcal{L}V^n}(k) &:= \mathcal{F}\left[[\mathcal{L}v]_{i^*}^n\right](k).\end{aligned} \tag{4.24}$$

Note here $[\mathcal{L}v]_{i^*}^n$ is obtained by:

$$[\mathcal{L}v]_{i^*}^n = \text{interpolation result of } \mathcal{F}^{-1}[\Psi(k)\cdot\mathcal{F}[v_i^n](k)] \text{ at } X(x_{i^*},\tau^n;\tau^{n+1}). \tag{4.25}$$

By the above definition, we apply the Fourier transform to equation (4.23) and get the following equation:

$$\frac{V^{n+1}(k)-\widehat{V^n}(k)}{\Delta\tau} = \frac{1}{2}\left[\Psi(k)\cdot V^{n+1}(k)+\widehat{\mathcal{L}V^n}(k)\right]. \tag{4.26}$$

Hence, following the similar derivation as before, we have:

$$V^{n+1}(k) = \frac{2\cdot\widehat{V^n}(k)+\Delta\tau\cdot\widehat{\mathcal{L}V^n}(k)}{2-\Delta\tau\cdot\Psi(k)}. \tag{4.27}$$

Instead of equation (4.20) for the fully implicit method, we can use equation (4.27) of the Crank-Nicolson method to do the Fourier space time-stepping process. In the implementation of Crank-Nicolson, we need to do the first two timesteps as fully implicit, known as Rannacher smoothing [10]. This gives us second order convergence for Crank-Nicolson.

## 4.4 Interpolation Methods

Interpolation is a numerical method of constructing new data points within the range of a discrete set of known data points. One of the simplest methods is linear interpolation. The interpolated value of the point that does not lie on any existing grid points is determined by the two neighbouring grid points in each respective dimension. Given two grid points $(x_0, y_0)$ and $(x_1, y_1)$, where we assume $x_0 \neq x_1$, we can perform a straight line (first order polynomial) between the two points. Then the value of an interpolated point $(x, y)$ between $(x_0, y_0)$ and $(x_1, y_1)$ is given by:

$$y = y_0 + (x - x_0) \cdot \frac{y_1 - y_0}{x_1 - x_0}. \tag{4.28}$$

The method of linear interpolation is very straightforward and easy to implement. But its disadvantages are obvious. Given $N$ data points $(x_i, y_i)$ where $i = 0, 1, \cdots, N-1$, if $h = \max_{i=1}^{N-1}(x_i - x_{i-1})$, the error of linear interpolation is $O(h^2)$. For the interpolation in a two-dimensional space, we perform linear interpolation first in one direction, and then again in the other direction. This interpolation can be used in the FST method for a two-factor model.

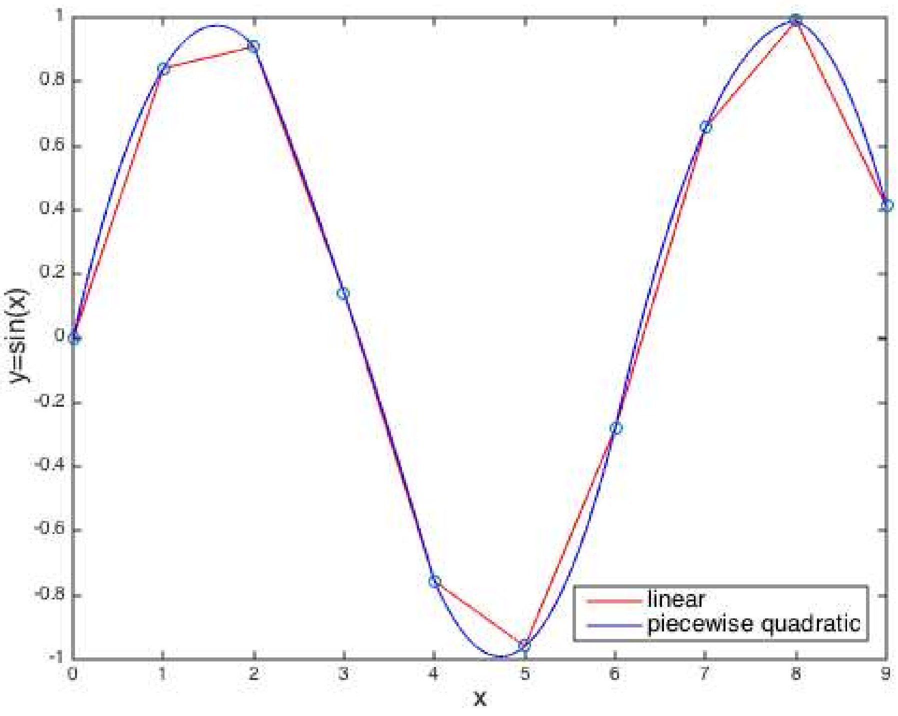

Figure 4.2: Linear interpolation and piecewise quadratic interpolation.

To get a more accurate interpolation result, we can extend linear interpolation to more than two data points. A typical second order interpolation method is piecewise quadratic interpolation. We first divide all the intervals into several pairs with each pair having two closed intervals. Then for the three points in each pair of intervals, we construct a quadratic

Lagrange polynomial. So the global interpolant will be a piecewise quadratic polynomial. Given $N$ data points $(x_i, y_i)$ where $i = 0, 1, \cdots, N-1$, if $h = \max_{i=1}^{N-1}(x_i - x_{i-1})$, the error of piecewise quadratic interpolation is $O(h^3)$. Figure (4.2) shows the linear and piecewise quadratic interpolations for a typical function. Another possibility, which forces more smoothness of the interpolant, is to use a piecewise cubic spline, with local error $O(h^4)$. However, a cubic spline forces continuity of the second derivatives. This property does not hold for an American option.

## 4.5 Error of the Semi-Lagrangian Method

We assume $\Delta x = O(\Delta\tau) = O(h)$ and the local interpolation error is $O(h^q)$. We also assume the global time-stepping error is $O(h^p)$, where $p = 1$ for fully implicit, and $p = 2$ for Crank-Nicolson. The space discretization error is $O(h^2)$ due to the trapezoidal rule equation (3.10). Hence, the final result of the global error for the FST time-stepping with the semi-Lagrangian method is given by:

$$\begin{aligned} \text{global error} &= O\left(\frac{h^q}{\Delta\tau}\right) + O(h^p) + O(h^2) \\ &= O(h^{q-1}) + O(h^p) + O(h^2). \end{aligned} \tag{4.29}$$

From equation (4.29), we can see that we need to use an interpolation method with $q \geq 3$ in order to get second order convergence with Crank-Nicolson, i.e. the piecewise quadratic or higher order interpolation method.

# Chapter 5

# Numerical Results

## 5.1 Commodity Options under a One-factor Model

### 5.1.1 FST Algorithm for Options under a One-factor Model

Under the commodity price one-factor model, we have the options pricing partial differential equation (2.8) that describes $\mathcal{V}(S,\tau)$:

$$\mathcal{V}_\tau = \frac{1}{2}\sigma^2 S^2 \mathcal{V}_{SS} + \kappa(\mu - \lambda - \log S) S \mathcal{V}_S. \tag{5.1}$$

In order to solve the partial differential equation (5.1) by the Fourier space time-stepping method, we first need to apply a logarithm transform to the asset price $S$. Thus we define:

$$\begin{aligned} x &:= \log S, \\ v(x,\tau) &:= \mathcal{V}(S,\tau), \end{aligned} \tag{5.2}$$

and rewrite equation (5.1) as a partial differential equation describing $v(x,\tau)$:

$$v_\tau = \frac{1}{2}\sigma^2 v_{xx} + \left[\kappa(\mu - \lambda - x) - \frac{1}{2}\sigma^2\right] v_x. \tag{5.3}$$

To simplify, we define:

$$a - bx := \kappa(\mu - \lambda - x) - \frac{1}{2}\sigma^2, \tag{5.4}$$

with the values of $a$ and $b$ being:

$$\begin{aligned} a &= \kappa(\mu - \lambda) - \frac{1}{2}\sigma^2, \\ b &= \kappa. \end{aligned} \tag{5.5}$$

To solve equation (5.3) using the method of characteristics, we first need to define a function $X(x, s; \tau)$ to be the solution to the following ordinary differential equation:

$$\begin{aligned} \frac{D}{Ds}X(x, s; \tau) &= \mu(X(x, s; \tau), \tau), \\ X(x, s; s) &= x, \end{aligned} \tag{5.6}$$

where:

$$\mu(x, \tau) = bx - a. \tag{5.7}$$

It is easy to get this solution as:

$$X(x, s; \tau) = xe^{b(s-\tau)} - \frac{a}{b}\left(e^{b(s-\tau)} - 1\right). \tag{5.8}$$

If we rewrite equation (5.3) as:

$$\frac{Dv}{D\tau} = \mathcal{L}v, \tag{5.9}$$

where $\mathcal{L}v$ is given by:

$$\mathcal{L}v = \frac{1}{2}\sigma^2 v_{xx}, \tag{5.10}$$

then we can apply the semi-Lagrangian method. Here the characteristic exponent $\Psi(k)$ is defined by the term $\mathcal{L}v$ and given by:

$$\Psi(k) := \frac{1}{2}\sigma^2(2\pi ik)^2. \tag{5.11}$$

Algorithms (3) and (4) show the processes of using the FST method with fully implicit and Crank-Nicolson to price a single asset option under a one-factor model.

**Algorithm 3:** FST with fully implicit for single asset commodity option under one-factor model.

```
Data: $S, K, \mu, T, \sigma, \kappa, \lambda, N, m$
Result: $V$
$\mathbf{x} \leftarrow (x_{\min}, x_{\min} + \frac{x_{\max}-x_{\min}}{N}, \cdots, x_{\min} + (N-1)\frac{x_{\max}-x_{\min}}{N})$;
$\mathbf{k} \leftarrow \frac{1}{x_{\max}-x_{\min}} \cdot (0, 1, \cdots, \frac{N}{2}, -\frac{N}{2}+1, \cdots, -1)$;
$\mathbf{p} \leftarrow \max(S \cdot \exp(\mathbf{x}) - K, 0)$ (call); $\max(K - S \cdot \exp(\mathbf{x}), 0)$ (put);
for $j \leftarrow 1$ to $N$ do
    $\mathbf{\Psi}_j \leftarrow \frac{1}{2}\sigma^2(2\pi i\mathbf{k}_j)^2$;
end
$\Delta\tau \leftarrow T/m$; $a \leftarrow \kappa(\mu - \lambda) - \frac{1}{2}\sigma^2$; $b \leftarrow \kappa$;
$\widehat{\mathbf{x}} \leftarrow \mathbf{x} \cdot \exp\{-b\Delta\tau\} - \frac{a}{b} \cdot (\exp\{-b\Delta\tau\} - 1)$; $\mathbf{v} \leftarrow \mathbf{p}$;
for $t \leftarrow 1$ to $m$ do
    $\mathbf{v} \leftarrow$ interpolation result of $\mathbf{v}$ at $x = \widehat{\mathbf{x}}$;
    $\bar{\mathbf{v}} \leftarrow DFT[\mathbf{v}]$;
    for $j \leftarrow 1$ to $N$ do
        $\bar{\mathbf{v}}_j \leftarrow \bar{\mathbf{v}}_j/(1 - \Delta\tau \cdot \mathbf{\Psi}_j)$;
    end
    $\mathbf{v} \leftarrow IDFT[\bar{\mathbf{v}}]$;
    if *American option* then
        $\mathbf{v} \leftarrow \max(\mathbf{v}, \mathbf{p})$;
    end
end
$V \leftarrow$ interpolation result of $\mathbf{v}$ at $x = 0$;
```

**Algorithm 4:** FST with Crank-Nicolson for single asset commodity option under one-factor model.

```
Data: $S, K, \mu, T, \sigma, \kappa, \lambda, N, m$
Result: $V$
$\mathbf{x} \leftarrow (x_{\min}, x_{\min} + \frac{x_{\max}-x_{\min}}{N}, \cdots, x_{\min} + (N-1)\frac{x_{\max}-x_{\min}}{N})$;
$\mathbf{k} \leftarrow \frac{1}{x_{\max}-x_{\min}} \cdot (0, 1, \cdots, \frac{N}{2}, -\frac{N}{2}+1, \cdots, -1)$;
$\mathbf{p} \leftarrow \max(S \cdot \exp(\mathbf{x}) - K, 0)$ (call); $\max(K - S \cdot \exp(\mathbf{x}), 0)$ (put);
for $j \leftarrow 1$ to $N$ do
    $\mathbf{\Psi}_j \leftarrow \frac{1}{2}\sigma^2(2\pi i \mathbf{k}_j)^2$;
end
$\Delta\tau \leftarrow T/m$; $a \leftarrow \kappa(\mu - \lambda) - \frac{1}{2}\sigma^2$; $b \leftarrow \kappa$;
$\widehat{\mathbf{x}} \leftarrow \mathbf{x} \cdot \exp\{-b\Delta\tau\} - \frac{a}{b} \cdot (\exp\{-b\Delta\tau\} - 1)$; $\mathbf{v} \leftarrow \mathbf{p}$;
for $t \leftarrow 1$ to $2$ do
    $\mathbf{v} \leftarrow$ interpolation result of $\mathbf{v}$ at $x = \widehat{\mathbf{x}}$;
    $\bar{\mathbf{v}} \leftarrow DFT[\mathbf{v}]$;
    for $j \leftarrow 1$ to $N$ do
        $\bar{\mathbf{v}}_j \leftarrow \bar{\mathbf{v}}_j / (1 - \Delta\tau \cdot \mathbf{\Psi}_j)$;
    end
    $\mathbf{v} \leftarrow IDFT[\bar{\mathbf{v}}]$;
    if American option then
        $\mathbf{v} \leftarrow \max(\mathbf{v}, \mathbf{p})$;
    end
end
for $t \leftarrow 3$ to $m$ do
    $\bar{\mathbf{v}} \leftarrow DFT[\mathbf{v}]$;
    for $j \leftarrow 1$ to $N$ do
        $\bar{\mathbf{L}}_j \leftarrow \bar{\mathbf{v}}_j \cdot \mathbf{\Psi}_j$;
    end
    $\mathbf{L} \leftarrow IDFT[\bar{\mathbf{L}}]$;
    $\mathbf{L} \leftarrow$ interpolation result of $\mathbf{L}$ at $x = \widehat{\mathbf{x}}$; $\mathbf{v} \leftarrow$ interpolation result of $\mathbf{v}$ at $x = \widehat{\mathbf{x}}$;
    $\bar{\mathbf{L}} \leftarrow DFT[\mathbf{L}]$; $\bar{\mathbf{v}} \leftarrow DFT[\mathbf{v}]$;
    for $j \leftarrow 1$ to $N$ do
        $\bar{\mathbf{v}}_j \leftarrow (2 \cdot \bar{\mathbf{v}}_j + \Delta\tau \cdot \bar{\mathbf{L}}_j)/(2 - \Delta\tau \cdot \mathbf{\Psi}_j)$;
    end
    $\mathbf{v} \leftarrow IDFT[\bar{\mathbf{v}}]$;
    if American option then
        $\mathbf{v} \leftarrow \max(\mathbf{v}, \mathbf{p})$;
    end
end
$V \leftarrow$ interpolation result of $\mathbf{v}$ at $x = 0$;
```

### 5.1.2 Monte Carlo for European Option under a One-factor Model

To check the results of our algorithms, we first use Monte Carlo simulations to price a European call under a one-factor model. Table (5.1) shows parameters of the option under a one-factor model. The option values with standard errors of Monte Carlo simulations are listed in Table (5.2).

| Parameter | Value |
|---|---|
| $S$ | 100.0 |
| $K$ | 100.0 |
| $\mu$ | 0.1 |
| $T$ | 1.0 |
| $\sigma$ | 0.1 |
| $\kappa$ | 0.1 |
| $\lambda$ | 0.2 |

Table 5.1: Single asset commodity option parameters under one-factor model.

| Simulations | Timesteps | Value | StdError | Ratio | Time(s) |
|---|---|---|---|---|---|
| 100000 | 1000 | 0.87632401 | 0.01730161 | | 2.41 |
| 200000 | 2000 | 0.87986616 | 0.01225448 | 1.41 | 10.78 |
| 400000 | 4000 | 0.88295630 | 0.00874660 | 1.40 | 42.97 |
| 800000 | 8000 | 0.87461189 | 0.00612923 | 1.43 | 171.57 |
| 1600000 | 16000 | 0.87940762 | 0.00435362 | 1.41 | 700.92 |

Table 5.2: Single asset commodity European call under one-factor model, Monte Carlo.

### 5.1.3 FST for European Option under a One-factor Model

As shown in Tables (5.3), (5.4) and (5.5), for a European option, the fully implicit time-stepping results in first order convergence for linear, piecewise quadratic, and cubic spline interpolations. Since fully implicit is a first order scheme, higher order interpolation method does not result in higher order convergence, as expected from equation (4.29).

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| 1024 | 8 | 0.92407328 | | | 0.00 |
| 2048 | 16 | 0.89925743 | 0.02481585 | | 0.01 |
| 4096 | 32 | 0.88865197 | 0.01060546 | 2.34 | 0.02 |
| 8192 | 64 | 0.88382188 | 0.00483009 | 2.20 | 0.05 |
| 16384 | 128 | 0.88152773 | 0.00229415 | 2.11 | 0.18 |

Table 5.3: Single asset commodity European call under one-factor model, fully implicit with linear interpolation, Monte Carlo solution = 0.87940762 (standard error = 0.00435362), first order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| 1024 | 8 | 0.89335220 | | | 0.11 |
| 2048 | 16 | 0.88479323 | 0.00855897 | | 0.36 |
| 4096 | 32 | 0.88155834 | 0.00323488 | 2.65 | 1.54 |
| 8192 | 64 | 0.88029902 | 0.00125932 | 2.57 | 6.04 |
| 16384 | 128 | 0.87977097 | 0.00052805 | 2.38 | 25.11 |

Table 5.4: Single asset commodity European call under one-factor model, fully implicit with piecewise quadratic interpolation (not vectorized), Monte Carlo solution = 0.87940762 (standard error = 0.00435362), first order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| 1024 | 8 | 0.89567513 | | | 0.00 |
| 2048 | 16 | 0.88514614 | 0.01052899 | | 0.01 |
| 4096 | 32 | 0.88161778 | 0.00352836 | 2.98 | 0.03 |
| 8192 | 64 | 0.88031029 | 0.00130749 | 2.70 | 0.10 |
| 16384 | 128 | 0.87977333 | 0.00053695 | 2.44 | 0.38 |

Table 5.5: Single asset commodity European call under one-factor model, fully implicit with cubic spline interpolation, Monte Carlo solution = 0.87940762 (standard error = 0.00435362), first order convergence, including both space and time errors.

As for Crank-Nicolson, Table (5.6) shows that first order convergence is obtained by linear interpolation. Tables (5.7) and (5.8) show that second order convergence is obtained by piecewise quadratic and cubic spline interpolations, consistent with equation (4.29). Moreover, the error of Crank-Nicolson with higher order interpolation is much less than the error of fully implicit with linear interpolation.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| 1024 | 8 | 0.91761155 | | | 0.00 |
| 2048 | 16 | 0.89594393 | 0.02166762 | | 0.01 |
| 4096 | 32 | 0.88698588 | 0.00895805 | 2.42 | 0.03 |
| 8192 | 64 | 0.88298696 | 0.00399892 | 2.24 | 0.11 |
| 16384 | 128 | 0.88110977 | 0.00187719 | 2.13 | 0.35 |

Table 5.6: Single asset commodity European call under one-factor model, Crank-Nicolson with linear interpolation, Monte Carlo solution = 0.87940762 (standard error = 0.00435362), first order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| 1024 | 8 | 0.88551418 | | | 0.22 |
| 2048 | 16 | 0.88133872 | 0.00417546 | | 0.72 |
| 4096 | 32 | 0.87989938 | 0.00143934 | 2.90 | 2.89 |
| 8192 | 64 | 0.87947228 | 0.00042710 | 3.37 | 11.43 |
| 16384 | 128 | 0.87935589 | 0.00011639 | 3.67 | 46.12 |

Table 5.7: Single asset commodity European call under one-factor model, Crank-Nicolson with piecewise quadratic interpolation (not vectorized), Monte Carlo solution = 0.87940762 (standard error = 0.00435362), second order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| 1024 | 8 | 0.88850829 | | | 0.00 |
| 2048 | 16 | 0.88164771 | 0.00686058 | | 0.02 |
| 4096 | 32 | 0.87990461 | 0.00174309 | 3.94 | 0.05 |
| 8192 | 64 | 0.87946350 | 0.00044112 | 3.95 | 0.21 |
| 16384 | 128 | 0.87935239 | 0.00011110 | 3.97 | 0.78 |

Table 5.8: Single asset commodity European call under one-factor model, Crank-Nicolson with cubic spline interpolation, Monte Carlo solution = 0.87940762 (standard error = 0.00435362), second order convergence, including both space and time errors.

### 5.1.4 FST for American Option under a One-factor Model

As shown in Tables (5.9), (5.10) and (5.11), for an American option, fully implicit obtains first order convergence for linear, piecewise quadratic, and cubic spline interpolations.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| 1024 | 16 | 1.79031816 | | | 0.00 |
| 2048 | 32 | 1.74177375 | 0.04854441 | | 0.01 |
| 4096 | 64 | 1.71742855 | 0.02434519 | 1.99 | 0.03 |
| 8192 | 128 | 1.70551935 | 0.01190920 | 2.04 | 0.11 |
| 16384 | 256 | 1.69950137 | 0.00601798 | 1.98 | 0.35 |

Table 5.9: Single asset commodity American call under one-factor model, fully implicit with linear interpolation, first order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| 1024 | 16 | 1.65717452 | | | 0.52 |
| 2048 | 32 | 1.68093682 | -0.02376229 | | 0.74 |
| 4096 | 64 | 1.68673988 | -0.00580306 | 4.09 | 2.97 |
| 8192 | 128 | 1.69003659 | -0.00329672 | 1.76 | 11.89 |
| 16384 | 256 | 1.69169903 | -0.00166244 | 1.98 | 46.82 |

Table 5.10: Single asset commodity American call under one-factor model, fully implicit with piecewise quadratic interpolation (not vectorized), first order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| 1024 | 16 | 1.67680538 | | | 0.00 |
| 2048 | 32 | 1.68087525 | -0.00406987 | | 0.02 |
| 4096 | 64 | 1.68647075 | -0.00559550 | 0.73 | 0.05 |
| 8192 | 128 | 1.68997803 | -0.00350729 | 1.60 | 0.21 |
| 16384 | 256 | 1.69170835 | -0.00173031 | 2.03 | 0.74 |

Table 5.11: Single asset commodity American call under one-factor model, fully implicit with cubic spline interpolation, first order convergence, including both space and time errors.

As for Crank-Nicolson, since the optimal exercise conditions are added explicitly, first order convergence is obtained for linear, piecewise quadratic, and cubic spline interpolations. The results are shown in Tables (5.12), (5.13) and (5.14).

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| 1024 | 16 | 1.77776410 | | | 0.00 |
| 2048 | 32 | 1.73149500 | 0.04626910 | | 0.02 |
| 4096 | 64 | 1.71125587 | 0.02023912 | 2.29 | 0.06 |
| 8192 | 128 | 1.70203543 | 0.00922044 | 2.20 | 0.21 |
| 16384 | 256 | 1.69762862 | 0.00440680 | 2.09 | 0.71 |

Table 5.12: Single asset commodity American call under one-factor model, Crank-Nicolson with linear interpolation, first order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| 1024 | 16 | 1.65677298 | | | 1.11 |
| 2048 | 32 | 1.67105627 | -0.01428329 | | 1.47 |
| 4096 | 64 | 1.68049809 | -0.00944182 | 1.51 | 5.88 |
| 8192 | 128 | 1.68651207 | -0.00601398 | 1.57 | 23.69 |
| 16384 | 256 | 1.69012193 | -0.00360986 | 1.67 | 94.25 |

Table 5.13: Single asset commodity American call under one-factor model, Crank-Nicolson with piecewise quadratic interpolation (not vectorized), first order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| 1024 | 16 | 1.65820241 | | | 0.01 |
| 2048 | 32 | 1.67097055 | -0.01276813 | | 0.03 |
| 4096 | 64 | 1.68039481 | -0.00942426 | 1.35 | 0.11 |
| 8192 | 128 | 1.68643979 | -0.00604499 | 1.56 | 0.41 |
| 16384 | 256 | 1.68988183 | -0.00344204 | 1.76 | 1.51 |

Table 5.14: Single asset commodity American call under one-factor model, Crank-Nicolson with cubic spline interpolation, first order convergence, including both space and time errors.

## 5.2 Commodity Options under a Two-factor Model

### 5.2.1 FST Algorithm for Options under a Two-factor Model

Under the commodity price two-factor model, we have the options pricing partial differential equation (2.13) that describes $\mathcal{V}(S, \delta, \tau)$:

$$\mathcal{V}_\tau = \frac{1}{2}\sigma_1^2 S^2 \mathcal{V}_{SS} + \sigma_1\sigma_2\rho S\mathcal{V}_{S\delta} + \frac{1}{2}\sigma_2^2\mathcal{V}_{\delta\delta} + (r-\delta)S\mathcal{V}_S + \kappa[(\alpha-\delta)-\lambda]\mathcal{V}_\delta. \tag{5.12}$$

In order to solve the partial differential equation (5.12) by the Fourier space time-stepping method, we first need to apply a logarithm transform to the asset price $S$. Thus we define:

$$\begin{aligned} x &:= \log S, \\ v(x,\delta,\tau) &:= \mathcal{V}(S,\delta,\tau), \end{aligned} \tag{5.13}$$

and rewrite equation (5.12) as a partial differential equation describing $v(x, \delta, \tau)$:

$$v_\tau = \frac{1}{2}\sigma_1^2 v_{xx} + \sigma_1\sigma_2\rho v_{x\delta} + \frac{1}{2}\sigma_2^2 v_{\delta\delta} + \left[(r-\delta) - \frac{1}{2}\sigma_1^2\right] v_x + \kappa[(\alpha-\delta)-\lambda]v_\delta. \tag{5.14}$$

To simplify, we define:

$$\begin{aligned} a_1 - b_1\delta &:= (r-\delta) - \frac{1}{2}\sigma_1^2, \\ a_2 - b_2\delta &:= \kappa[(\alpha-\delta)-\lambda], \end{aligned} \tag{5.15}$$

with the values of $a_1, b_1, a_2, b_2$ being:

$$\begin{aligned} a_1 &= r - \frac{1}{2}\sigma_1^2, \\ b_1 &= 1, \\ a_2 &= \kappa(\alpha-\lambda), \\ b_2 &= \kappa. \end{aligned} \tag{5.16}$$

To solve equation (5.14) using the method of characteristics, we first need to set the vector:

$$\mathbf{x} := \begin{bmatrix} x \\ \delta \end{bmatrix}. \tag{5.17}$$

Then we define a function $X(\mathbf{x}, s; \tau)$ to be the solution to the following ordinary differential equation:

$$\begin{aligned} \frac{D}{Ds} X(\mathbf{x}, s; \tau) &= \mu(X(\mathbf{x}, s; \tau), \tau), \\ X(\mathbf{x}, s; s) &= \mathbf{x}, \end{aligned} \tag{5.18}$$

where:

$$\mu(\mathbf{x}, \tau) = \begin{bmatrix} b_1\delta - a_1 \\ b_2\delta - a_2 \end{bmatrix}. \tag{5.19}$$

The solution is seen to be:

$$X(\mathbf{x}, s; \tau) = \begin{bmatrix} x + \delta(e^{b_1(s-\tau)} - 1) - \frac{a_1}{b_1}(e^{b_1(s-\tau)} - 1) \\ \delta e^{b_2(s-\tau)} - \frac{a_2}{b_2}(e^{b_2(s-\tau)} - 1) \end{bmatrix}. \tag{5.20}$$

If we rewrite equation (5.14) as:

$$\frac{Dv}{D\tau} = \mathcal{L}v, \tag{5.21}$$

where $\mathcal{L}v$ is given by:

$$\mathcal{L}v := \frac{1}{2}\sigma_1^2 v_{xx} + \sigma_1\sigma_2\rho v_{x\delta} + \frac{1}{2}\sigma_2^2 v_{\delta\delta}, \tag{5.22}$$

then we can apply the semi-Lagrangian method. Here the characteristic exponent $\Psi(k_1, k_2)$ is defined by the term $\mathcal{L}v$ and given by:

$$\Psi(k_1, k_2) := \frac{1}{2}\sigma_1^2(2\pi i k_1)^2 + \sigma_1\sigma_2\rho(2\pi i k_1)(2\pi i k_2) + \frac{1}{2}\sigma_2^2(2\pi i k_2)^2. \tag{5.23}$$

Algorithms (5) and (6) show the processes of using the FST method with the fully implicit and Crank-Nicolson methods to price a single asset option under a two-factor model.

**Algorithm 5:** FST with fully implicit for single asset commodity option under two-factor model.

```
Data: $S, \delta, K, r, T, \sigma_1, \sigma_2, \rho, \kappa, \alpha, \lambda, N, m$
Result: $V$
$\mathbf{x1}, \mathbf{x2} \leftarrow (x_{\min}, x_{\min} + \frac{x_{\max}-x_{\min}}{N}, \cdots, x_{\min} + (N-1)\frac{x_{\max}-x_{\min}}{N})$;
$\mathbf{k1}, \mathbf{k2} \leftarrow \frac{1}{x_{\max}-x_{\min}} \cdot (0, 1, \cdots, \frac{N}{2}, -\frac{N}{2}+1, \cdots, -1)$;
for $l \leftarrow 1$ to $N$ do
    $\mathbf{p}_{l:} \leftarrow \max(S \cdot \exp(\mathbf{x1}_l) - K, 0)$ (call); $\max(K - S \cdot \exp(\mathbf{x1}_l), 0)$ (put);
end
for $l \leftarrow 1$ to $N$ do
    for $j \leftarrow 1$ to $N$ do
        $\mathbf{\Psi}_{lj} \leftarrow \frac{1}{2}\sigma_1^2(2\pi i \mathbf{k1}_l)^2 + \frac{1}{2}\sigma_2^2(2\pi i \mathbf{k2}_j)^2 + \sigma_1\sigma_2\rho(2\pi i \mathbf{k1}_l)(2\pi i \mathbf{k2}_j)$;
    end
end
$\Delta\tau \leftarrow T/m$; $a_1 \leftarrow r - \frac{1}{2}\sigma_1^2$; $b_1 \leftarrow 1$; $a_2 \leftarrow \kappa(\alpha - \lambda)$; $b_2 \leftarrow \kappa$;
$\widehat{\mathbf{x2}} \leftarrow \mathbf{x2} \cdot \exp\{-b_2\Delta\tau\} - \frac{a_2}{b_2} \cdot (\exp\{-b_2\Delta\tau\} - 1)$; $\mathbf{v} \leftarrow \mathbf{p}$;
for $t \leftarrow 1$ to $m$ do
    for $j \leftarrow 1$ to $N$ do
        $\widehat{\mathbf{x1}} \leftarrow \mathbf{x1} + \mathbf{x2}_j \cdot (\exp\{-b_1\Delta\tau\} - 1) - \frac{a_1}{b_1} \cdot (\exp\{-b_1\Delta\tau\} - 1)$;
        $\mathbf{v}_{:j} \leftarrow$ interpolation result of $\mathbf{v}$ at $x_1 = \widehat{\mathbf{x1}}, x_2 = \widehat{\mathbf{x2}}_j$;
    end
    $\bar{\mathbf{v}} \leftarrow DFT[\mathbf{v}]$;
    for $l \leftarrow 1$ to $N$ do
        for $j \leftarrow 1$ to $N$ do
            $\bar{\mathbf{v}}_{lj} \leftarrow \bar{\mathbf{v}}_{lj}/(1 - \Delta\tau \cdot \mathbf{\Psi}_{lj})$;
        end
    end
    $\mathbf{v} \leftarrow IDFT[\bar{\mathbf{v}}]$;
    if American option then
        $\mathbf{v} \leftarrow \max(\mathbf{v}, \mathbf{p})$;
    end
end
$V \leftarrow$ interpolation result of $\mathbf{v}$ at $x_1 = 0, x_2 = 0$;
```

**Algorithm 6:** FST with Crank-Nicolson for single asset commodity option under two-factor model.

```
Data: $S, \delta, K, r, T, \sigma_1, \sigma_2, \rho, \kappa, \alpha, \lambda, N, m$
Result: $V$
$\mathbf{x1}, \mathbf{x2} \leftarrow (x_{\min}, x_{\min} + \frac{x_{\max}-x_{\min}}{N}, \cdots, x_{\min} + (N-1)\frac{x_{\max}-x_{\min}}{N})$;
$\mathbf{k1}, \mathbf{k2} \leftarrow \frac{1}{x_{\max}-x_{\min}} \cdot (0, 1, \cdots, \frac{N}{2}, -\frac{N}{2}+1, \cdots, -1)$;
for $l \leftarrow 1$ to $N$ do
    $\mathbf{p}_{l:} \leftarrow \max(S \cdot \exp(\mathbf{x1}_l) - K, 0)$ (call); $\max(K - S \cdot \exp(\mathbf{x1}_l), 0)$ (put);
end
for $l \leftarrow 1$ to $N$ do
    for $j \leftarrow 1$ to $N$ do
        $\mathbf{\Psi}_{lj} \leftarrow \frac{1}{2}\sigma_1^2(2\pi i\mathbf{k1}_l)^2 + \frac{1}{2}\sigma_2^2(2\pi i\mathbf{k2}_j)^2 + \sigma_1\sigma_2\rho(2\pi i\mathbf{k1}_l)(2\pi i\mathbf{k2}_j)$;
    end
end
$\Delta\tau \leftarrow T/m$; $a_1 \leftarrow r - \frac{1}{2}\sigma_1^2$; $b_1 \leftarrow 1$; $a_2 \leftarrow \kappa(\alpha - \lambda)$; $b_2 \leftarrow \kappa$;
$\widehat{\mathbf{x2}} \leftarrow \mathbf{x2} \cdot \exp\{-b_2\Delta\tau\} - \frac{a_2}{b_2} \cdot (\exp\{-b_2\Delta\tau\} - 1)$; $\mathbf{v} \leftarrow \mathbf{p}$;
for $t \leftarrow 1$ to $2$ do
    FST with fully implicit algorithm
end
for $t \leftarrow 3$ to $m$ do
    $\bar{\mathbf{v}} \leftarrow DFT[\mathbf{v}]$;
    for $l \leftarrow 1$ to $N$ do
        for $j \leftarrow 1$ to $N$ do
            $\bar{\mathbf{L}}_{lj} \leftarrow \bar{\mathbf{v}}_{lj} \cdot \mathbf{\Psi}_{lj}$;
        end
    end
    $\mathbf{L} \leftarrow IDFT[\bar{\mathbf{L}}]$;
    for $j \leftarrow 1$ to $N$ do
        $\widehat{\mathbf{x1}} \leftarrow \mathbf{x1} + \mathbf{x2}_j \cdot (\exp\{-b_1\Delta\tau\} - 1) - \frac{a_1}{b_1} \cdot (\exp\{-b_1\Delta\tau\} - 1)$;
        $\mathbf{L}_{:j} \leftarrow$ interpolation result of $\mathbf{L}$ at $x_1 = \widehat{\mathbf{x1}}, x_2 = \widehat{\mathbf{x2}}_j$;
        $\mathbf{v}_{:j} \leftarrow$ interpolation result of $\mathbf{v}$ at $x_1 = \widehat{\mathbf{x1}}, x_2 = \widehat{\mathbf{x2}}_j$;
    end
    $\bar{\mathbf{L}} \leftarrow DFT[\mathbf{L}]$; $\bar{\mathbf{v}} \leftarrow DFT[\mathbf{v}]$;
    for $l \leftarrow 1$ to $N$ do
        for $j \leftarrow 1$ to $N$ do
            $\bar{\mathbf{v}}_{lj} \leftarrow (2 \cdot \bar{\mathbf{v}}_{lj} + \Delta\tau \cdot \bar{\mathbf{L}}_{lj})/(2 - \Delta\tau \cdot \mathbf{\Psi}_{lj})$;
        end
    end
    $\mathbf{v} \leftarrow IDFT[\bar{\mathbf{v}}]$;
    if American option then
        $\mathbf{v} \leftarrow \max(\mathbf{v}, \mathbf{p})$;
    end
end
$V \leftarrow$ interpolation result of $\mathbf{v}$ at $x_1 = 0, x_2 = 0$;
```

### 5.2.2 Monte Carlo for European Option under a Two-factor Model

To check the results of our algorithms, we first use Monte Carlo simulations to price a European put under a two-factor model. Table (5.15) shows parameters of the option under a two-factor model. The option values with standard errors of Monte Carlo simulations are listed in Table (5.16).

| Parameter | Value |
|---|---|
| $S$ | 100.0 |
| $\delta$ | 100.0 |
| $K$ | 100.0 |
| $r$ | 0.1 |
| $T$ | 1.0 |
| $\sigma_1$ | 0.1 |
| $\sigma_2$ | 0.2 |
| $\rho$ | 0.5 |
| $\kappa$ | 0.1 |
| $\alpha$ | 0.1 |
| $\lambda$ | 0.2 |

Table 5.15: Single asset commodity option parameters under two-factor model.

| Simulations | Timesteps | Value | StdError | Ratio | Time(s) |
|---|---|---|---|---|---|
| 100000 | 1000 | 1.10924986 | 0.01914251 | | 6.19 |
| 200000 | 2000 | 1.09865181 | 0.01349634 | 1.42 | 21.31 |
| 400000 | 4000 | 1.10506938 | 0.00956445 | 1.41 | 89.26 |
| 800000 | 8000 | 1.10213963 | 0.00673322 | 1.42 | 353.96 |
| 1600000 | 16000 | 1.10497142 | 0.00477254 | 1.41 | 1509.14 |

Table 5.16: Single asset commodity European put under two-factor model, Monte Carlo.

### 5.2.3 FST for European Option under a Two-factor Model

As shown in Tables (5.17), (5.18) and (5.19), for a European option, fully implicit obtains first order convergence for linear, piecewise quadratic, and cubic spline interpolations. Since fully implicit is a first order scheme, only first order convergence is obtained, regardless of the interpolation methods.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| $256^2$ | 8 | 1.82624602 | | | 0.85 |
| $512^2$ | 16 | 1.42184384 | 0.40440219 | | 3.62 |
| $1024^2$ | 32 | 1.25043458 | 0.17140926 | 2.36 | 17.15 |
| $2048^2$ | 64 | 1.17387687 | 0.07655771 | 2.24 | 94.05 |

Table 5.17: Single asset commodity European put under two-factor model, fully implicit with linear interpolation, Monte Carlo solution = 1.10497142 (standard error = 0.00477254), first order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| $256^2$ | 8 | 1.40466843 | | | 13.27 |
| $512^2$ | 16 | 1.20054602 | 0.20412241 | | 103.85 |
| $1024^2$ | 32 | 1.13850206 | 0.06204396 | 3.29 | 761.74 |
| $2048^2$ | 64 | 1.11743602 | 0.02106603 | 2.95 | 6052.46 |

Table 5.18: Single asset commodity European put under two-factor model, fully implicit with piecewise quadratic interpolation (not vectorized), Monte Carlo solution = 1.10497142 (standard error = 0.00477254), first order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| $256^2$ | 8 | 1.39186878 | | | 0.90 |
| $512^2$ | 16 | 1.19800053 | 0.19386825 | | 4.31 |
| $1024^2$ | 32 | 1.13780988 | 0.06019065 | 3.22 | 24.22 |
| $2048^2$ | 64 | 1.11725556 | 0.02055432 | 2.93 | 145.07 |

Table 5.19: Single asset commodity European put under two-factor model, fully implicit with cubic spline interpolation, Monte Carlo solution = 1.10497142 (standard error = 0.00477254), first order convergence, including both space and time errors.

As for Crank-Nicolson, Table (5.20) shows that first order convergence is obtained by linear interpolation. Tables (5.21) and (5.22) show that second order convergence is obtained by piecewise quadratic and cubic spline interpolations, consistent with equation (4.29). Moreover, the error of Crank-Nicolson with higher order interpolation is much less than the error of fully implicit with linear interpolation.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| $256^2$ | 8 | 1.74336591 | | | 2.82 |
| $512^2$ | 16 | 1.37444169 | 0.36892423 | | 7.49 |
| $1024^2$ | 32 | 1.22567920 | 0.14876248 | 2.48 | 30.53 |
| $2048^2$ | 64 | 1.16130732 | 0.06437188 | 2.31 | 174.09 |

Table 5.20: Single asset commodity European put under two-factor model, Crank-Nicolson with linear interpolation, Monte Carlo solution = 1.10497142 (standard error = 0.00477254), first order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| $256^2$ | 8 | 1.31770472 | | | 33.98 |
| $512^2$ | 16 | 1.15313917 | 0.16456556 | | 243.63 |
| $1024^2$ | 32 | 1.11396369 | 0.03917548 | 4.20 | 1980.87 |
| $2048^2$ | 64 | 1.10497226 | 0.00899142 | 4.36 | 15528.32 |

Table 5.21: Single asset commodity European put under two-factor model, Crank-Nicolson with piecewise quadratic interpolation (not vectorized), Monte Carlo solution = 1.10497142 (standard error = 0.00477254), second order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| $256^2$ | 8 | 1.29593924 | | | 1.95 |
| $512^2$ | 16 | 1.14593922 | 0.15000003 | | 11.33 |
| $1024^2$ | 32 | 1.11172469 | 0.03421453 | 4.38 | 60.00 |
| $2048^2$ | 64 | 1.10433488 | 0.00738981 | 4.63 | 375.47 |

Table 5.22: Single asset commodity European put under two-factor model, Crank-Nicolson with cubic spline interpolation, Monte Carlo solution = 1.10497142 (standard error = 0.00477254), second order convergence, including both space and time errors.

### 5.2.4 FST for American Option under a Two-factor Model

As shown in Tables (5.23), (5.24) and (5.25), for an American option, fully implicit obtains first order convergence for linear, piecewise quadratic, and cubic spline interpolations.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| $1024^2$ | 8 | 2.03947509 | | | 4.18 |
| $2048^2$ | 16 | 2.12046638 | -0.08099130 | | 22.95 |
| $4096^2$ | 32 | 2.16632205 | -0.04585567 | 1.77 | 140.31 |
| $8192^2$ | 64 | 2.18831556 | -0.02199351 | 2.08 | 890.93 |

Table 5.23: Single asset commodity American put under two-factor model, fully implicit with linear interpolation, first order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| $1024^2$ | 8 | 1.98621815 | | | 239.00 |
| $2048^2$ | 16 | 2.09618281 | -0.10996466 | | 1902.34 |
| $4096^2$ | 32 | 2.15425551 | -0.05807270 | 1.89 | 15141.15 |
| $8192^2$ | 64 | 2.18258366 | -0.02832815 | 2.05 | 119615.09 |

Table 5.24: Single asset commodity American put under two-factor model, fully implicit with piecewise quadratic interpolation (not vectorized), first order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| $1024^2$ | 8 | 1.99015104 | | | 7.71 |
| $2048^2$ | 16 | 2.09620372 | -0.10605269 | | 48.63 |
| $4096^2$ | 32 | 2.15427876 | -0.05807503 | 1.83 | 343.02 |
| $8192^2$ | 64 | 2.18231558 | -0.02803683 | 2.07 | 2512.16 |

Table 5.25: Single asset commodity American put under two-factor model, fully implicit with cubic spline interpolation, first order convergence, including both space and time errors.

As for Crank-Nicolson, since the optimal exercise conditions are added explicitly, first order convergence is obtained for linear, piecewise quadratic, and cubic spline interpolations. The results are shown in Tables (5.26), (5.27) and (5.28).

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| $1024^2$ | 8 | 2.04711775 | | | 8.36 |
| $2048^2$ | 16 | 2.11712352 | -0.07000577 | | 45.90 |
| $4096^2$ | 32 | 2.16032182 | -0.04319831 | 1.62 | 280.63 |
| $8192^2$ | 64 | 2.18333167 | -0.02300985 | 1.88 | 1781.85 |

Table 5.26: Single asset commodity American put under two-factor model, Crank-Nicolson with linear interpolation, first order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| $1024^2$ | 8 | 2.02473703 | | | 478.00 |
| $2048^2$ | 16 | 2.09477973 | -0.07004270 | | 3804.67 |
| $4096^2$ | 32 | 2.14887298 | -0.05409325 | 1.29 | 30282.30 |
| $8192^2$ | 64 | 2.17776339 | -0.02889041 | 1.87 | 239230.17 |

Table 5.27: Single asset commodity American put under two-factor model, Crank-Nicolson with piecewise quadratic interpolation (not vectorized), first order convergence, including both space and time errors.

| Nodes | Timesteps | Value | Change | Ratio | Time(s) |
|---|---|---|---|---|---|
| $1024^2$ | 8 | 1.99411970 | | | 15.18 |
| $2048^2$ | 16 | 2.09192908 | -0.09780938 | | 96.37 |
| $4096^2$ | 32 | 2.14800900 | -0.05607992 | 1.74 | 692.98 |
| $8192^2$ | 64 | 2.17727027 | -0.02926127 | 1.92 | 5093.83 |

Table 5.28: Single asset commodity American put under two-factor model, Crank-Nicolson with cubic spline interpolation, first order convergence, including both space and time errors.

# Chapter 6

# Conclusions

In this paper, we present the framework for pricing contingent claims by the Fourier space time-stepping method. The pricing partial differential equation can be converted to an ordinary differential equation in Fourier space. Then the time-stepping can be completed in one timestep between discrete monitoring dates where constraints and conditions are applied. Since the ordinary differential equation can be solved accurately and efficiently in Fourier space, this method can be seen as an useful method for pricing contingent claims, especially for path-dependent options like American and some types of Asian options.

Furthermore, we also extend the Fourier space time-stepping method to some mean-reverting models, which are widely used in commodity prices. The semi-Lagrangian method can be used to deal with the stochastic drift terms in the pricing partial differential equations of the commodity one-factor and two-factor models. We simply need to add an interpolation step in the Fourier space time-stepping method, which gives us quite reasonable results for commodity European and American options. For European options, first order convergence is obtained by fully implicit with linear, piecewise quadratic, and cubic spline interpolations. As for Crank-Nicolson with Rannacher smoothing, linear interpolation gives us first order convergence while piecewise quadratic and cubic spline interpolations give us second order convergence. For American options, as the optimal exercise conditions are added explicitly, for both fully implicit and Crank-Nicolson, first order convergence is obtained.

As discussed in this paper, further research can include

- Use a second order backward differentiation formula (BDF) to approximate the time derivative.

- Extend the Fourier space time-stepping method to other types of contingent claims in commodity markets.
- Use graphics processing units (GPUs) to speed up the computational processes of the fast Fourier transform (FFT) and the inverse fast Fourier transform (IFFT).
- Use graphics processing units (GPUs) to speed up the interpolation operations.

# APPENDICES

# Appendix A

# Wrap-around Error

In the computation of the Fourier space time-stepping method, the Fourier transform assumes a periodic domain implicitly, while the real domain of the log asset price is aperiodic. Consequently, when the Fourier transform is applied, it causes values at the end of the grid to be "wrapped around" to the other side of the grid. The values near the far right end of the domain, the log asset prices $x$ close to $x_{\max}$, will wrap around and produce wrong solutions at the left end of the grid, similarly with the asset prices close to $x_{\min}$.

A simple method to reduce the wrap-around error is to use zero padding. For the domain in real space, we add zeros and the DFT operation is done on a grid with the size twice as the original length. And for the domain in Fourier space, the number of nodes is also doubled. Then during the FST process, after applying the inverse Fourier transform, the added zero value nodes are deleted. If one just tries to solve an option price with an initial asset price $S$, one can use this $S$ as the centre of the grid. Then the wrap-around error will not occur. However, if one tries to solve problems with several initial asset prices at one time, the warp-around error will occur when an initial asset price $S$ is near the end of the grid. In this case, one can set the centre of the grid as the mean value of the maximum initial asset price and the minimum initial asset price. Then with zero padding, all the problems should have reasonable results. Algorithm (7) shows the process to apply zero padding for the FST method. Note that the grid in Algorithm (7) is centred at $S_c$.

With the option parameters shown in Table (A.1), Table (A.2) shows the solutions of the FST method with and without zero padding. Note that the grid here is centred (in the log asset price domain) at $S_c = 100$. The FST solution without zero padding when $S = S_c$ is very close to the closed form solution. But for the values near the end of the grid, the FST solutions without zero padding are quite unreasonable while the FST solutions with zero padding still give us results that are close to the closed form solutions.

**Algorithm 7:** FST with fully implicit for single asset option under BSM model (with zero padding).

```
Data: $S, K, r, T, \sigma, N, m, S_c$
Result: $V$
$\mathbf{x} \leftarrow (x_{\min}, x_{\min} + \frac{x_{\max}-x_{\min}}{N}, \cdots, x_{\min} + (N-1)\frac{x_{\max}-x_{\min}}{N})$;
$\mathbf{k} \leftarrow \frac{1}{2\cdot(x_{\max}-x_{\min})} \cdot (\underbrace{0, 1, \cdots, N, -N+1, \cdots, -1}_{2N})$;
$\mathbf{p}' \leftarrow \max(S_c \cdot \exp(\mathbf{x}) - K, 0)$ (call); $\max(K - S_c \cdot \exp(\mathbf{x}), 0)$ (put);
$\mathbf{p} \leftarrow (\underbrace{0, \cdots, 0}_{\frac{N}{2}}, \underbrace{\mathbf{p}'}_{N}, \underbrace{0, \cdots, 0}_{\frac{N}{2}})$;
for $j \leftarrow 1$ to $2N$ do
    $\mathbf{\Psi}_j \leftarrow \frac{1}{2}\sigma^2(2\pi i\mathbf{k}_j)^2 + \left(r - \frac{1}{2}\sigma^2\right)(2\pi i\mathbf{k}_j) - r$;
end
$\Delta\tau \leftarrow T/m$; $\mathbf{v} \leftarrow \mathbf{p}$;
for $t \leftarrow 1$ to $m$ do
    $\bar{\mathbf{v}} \leftarrow DFT[\mathbf{v}]$;
    for $j \leftarrow 1$ to $N$ do
        $\bar{\mathbf{v}}_j \leftarrow \bar{\mathbf{v}}_j/(1 - \Delta\tau \cdot \mathbf{\Psi}_j)$;
    end
    $\mathbf{v} \leftarrow IDFT[\bar{\mathbf{v}}]$;
    if American option then
        $\mathbf{v} \leftarrow \max(\mathbf{v}, \mathbf{p})$;
    end
end
for $j \leftarrow 1$ to $N$ do
    $\mathbf{v}'_j \leftarrow \mathbf{v}_{j+\frac{N}{2}}$;
end
$V \leftarrow$ interpolation result of $\mathbf{v}'$ at $p' = S$;
```

| Parameter | Value |
|---|---|
| $S$ | 100.0 |
| $K$ | 100.0 |
| $r$ | 0.1 |
| $T$ | 1.0 |
| $\sigma$ | 0.2 |

Table A.1: Single asset option parameters under BSM model.

| S | FST (without padding) | FST (with padding) | Closed form solution |
|---|---|---|---|
| 0.0001 | 138163.00084861 | 0.00000000 | 0.00000000 |
| 0.0010 | 146640.82401928 | 0.00000000 | 0.00000000 |
| 0.0100 | 202198.48160904 | 0.00000000 | 0.00000000 |
| 0.1000 | 60.06458846 | -0.00000000 | 0.00000000 |
| 1.0000 | 0.00000000 | -0.00000000 | 0.00000000 |
| 10.0000 | 0.00000000 | -0.00000000 | 0.00000000 |
| 100.0000 | 13.26967804 | 13.26967804 | 13.26967658 |
| 1000.0000 | 909.51625820 | 909.51625820 | 909.51625820 |
| 10000.0000 | 9909.51625820 | 9909.51625820 | 9909.51625820 |
| 100000.0000 | 99002.05763603 | 99002.05763603 | 99909.51625820 |

Table A.2: Single asset European call under BSM model, wrap-around error.

# References

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

[2] Z. Chen and P. A. Forsyth, *A semi-Lagrangian approach for natural gas storage valuation and optimal operation*, SIAM Journal on Scientific Computing, 30 (2007), pp. 339–368.

[3] Y. d'Halluin, P. A. Forsyth, and G. Labahn, *A semi-Lagrangian approach for American Asian options under jump diffusion*, SIAM Journal on Scientific Computing, 27 (2005), pp. 315–345.

[4] R. Gibson and E. S. Schwartz, *Stochastic convenience yield and the pricing of oil contingent claims*, The Journal of Finance, 45 (1990), pp. 959–976.

[5] T. R. Hurd and Z. Zhou, *A Fourier transform method for spread option pricing*, SIAM Journal on Financial Mathematics, 1 (2010), pp. 142–157.

[6] K. R. Jackson, S. Jaimungal, and V. Surkov, *Fourier space time-stepping for option pricing with Lévy models*, Journal of Computational Finance, 12 (2008), pp. 1–28.

[7] S. Jaimungal and V. Surkov, *Lévy-based cross-commodity models and derivative valuation*, SIAM Journal on Financial Mathematics, 2 (2011), pp. 464–487.

[8] K. R. Miltersen and E. S. Schwartz, *Pricing of options on commodity futures with stochastic term structures of convenience yields and interest rates*, Journal of Financial and Quantitative Analysis, 33 (1998), pp. 33–59.

[9] D. M. Pooley, K. R. Vetzal, and P. A. Forsyth, *Convergence remedies for non-smooth payoffs in option pricing*, Journal of Computational Finance, 6 (2003), pp. 25–40.

[10] R. Rannacher, *Finite element solution of diffusion problems with irregular data*, Numerische Mathematik, 43 (1984), pp. 309–327.

[11] S. A. Ross, *Hedging long run commitments: Exercises in incomplete market pricing*, Economic Notes, 26 (1997), pp. 385–420.

[12] E. S. Schwartz, *The stochastic behavior of commodity prices: Implications for valuation and hedging*, The Journal of Finance, 52 (1997), pp. 923–973.

[13] E. S. Schwartz and J. E. Smith, *Short-term variations and long-term dynamics in commodity prices*, Management Science, 46 (2000), pp. 893–911.

[14] A. Staniforth and J. Côté, *Semi-Lagrangian integration schemes for atmospheric models-a review*, Monthly Weather Review, 119 (1991), pp. 2206–2223.

[15] V. Surkov, *Option pricing using Fourier space time-stepping framework*, PhD thesis, University of Toronto, 2009.

[16] A. Ware, *Accurate semi-Lagrangian time stepping for stochastic optimal control problems with application to the valuation of natural gas storage*, SIAM Journal on Financial Mathematics, 4 (2013), pp. 427–451.

[17] A. F. Ware, *Fast approximate Fourier transforms for irregularly spaced data*, SIAM review, 40 (1998), pp. 838–856.