Numerical Analysis of Two-Asset Options in a Finite Liquidity Framework

McMaster University

Shuai(Kevin) Zhang

Department of Mathematics and Statistics

McMaster University

A thesis submitted for the partial fulfillment of the requirements for

Master of Science

September 3, 2020

Dedicated to my father and mother
Jian-Kang and Xiao Yan
For never giving up on me. I wouldn't of made this far in life without your love and support. I'm forever grateful to you.

Acknowledgements

I would like to thank my supervisor Prof. Traian Pirvu for his constant support throughout my academic career. I have known him since my second year of undergraduate studies, Prof. Pirvu is a mathematician and friend who inspired me. Having the opportunity of working with him was truly an honour. I would also like to thank the professors on this defence committee. In particular, I would like to thank Prof. Shui Feng for being an excellent mentor who supported me with my studies and research. I would also like to thank Prof. Paul McNicholas for being on my committee.

I want to thank Daniel Bezdek for showing me the intricacy of neural networks. In particular, I'm very grateful that he shared some of the tricks and insights he ran into in his profession. I'm thankful to all of my fellow classmates. Especially to Anthony Tan for helping me make corrections and Nik Pocuca for helping me with data generation.

Last but not least, I want to thank my mother and father. My mother Xiao Yan for encouraging and supporting me during all those late nights, always with a smile on her face. My father Jian-Kang for being an endearing role model.

Abstract

In this manuscript, we develop a finite liquidity framework for two-asset markets. In contrast to the standard multi-asset Black-Scholes framework, trading in our market model has a direct impact on the asset's price. The price impact is incorporated into the dynamics of the first asset through a specific trading strategy, as in large trader liquidity models. We adopt Euler-Maruyama and Milstein scheme in the simulation of asset prices. Exchange and Spread option values are numerically estimated by Monte Carlo with the Margrabe option as a controlled variate. The time complexity of these numerical schemes is included. Finally, we provide some deep learning frameworks to implement these pricing models effectively.

Introduction

The Black-Scholes (BS) model was truly a breakthrough for the pricing of single-asset options. It assumes participants operate in a perfect liquid, friction-less and complete market. In practice, one or more of these assumptions are violated. When the liquidity restriction is relaxed, trading will impact the price of the underlying assets. Wilmott (2000) [49] was one of the pioneers of these price impact models. He considered price impacts depending upon different trading strategies such as buy and hold, limit order and portfolio optimization. To account for price impact, Liu and Yong (2005) [33] included an additional term in the asset price stochastic differential equation (SDE). This inclusion indirectly adds a valuation adjustment to the price of the option. Such an adjustment stems from a lack of liquidity, and may be classified as *liquidity valuation adjustment* (LVA). Various non-linear BS-like partial differential equations (PDE), capturing the resulting price impact from trading have been studied [3, 7, 16, 23]. All these models share the similarity of being single-asset LVA models.

Exchange Options provide the utility of exchanging one asset for another. Under the BS assumption for two-asset markets, Margrabe (1978) [34] derived a closed-form solution for the price of Exchange Options. The Exchange Option plays an essential role in currency markets. The *Foreign Exchange* (FX) *Option* is an Exchange Option where the assets are currencies. A common concern is raised when one considers the interaction between liquid and illiquid currencies. A trader might ask, "How reliable is the price of a 3-month European style USD/UAH (Hryvnia, an infrequently traded currency) FX Option?". In this work, we are interested in these types of scenarios. Recent studies on Exchange Options, such as [5, 6, 24, 47], exhibit deviation from the assumptions of BS. The aforementioned studies predominately involve stochastic volatility models.

The *Spread Option* also provides the utility of interchanging one asset for another, with the additional cost of a *strike* price. Under the BS assumption for two-asset markets, there is no closed-form solution for the price of Spread Options. Various groups have worked on numerical methods for Spread Option in the past, some excellent examples can be found in [11,15,29,31,32]. The Spread Option plays an essential role in commodity markets. The *Crush Spread* is a Spread Option between soybean futures and soybean oil. *Crack Spread* is a type of Spread Option specifically designed for crude oil and its refined product.

The *Spark Spread* provides the utility of Spread Option for wholesale electricity price and against the cost of production. Just like the Exchange Option, one of the underlying assets could be illiquid in nature. Pirvu et al. studied Spread Option pricing in the presence of full or partial price impact [1, 38]. Their approach was to implement a finite difference scheme.

In this manuscript, we consider a binary-asset market with a single illiquid asset. Under this consideration, we construct a price impact model, called the *finite liquidity market model* (FLMM). The model is a system of SDEs, one for each asset. The liquid asset is unchanged, the illiquid is modified to incorporate the resulting price impact from trading. Existence and uniqueness conditions on the SDES are established for the FLMM (see B.1). We also discovered a unique risk-neutral measure for FLMM. By replicating a portfolio, We derive the partial differential equation (PDE) characterization of option prices. Finally, we match the solution of the PDE to the risk-neutral price formula by invoking the *discounted Feymann Kac's formula*. This concludes the establishment of model frameworks. Next, we consider a market consisting of market makers, who trade by Delta Hedging. We utilize both the Euler-Maruyama and Milstein method [21, 25] and simulate the FLMM SDEs. We build Monte Carlo (MC) estimators for Exchange and Spread option then explored various reduction techniques. Motivated by [17, 45], we applied these deep learning methods to FLMM and achieves accurate high-speed pricing.

The remainder of the content written in this manuscript is organized in the following sections. Section 1 discusses the basic two-asset model frameworks. In Section 2, we analyze the pricing formula of Exchange and Spread option. Section 3 discusses the FLMM frameworks. In Section 4, we explore pricing methods for Exchange Option under FLMM. In Section 5, we explore pricing methods for Spread Option under FLMM. Section 6 contains the implantation of *Deeply Learning Derivative* and *Deep Galerkin Method* under FLMM. In Section 7, we make some concluding statements for the readers. The last Section is an Appendix containing the formulas and proofs of our results.

Abbreviations

FLMM	Finite liquidity market model
PDE	Partial differential equation
SDE	Stochastic differential equation
BM	Brownian Motion
GBM	Geometric Brownian Motion
RN	Radon Nikodym
BS	Black-Scholes
BSM	Black-Scholes-Merton
E-M	Euler-Maruyama
FBSDE	Forward backward stochastic differential equation
FT	Fourier transform
FFT	Fast Fourier transform
FFN	Feed forward network

Notations

Ω	Sample space
$\mathscr{F}(t)$	Brownian filtration
$\mathbb{P}$	Real world measure
$\mathbb{E}$	Real world expectation
$W(t)$	$\mathbb{P}$-Brownian Motion
$\widetilde{\mathbb{P}}$	Risk-neutral measure
$\widetilde{\mathbb{E}}$	Risk-neutral expectation
$\widetilde{W}(t)$	$\widetilde{\mathbb{P}}$-Brownian Motion
$\big(\Omega, \mathbb{P}, \mathscr{F}(t)\big)$	Filtered probability space of of Brownian Motion
$Z(t)$	Doléans-Dade exponential
$N(x)$	Normal cumulative distribution function
$N'(x)$	Normal probability density function
$S(t)$	Asset process
T	Maturity of the option
τ	Time to maturity of the option
$\otimes$	Tensor product
$\bar{\otimes}$	Outer product
$\mathcal{T}^{(n)}$	$n-th$ dimensional tensor
θ_n	Naive Monte Carlo estimator
θ_a	Antithetic Variate estimator
θ_c	Control Variate estimator
$\mathcal{O}(n)$	Computational complexity

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Basic Two-Assets Market Model

1.1 Market Model

Brownian Motion is a continuous time stochastic process of the Lévy family, it possesses the following properties:

- $W(0) = 0$,
- increments of $W(t)$ are independent,
- $W(t) - W(s) \sim N(0, t-s)$ for all $t > s$,
- $W(t)$ is continuous in t almost surely.

Brownian filtered probability space $\big(\Omega, \mathbb{P}, \{\mathscr{F}(t)\}_{t\geqslant 0}\big)$ is the name given to a space in which BM can be constructed. Unless otherwise specified, the probability spaces in this thesis are assumed to be a peculiar $\big(\Omega, \mathbb{P}, \{\mathscr{F}(t)\}_{t\geqslant 0}\big)$. Brownian Motion are *martingales*, a family of processes that has no tendency to rise nor fall. Brownian path is a particular realization of BM, and it may take on negative values. For these reasons, BM are not suitable to model asset prices. Asset prices should be characterized by processes that are not only positive almost surely, but also posses non-zero drift.

Geometric Brownian Motion is defined to be the solution to the stochastic differential equation:

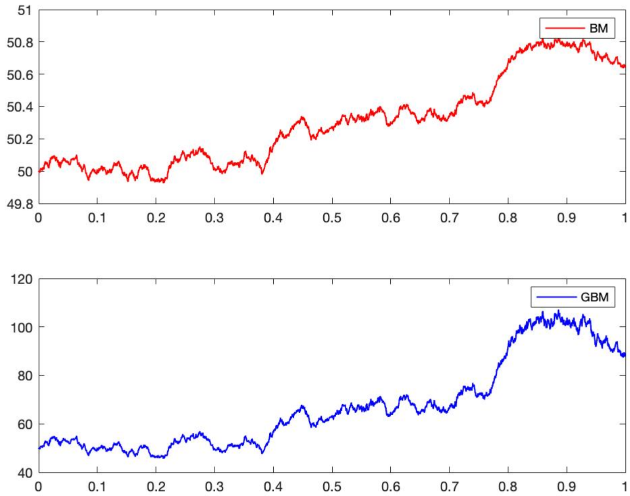
$$\frac{dS(t)}{S(t)} = \mu_1(t)dt + \sigma(t)dW(t).$$

Solving this SDE will yield:

$$S(t) = S(0)\exp\Big\{\int_0^t \big(\mu_1(u) - \frac{1}{2}\sigma^2(u)\big)du + \int_0^t \sigma(u)dW(u)\Big\}.$$

Using GBM to model assets is more suitable because it is positive almost surely and has a tendency to change over time. This is the basis for the model considered by Merton (1971) [35].

Figure 1.1.1: Scaled Brownian Motion with Drift and Geometric Brownian Motion



To price derivatives in multi-asset market scenarios, modelling the behaviour of assets is of uttermost importance.

In Chapter 5 of Shreve's book (2004) [44], he assumes risky assets have the dynamics of Geometric Brownian Motions that are driven by multiple independent Brownian Motions. When there are two risky assets and a stochastic interest rate, the market dynamics can be illustrated as the following:

$$\begin{aligned}
\frac{dS_1(t)}{S_1(t)} &= \mu_1(t)dt + \sigma_1(t)dW_1(t), \\
\frac{dS_2(t)}{S_2(t)} &= \mu_2(t)dt + \sigma_2(t)\rho(t)dW_1(t) + \sigma_2(t)\sqrt{1-\rho(t)^2}dW_2(t), \\
\frac{dD(t)}{D(t)} &= -R(t)dt.
\end{aligned}$$

In this model, $S_1(t)$, $S_2(t)$ and $D(t)$ denote the risky assets and discount process respectively. The Brownian Motions $W_1(t)$ and $W_2(t)$ are independent. The interest rate process $R(t)$ is stochastic in nature. $\rho(t)$ is an instantaneous correlation process. The future price of the assets will be determined by the following factors:

- The drift terms $\mu_1(t)$, $\mu_2(t)$ and $R(t)$ represents the mean rate of return.

- The volatility terms $\sigma_1(t)$ and $\sigma_2(t)$ represents the uncertainty of returns.
- The instantaneous correlation process $\rho(t)$ represents the relationship between the two risky assets. It takes on a value between -1 and 1. It can also be used through Lêvy's theorem to create correlated Brownian Motions.

In 1973, Fischer Black and Myron Scholes presented the Black-Scholes model for pricing European style options [10]. The BS model has the following additional assumptions:

- There exists a risk-neutral measure for the market.
- The market is perfectly liquid, i.e. any buy/sell order will be executed at a linear price (the size of order times price per share).
- Interest rate and volatility terms are constants.

The market dynamics of BSM model is given by the following SDEs:

$$\begin{aligned}
\frac{dS_1(t)}{S_1(t)} &= \mu_1(t)dt + \sigma_1 dW_1(t), \\
\frac{dS_2(t)}{S_2(t)} &= \mu_2(t)dt + \sigma_2\rho dW_1(t) + \sigma_2\sqrt{1-\rho^2}dW_2(t), \\
\frac{dD(t)}{D(t)} &= -rdt.
\end{aligned} \tag{1.1.1}$$

1.2 Two-Dimensional Girsanov's Theorem

In probability theory, the Girsanov theorem describes how the dynamics of stochastic processes change after measure change. The theorem is especially significant in financial mathematics as it allows for the recognition of Brownian Motions after measure change from the real world to a risk-neutral setting.

Theorem 1.2 (Girsanov). *Let $\mathbf{\Theta}(t) = \big(\Theta_1(t), \Theta_2(t)\big)$ be a 2-dimensional adapted process. Define*

$$\begin{aligned}
Z(t) &= \exp\Big\{-\int_0^t \Theta_1(u)dW_1(u) - \int_0^t \Theta_2(u)dW_2(u) - \frac{1}{2}\int_0^t \big(\Theta_1^2(u)+\Theta_2^2(u)\big)du\Big\}, \\
\widetilde{W}(t) &= \begin{bmatrix} \widetilde{W}_1(t) \\ \widetilde{W}_2(t) \end{bmatrix} = \begin{bmatrix} W_1(t) + \int_0^t \Theta_1(u)du \\ W_2(t) + \int_0^t \Theta_2(u)du \end{bmatrix}.
\end{aligned}$$

Set $Z = Z(T)$, it is clear that $\mathbb{E}(Z) = 1$. Then, under the probability measure $\widetilde{\mathbb{P}}$ given by:

$$\widetilde{\mathbb{P}}(A) = \int_A Z(\omega) d\mathbb{P}(\omega) \quad \textit{for all} \quad A \in \mathscr{F},$$

the process $\widetilde{\boldsymbol{W}}(t)$ is a 2-dimensional Brownian Motion.

1.3 Risk Neutral Measure and Market Completeness

In financial mathematics, a risk-neutral measure is a probability measure such that all the discounted asset processes are martingales under this measure. A risk-neutral measure exists if and only if the market is arbitrage-free according to The *first theorem of asset pricing*.

A risk-neutral measure for the BSM market model (1.1.1) can be defined in the following way:

Let

$$\boldsymbol{\Theta}(t) = \begin{bmatrix} \Theta_1(t) \\ \Theta_2(t) \end{bmatrix} = \begin{bmatrix} \frac{\mu_1(t) - r}{\sigma_1} \\ \frac{\mu_2(t) - r}{\sigma_2} \end{bmatrix},$$

Here, $\boldsymbol{\Theta}(t)$ is called the *market price of risk*, or in finance it is often refered to as the *Sharpe ratio*. By Girsanov's Theorem 1.2, there will be a new equivalent probability measure $\widetilde{\mathbb{P}}$ generated by $\boldsymbol{\Theta}(t)$.

Under $\widetilde{\mathbb{P}}$, the market model becomes:

$$\begin{aligned} \frac{dS_1(t)}{S_1(t)} &= rdt + \sigma_1 dW_1(t), \\ \frac{dS_2(t)}{S_2(t)} &= rdt + \sigma_2 \rho dW_1(t) + \sigma_2 \sqrt{1 - \rho^2} dW_2(t), \\ \frac{dD(t)}{D(t)} &= -rdt, \end{aligned} \tag{1.3.1}$$

therefore $\widetilde{\mathbb{P}}$ is a risk-neutral measure since the discounted prices will be $\widetilde{\mathbb{P}}$ martingales.

In this model, the choice of $\widetilde{\mathbb{P}}$ is unique. This leads to the concept of a *complete* market.

Definition 1 (Complete Market). *A market model is complete if every derivative can be hedged.*

Theorem 1.3 (Fundamental Theorems of Asset Pricing).

- *The existence of a risk-neutral measure eliminates arbitrage opportunities, according to The First Theorem of Asset Pricing.*
- *A market model is complete if and only if the risk-neutral measure is unique, according to The Second Theorem of Asset Pricing.*

In fact, the BSM market model (1.1.1) is complete.

1.4 Replicating Portfolio Approach for Pricing Derivative Securities and Derivation of Black-Scholes-Merton PDE

In chapter 4 of Shreve's book (2004) [44], he derives the Black-Scholes-Merton partial differential equation for the price of an European option. Shreve achieved this by determining the initial capital required to completely hedge a short position in the option. This method can also be applied in a multi-asset scenario.

Consider an investor in the BSM market model of section (1.1.1). Suppose the investor has a self-financing portfolio $X(t)$ constructed with the following properties:

1. The portfolio has an account for the first risky asset, with a position of $\Delta_1(t)$ units. The account has a capital gain of $\Delta_1(t)dS_1(t)$.
2. The portfolio has an account for the second risky asset, with a position of $\Delta_2(t)$ units. The account has a capital gain of $\Delta_2(t)dS_2(t)$.
3. The portfolio also has a money market account, with a value of $X(t) - \Delta_1(t)S_1(t) - \Delta_2(t)S_2(t)$. It accrues interest at rate of $r\big(X(t) - \Delta_1(t)S_1(t) - \Delta_2(t)S_2(t)\big)dt$.

The differential of investor's portfolio is:

$$
\begin{aligned}
&dX(t)\\
&= \Delta_1(t)dS_1(t) + \Delta_2(t)dS_2(t) + r\big(X(t) - \Delta_1(t)S_1(t) - \Delta_2(t)S_2(t)\big)dt\\
&= \Delta_1(t)\big(\mu_1(t)S_1(t)dt + \sigma_1 S_1(t)dW_1(t)\big) + \Delta_2(t)\big(\mu_2(t)S_2(t)dt\\
&+ \sigma_2\rho S_2(t)dW_1(t) + \sigma_2\sqrt{1-\rho^2}S_2(t)dW_2(t)\big) + r\big(X(t) - \Delta_1(t)S_1(t) - \Delta_2(t)S_2(t)\big)dt\\
&= rX(t)dt + \Delta_1(t)\big(\mu_1(t) - r\big)S_1(t)dt + \Delta_2(t)\big(\mu_2(t) - r\big)S_2(t)dt\\
&+ \big(\Delta_1(t)\sigma_1 S_1(t) + \Delta_2(t)\sigma_2\rho S_2(t)\big)dW_1(t) + \Delta_2(t)\sigma_2\sqrt{1-\rho^2}S_2(t)dW_2(t).
\end{aligned}
$$

While the differential of the discounted portfolio is:

$$\begin{aligned}
&d\big(D(t)X(t)\big) \\
&= -rD(t)X(t)dt + D(t)dX(t) = D(t)\Big(\Delta_1(t)\big(\mu_1(t)-r\big)S_1(t)dt \\
&+ \Delta_2(t)\big(\mu_2(t)-r\big)S_2(t)dt + \big(\Delta_1(t)\sigma_1 S_1(t) + \Delta_2(t)\sigma_2\rho S_2(t)\big)dW_1(t) \\
&+ \Delta_2(t)\sigma_2\sqrt{1-\rho^2}S_2(t)dW_2(t)\Big).
\end{aligned} \tag{1.4.1}$$

The goal is to price an option whose payoff is $V\big(T, S_1(T), S_2(T)\big)$ at some pre-specified time horizon T, for a given payoff function $V\big(T, s_1, s_2\big)$. We search for the option price at a time $t < T$ of the form $V\big(t, S_1(t), S_2(t)\big)$ for some function $V\big(t, s_1, s_2\big)$.

The differential of that discounted option price is:

$$\begin{aligned}
&d\Big(D(t)V\big(t, S_1(t), S_2(t)\big)\Big) \\
&= -rD(t)V\big(t, S_1(t), S_2(t)\big)dt + D(t)\big(V_t dt + V_{s_1}dS_1(t) + V_{s_2}dS_2(t) \\
&+ \frac{1}{2}V_{s_1s_1}dS_1(t)dS_1(t) + \frac{1}{2}V_{s_2s_2}dS_2(t)dS_2(t) + V_{s_1s_2}dS_1(t)dS_2(t)\big) \\
&= -rD(t)V\big(t, S_1(t), S_2(t)\big)dt + D(t)\Big(V_t dt + V_{s_1}S_1(t)\big(\mu_1(t)dt + \sigma_1 dW_1(t)\big) \\
&+ V_{s_2}S_2(t)\big(\mu_2(t)dt + \sigma_2\rho dW_1(t)\big) + \sigma_2\sqrt{1-\rho^2}dW_2(t)\big) + \frac{1}{2}V_{s_1s_1}\sigma_1^2 S_1^2(t)dt \\
&+ V_{s_1s_2}\rho\sigma_1\sigma_2 S_1(t)S_2(t)dt + \frac{1}{2}V_{s_2s_2}\sigma_2^2 S_2^2(t)dt\Big) \\
&= D(t)\Big\{\Big(-rV\big(t, S_1(t), S_2(t)\big) + V_t + \mu_1(t)S_1(t)V_{s_1} + \mu_2(t)S_2(t)V_{s_2} \\
&+ \frac{1}{2}V_{s_1s_1}\sigma_1^2 S_1^2(t) + \rho\sigma_1\sigma_2 S_1(t)S_2(t)V_{s_1s_2} + \frac{1}{2}V_{s_2s_2}\sigma_2^2 S_2^2(t)\Big)dt \\
&+ \big(\sigma_1 S_1(t)V_{s_1} + \rho\sigma_2 S_2(t)V_{s_2}\big)dW_1(t) + \sqrt{1-\rho^2}\sigma_2 S_2(t)V_{s_2}dW_2(t)\Big\}.
\end{aligned} \tag{1.4.2}$$

Since this market model is complete, then $X(t)$ can be used as a long portfolio to hedge a short position in $V\big(t, S_1(t), S_2(t)\big)$. This is only possible if $X(t) = V\big(t, S_1(t), S_2(t)\big)$ for all $t \in [0, T]$. This leads to the conclusion that the diffusion terms of $d\big(D(t)X(t)\big)$ and $d\Big(D(t)V\big(t, S_1(t), S_2(t)\big)\Big)$ are equal. The *delta-hedge* can be obtained by equating the differential terms from equations (1.4.1) and (1.4.2). In the BSM market model (1.1.1), the delta-hedges are obtained by equating the diffusion terms, whence $\Delta_1(t) = V_{s_1}$ and $\Delta_2(t) = V_{s_2}$.

Next, equate the drift terms from $d\big(D(t)X(t)\big)$ and $d\Big(D(t)V\big(t, S_1(t), S_2(t)\big)\Big)$, the results is:

$$rV(t, s_1, s_2) = V_t + rs_1V_{s_1} + rs_2V_{s_2} + \frac{1}{2}V_{s_1s_1}\sigma_1^2 s_1^2 + \rho\sigma_1\sigma_2 s_1 s_2 V_{s_1s_2} + \frac{1}{2}V_{s_2s_2}\sigma_2^2 s_2^2, \tag{1.4.3}$$

with $0 < s_1, s_2 < \infty$, $0 \leqslant t \leqslant T$ and the terminal condition:

$$V(T, s_1, s_2) = h(s_1, s_2),$$

where $h(s_1, s_2)$ is the pay off function of any Two-Asset option. This is the *Second-order Black-Scholes-Merton PDE.*

For of $|\rho| \leqslant 1$, Equation (1.4.3) is an parabolic PDE. Chapter 4 of Friedman's book (1975) [18] provides the proof for existence and uniqueness for this family of PDEs.

1.5 Risk Neutral Pricing Formula

An investor often wants to evaluate the value of a position prior to maturity. In financial mathematics, the value of an option is a $\mathscr{F}(t)$-adaptive process denote by $V\big(t, S_1(t), S_2(t)\big)$. To understand the logic behind this, let the payoff of an option be $h\big(T, S_1(T), S_2(T)\big) = V\big(T, S_1(T), S_2(T)\big)$, i.e., a $\mathscr{F}(T)$-measurable random variable. Now, consider an investors starting with the portfolio wealth $X(0)$, and adopting some hedging strategy $\mathbf{\Delta}(t)$ representing the vector of the number of shares held in the portfolio at t. In order to hedge a short position in the option, the following needs to hold:

$$X(T) = V\big(T, S_1(T), S_2(T)\big) = h\big(T, S_1(T), S_2(T)\big) \quad \text{almost surely.} \tag{1.5.1}$$

In matter of fact, for the hedging strategy $\mathbf{\Delta}(t)$ to be successful, $X(t)$ has to be equal to $V\big(t, S_1(t), S_2(t)\big)$ for all $t \in [0, T]$.

Since the change in an investor's discounted portfolio only depends on discounted asset prices, then $D(t)X(t)$ is a $\widetilde{\mathbb{P}}$ martingale. Taking the discounted expectation on both side of equation (1.5.1) will give:

$$\widetilde{\mathbb{E}}[D(T)V\big(T, S_1(T), S_2(T)\big)|\mathscr{F}(t)] = \widetilde{\mathbb{E}}[D(T)X(T)|\mathscr{F}(t)] = D(t)X(t),$$

here $X(t)$ is the wealth needed at time t to hedge a short position in the option with the maturity payoff $h\big(T, S_1(T), S_2(T)\big)$. For this reason, $V\big(t, S_1(t), S_2(t)\big)$ is referred to as the value of the option at time t. Naturally, the expression:

$$V(t, s_1, s_2) = \widetilde{\mathbb{E}}[\frac{D(T)}{D(t)}V\big(T, S_1(T), S_2(T)\big)|\mathscr{F}(t)],$$

is the *risk-neutral pricing formula.*

In the BSM market model of (1.3.1), the risk-neutral pricing formula is:

$$V(t, s_1, s_2) = \widetilde{\mathbb{E}}[e^{-r(T-t)}V\big(T, S_1(T), S_2(T)\big)|\mathscr{F}(t)].$$

Computing this conditional expectation will lead to *Black-Scholes-Merton pricing formula* of an European style option.

1.6 Feymann-Kac Formula

Feymann-Kac formula relates the conditional expectation of a Markov process to a PDE. In financial mathematics, it provides a key role by connecting Black-Scholes-Merton PDE to Black-Scholes pricing formula.

Theorem 1.6 (**Discounted Feymann-Kac**). *Let $W_1(t)$ and $W_2(t)$ be two independent Brownian Motions. Consider a 2-dimensional Markov process $\mathbf{X}(t) = \big(X_1(t), X_2(t)\big)$ with corresponding SDEs:*

$$dX_1(t) = \alpha_1\big(t, X_1(t), X_2(t)\big)dt + \phi_{11}\big(t, X_1(t), X_2(t)\big)dW_1(t) + \phi_{12}\big(t, X_1(t), X_2(t)\big)dW_2(t),$$
$$dX_2(t) = \alpha_2\big(t, X_1(t), X_2(t)\big)dt + \phi_{21}\big(t, X_1(t), X_2(t)\big)dW_1(t) + \phi_{22}\big(t, X_1(t), X_2(t)\big)dW_2(t).$$

Given the initial values $X_1(t) = x_1$ and $X_2(t) = x_2$, where $t \in [0, T]$. Let the function $g(t, x_1, x_2)$ be the solution to the PDE:

$$g_t + \alpha_1 g_{x_1} + \alpha_2 g_{x_2} + \frac{1}{2}(\phi_{11}^2 + \phi_{12}^2)g_{x_1x_1} + (\phi_{11}\phi_{21} + \phi_{12}\phi_{22})g_{x_1x_2} + \frac{1}{2}(\phi_{21}^2 + \phi_{22}^2)g_{x_2x_2}$$
$$= rg(t, x_1, x2),$$

with the terminal condition $g(T, x_1, x_2) = f\big(x_1, x_2\big)$ for all x_1 and x_2. Then $g(t, x_1, x_2)$ will satisfy the conditional expectation:

$$g(t, x_1, x_2) = \mathbb{E}[e^{-r(T-t)}f\big(X_1(T), X_2(T)\big)|X_1(t) = x_1, X_2(t) = x_2].$$

Furthermore, $e^{-rt}g\big(t, X_1(t), X_1(t)\big)$ is a martingale.

Let $h(s_1, s_2)$ be the payoff function of an option, given the asset values at maturity are $S_1(T) = s_1$ and $S_2(T) = s_2$. Let $V(t, s_1, s_2)$ be the solution of the Black-Scholes PDE:

$$rV(t, s_1, s_2) = V_t + rs_1V_{s_1} + rs_2V_{s_2} + \frac{1}{2}V_{s_1s_1}\sigma_1^2 s_1^2 + \rho\sigma_1\sigma_2 s_1 s_2 V_{s_1s_2} + \frac{1}{2}V_{s_2s_2}\sigma_2^2 s_2^2,$$

with the terminal condition $V(T, s_1, s_2) = h\big(s_1, s_2\big)$. Then, by the discounted version of Feymann-Kac Theorem 1.6, $V(t, s_1, s_2)$ will satisfy the risk-neutral pricing formula:

$$V(t, s_1, s_2) = \widetilde{\mathbb{E}}[e^{-r(T-t)}h\big(S_1(T), S_2(T)\big)|S_1(t) = s_1, S_2(t) = s_2].$$

Indeed $e^{-rt}V(t, s_1, s_2)$ is a Martingale, that is

$$e^{-rt}V(t, s_1, s_2) = \widetilde{\mathbb{E}}[e^{-rT}V\big(t, S_1(T), S_2(T)\big)|S_1(t) = s_1, S_2(t) = s_2].$$

This shows under market model (1.3.1), the Black-Scholes PDE's characterization of the option price is equivalent to the risk neutral pricing formula. Pricing any Two-Asset Option would be equivalent to solving this PDE.

Chapter 2

Options on Two-Assets

2.1 Exchange option

An exchange option provides the utility of exchanging one asset for another. It is a special case of the spread option when the strike price $K = 0$. Some examples of exchange option are currency swaption, interest rate swaption and commodity swaption. In Chapter 25 of Hull's book (2006) [27], it was mentioned that a stock tender offer can also be seen as an exchange option. Essentially, it provides current shareholders the option to exchange their shares for shares of another company.

2.1.1 Pricing Formula

From the risk-neutral pricing formula, the value of the exchange option is:

$$V\big(t, S_1(t), S_2(t)\big) = \widetilde{\mathbb{E}}\big[e^{-r(T-t)}\big(S_1(T) - S_2(T)\big)\mathbb{1}_{S_1(T)>S_2(T)}|\mathscr{F}(t)\big]. \tag{2.1.1}$$

In 1978, Margrabe [34] computed this conditional expectation in closed form and obtained:

Theorem 2.1.1 (Margrabe's formula).

$$\widetilde{\mathbb{E}}\big[e^{-r(T-t)}\big(S_1(T) - S_2(T)\big)\mathbb{1}_{S_1(T)>S_2(T)}|\mathscr{F}(t)\big] = S_1(t)N(d_+) - S_2(t)N(d_-),$$

where $d_\pm = \frac{\log\left(\frac{S_1(t)}{S_2(t)}\right) \pm \frac{1}{2}\nu^2(T-t)}{\nu\sqrt{T-t}}$, *and* $\nu^2 = \sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho$.

Margrabe was able to achieve this by using the fact that the ratio process $\frac{S_1}{S_2}(t)$ is also a GBM, and cleaver use of Numeraire measures. We will provide a derivation method similar to Margrabe's approach.

Proof: Define two new Numeraire measures $\mathbb{P}_1$ and $\mathbb{P}_2$ with their respective Radon-Nikodym processes of $\widetilde{\mathbb{P}}$ as:

$$\begin{aligned} Z_1(t) &= \frac{e^{-rt}S_1(t)}{S_1(0)} = \exp\Big\{\sigma_1\widetilde{W}_1(t) - \frac{1}{2}\sigma_1^2 t\Big\}, \\ Z_2(t) &= \frac{e^{-rt}S_2(t)}{S_2(0)} = \exp\Big\{\sigma_2[\rho\tilde{W}_1(t) + \sqrt{1-\rho^2}\widetilde{W}_2(t)] - \frac{1}{2}\sigma_2^2 t\Big\}. \end{aligned}$$

Then, their vector valued generator processes are:

$$\mathbf{\Theta}_1(t) = \begin{bmatrix} -\sigma_1 \\ 0 \end{bmatrix}, \quad \mathbf{\Theta}_2(t) = \begin{bmatrix} -\sigma_2\rho \\ -\sigma_2\sqrt{1-\rho^2} \end{bmatrix}.$$

By applying Girsanov's theorem, under the Numeraire measure $\mathbb{P}_1$, the processes

$$\begin{aligned} W_1^{(1)}(t) &= \widetilde{W}_1(t) - \sigma_1 t, \quad \text{and} \\ W_2^{(1)}(t) &= \widetilde{W}_2(t), \end{aligned}$$

are two independent Brownian Motions. Under $\mathbb{P}_2$, the processes

$$\begin{aligned} W_1^{(2)}(t) &= \widetilde{W}_1(t) - \sigma_2\rho t, \quad \text{and} \\ W_2^{(2)}(t) &= \widetilde{W}_2(t) - \sigma_2\sqrt{1-\rho^2}t \end{aligned}$$

are also two independent Brownian Motions in light of Girsanov's theorem. The computation of (2.1.1) will reduce to:

$$\begin{aligned} &V\big(t, S_1(t), S_2(t)\big) \\ &= \widetilde{\mathbb{E}}\big[e^{-r(T-t)}S_1(T)\mathbb{1}_{S_1(T)>S_2(T)}|\mathscr{F}(t)\big] - \widetilde{\mathbb{E}}\big[e^{-r(T-t)}S_2(T)\mathbb{1}_{S_1(T)>S_2(T)}|\mathscr{F}(t)\big] \\ &= \frac{S_1(t)}{Z_1(t)}\widetilde{\mathbb{E}}\big[Z_1(T)\mathbb{1}_{S_1(T)>S_2(T)}|\mathscr{F}(t)\big] - \frac{S_2(t)}{Z_2(t)}\widetilde{\mathbb{E}}\big[Z_2(T)\mathbb{1}_{S_1(T)>S_2(T)}|\mathscr{F}(t)\big] \\ &= S_1(t)\mathbb{E}^{(1)}\big[\mathbb{1}_{S_1(T)>S_2(T)}|\mathscr{F}(t)\big] - S_2(t)\mathbb{E}^{(2)}\big[\mathbb{1}_{S_1(T)>S_2(T)}|\mathscr{F}(t)\big] \\ &= S_1(t)\mathbb{P}^{(1)}\Big(S_1(T) > S_2(T)|\mathscr{F}(t)\Big) - S_2(t)\mathbb{P}^{(2)}\Big(S_1(T) > S_2(T)|\mathscr{F}(t)\Big). \end{aligned} \tag{2.1.2}$$

Under the Numeraire measure $\mathbb{P}_1$, the SDEs of the asset processes will behave in the following manner:

$$\begin{aligned} S_1(t) &= S_1(0)\exp\Big\{\sigma_1 W_1^{(1)}(t) + (r + \frac{1}{2}\sigma_1^2)t\Big\}, \\ S_2(t) &= S_2(0)\exp\Big\{\sigma_2[\rho W_1^{(1)}(t) + \sqrt{1-\rho^2}W_2^{(1)}(t)] + (r - \frac{1}{2}\sigma_2^2 + \sigma_1\sigma_2\rho)t\Big\}. \end{aligned}$$

The first probability term of equation (2.1.2) can be evaluated as:

$$
\begin{aligned}
&\mathbb{P}^{(1)}\Big(S_1(T) > S_2(T)|\mathscr{F}(t)\Big) \\
&= \mathbb{P}^{(1)}\Big(S_2(t)\exp\Big\{\sigma_2[\rho\big(W_1^{(1)}(T) - W_1^{(1)}(t)\big) + \sqrt{1-\rho^2}\big(W_2^{(1)}(T) - W_2^{(1)}(t)\big)] \\
&+ (r - \frac{1}{2}\sigma_2^2 + \sigma_1\sigma_2\rho)(T-t)\Big\} \\
&< S_1(t)\exp\Big\{\sigma_1\big(W_1^{(1)}(T) - W_1^{(1)}(t)\big) + \big(r + \frac{1}{2}\sigma_1^2\big)(T-t)\Big\}|\mathscr{F}(t)\Big) \\
&= \mathbb{P}^{(1)}\Big(\exp\Big\{(\sigma_2\rho - \sigma_1)\big(W_1^{(1)}(T) - W_1^{(1)}(t)\big) + \sigma_2\sqrt{1-\rho^2}\big(W_2^{(1)}(T) - W_2^{(1)}(t)\big)\Big\} \\
&< \frac{S_1(t)}{S_2(t)}\exp\Big\{\frac{1}{2}(\sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho)(T-t)\Big\}|\mathscr{F}(t)\Big) \\
&= \mathbb{P}^{(1)}\Big((\sigma_2\rho - \sigma_1)\big(W_1^{(1)}(T) - W_1^{(1)}(t)\big) + \sigma_2\sqrt{1-\rho^2}\big(W_2^{(1)}(T) - W_2^{(1)}(t)\big) \\
&< \log\big(\frac{S_1(t)}{S_2(t)}\big) + \frac{1}{2}(\sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho)(T-t)|\mathscr{F}(t)\Big).
\end{aligned}
$$

Let

$$
\begin{aligned}
\nu^2 &= \sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho, \quad \text{and} \\
X_1 &= (\sigma_2\rho - \sigma_1)\big(W_1^{(1)}(T) - W_1^{(1)}(t)\big) + \sigma_2\sqrt{1-\rho^2}\big(W_2^{(1)}(T) - W_2^{(1)}(t)\big),
\end{aligned}
$$

then X_1 is Gaussian $N\big(0, \nu^2(T-t)\big)$. Define $Y_1 = \frac{X_1}{\nu\sqrt{T-t}}$, then Y_1 is a standard Gaussian $N\big(0,1\big)$. The first probability term simplifies to:

$$
\mathbb{P}^{(1)}\Big(S_1(T) > S_2(T)|\mathscr{F}(t)\Big) = \mathbb{P}^{(1)}\Big(Y_1 < \frac{\log\big(\frac{S_1(t)}{S_2(t)}\big) + \frac{1}{2}\nu^2(T-t)}{\nu\sqrt{T-t}}\Big) = N(d_+),
$$

where $d_+ = \frac{\log\big(\frac{S_1(t)}{S_2(t)}\big) + \frac{1}{2}\nu^2(T-t)}{\nu\sqrt{T-t}}$.

Under the Numeraire measure $\mathbb{P}_2$, the SDEs of the asset processes will become:

$$
\begin{aligned}
S_1(t) &= S_1(0)\exp\Big\{\sigma_1 W_1^{(2)}(t) + (r - \frac{1}{2}\sigma_1^2 + \sigma_1\sigma_2\rho)t\Big\}, \\
S_2(t) &= S_2(0)\exp\Big\{\sigma_2[\rho W_1^{(2)}(t) + \sqrt{1-\rho^2}W_2^{(2)}(t)] + (r + \frac{1}{2}\sigma_2^2)t\Big\}.
\end{aligned}
$$

The second probability term of equation (2.1.2) can be evaluated as:

$$
\begin{aligned}
&\mathbb{P}^{(2)}\Big(S_1(T) > S_2(T)|\mathscr{F}(t)\Big)\\
&= \mathbb{P}^{(2)}\Big(S_2(t)\exp\Big\{\sigma_2[\rho\big(W_1^{(2)}(T) - W_1^{(2)}(t)\big) + \sqrt{1-\rho^2}\big(W_2^{(2)}(T) - W_2^{(2)}(t)\big)]\\
&+ (r + \frac{1}{2}\sigma_2^2)(T-t)\Big\}\\
&< S_1(t)\exp\Big\{\sigma_1\big(W_1^{(2)}(T) - W_1^{(2)}(t)\big) + \big(r - \frac{1}{2}\sigma_1^2 + \sigma_1\sigma_2\rho\big)(T-t)\Big\}|\mathscr{F}(t)\Big)\\
&= \mathbb{P}^{(2)}\Big(\exp\Big\{(\sigma_2\rho - \sigma_1)\big(W_1^{(2)}(T) - W_1^{(2)}(t)\big) + \sigma_2\sqrt{1-\rho^2}\big(W_2^{(2)}(T) - W_2^{(2)}(t)\big)\Big\}\\
&< \frac{S_1(t)}{S_2(t)}\exp\Big\{-\frac{1}{2}(\sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho)(T-t)\Big\}|\mathscr{F}(t)\Big)\\
&= \mathbb{P}^{(2)}\Big((\sigma_2\rho - \sigma_1)\big(W_1^{(2)}(T) - W_1^{(2)}(t)\big) + \sigma_2\sqrt{1-\rho^2}\big(W_2^{(2)}(T) - W_2^{(2)}(t)\big)\\
&< \log\big(\frac{S_1(t)}{S_2(t)}\big) - \frac{1}{2}(\sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho)(T-t)|\mathscr{F}(t)\Big).
\end{aligned}
$$

Let

$$
\begin{aligned}
\nu^2 &= \sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho, \quad \text{and}\\
X_2 &= (\sigma_2\rho - \sigma_1)\big(W_1^{(2)}(T) - W_1^{(2)}(t)\big) + \sigma_2\sqrt{1-\rho^2}\big(W_2^{(2)}(T) - W_2^{(2)}(t)\big),
\end{aligned}
$$

then X_2 is a Gaussian with $N\big(0, \nu^2(T-t)\big)$. Define $Y_2 = \frac{X_2}{\nu\sqrt{T-t}}$, then Y_2 is a standard Gaussian $N\big(0,1\big)$. The second probability term simplifies to:

$$
\mathbb{P}^{(2)}\Big(S_1(T) > S_2(T)|\mathscr{F}(t)\Big) = \mathbb{P}^{(2)}\Big(Y_2 < \frac{\log\big(\frac{S_1(t)}{S_2(t)}\big) - \frac{1}{2}\nu^2(T-t)}{\nu\sqrt{T-t}}\Big) = N(d_-),
$$

where $d_- = \frac{\log\big(\frac{S_1(t)}{S_2(t)}\big) - \frac{1}{2}\nu^2(T-t)}{\nu\sqrt{T-t}}$.
The resulting option pricing formula is:

$$
\begin{aligned}
\tilde{\mathbb{E}}\big[e^{-r(T-t)}\big(S_1(T) - S_2(T)\big)\mathbb{1}_{S_1(T)>S_2(T)}|\mathscr{F}(t)\big] &= V\big(t, S_1(t), S_2(t)\big)\\
V\big(t, s_1, s_2\big) &= s_1 N(d_+) - s_2 N(d_-),
\end{aligned}
$$

where $d_\pm = \frac{\log\big(\frac{s_1}{s_2}\big) \pm \frac{1}{2}\nu^2(T-t)}{\nu\sqrt{T-t}}$, and $\nu^2 = \sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho$. □

This is the same as Margrabe's formula.

2.1.2 Greeks of Exchange option

In financial mathematics, Greeks are option's price sensitivity to change of its parameters. Greeks are important because it allows portfolios to be protected against specific undesirable exposures. The undesired exposures can be eliminated by the process hedging. Exchange option has many Greeks because its numerous amount of parameters.

Starting with the first-order Greeks, Delta is the option's price change due to price change in the underlying asset. Exchange option has two Deltas, they are defined to be:

$$\Delta_1(t) = \frac{\partial V(t, s_1, s_2)}{\partial s_1}, \quad \Delta_2(t) = \frac{\partial V(t, s_1, s_2)}{\partial s_2}.$$

Theorem 2.2.1 (**Exchange option Delta**).

$$\Delta_1(t) = N(d_+), \quad \Delta_2(t) = -N(d_-).$$

Proof: Please refer to A.1.1 in Appendix 1. □

Deltas have many uses, some examples are Delta-hedging or Delta-spread strategy. Figure 2.1.1 illustrates the behaviour of Δ_1 caused by prices of the two assets at initiation. The option parameters used for all subsequent Greeks(including Delta) are $T = 1$, $\sigma_1 = 0.4$, $\sigma_2 = 0.2$, $\rho = 0.5$ and $r = 0.05$.

Figure 2.1.1: Exchange option Δ_1

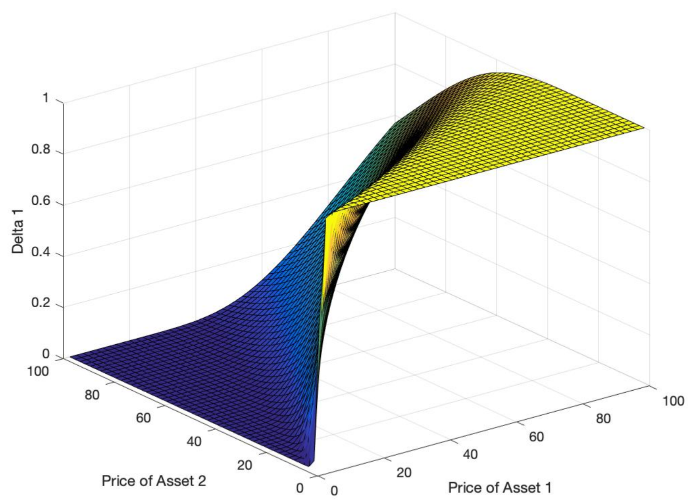
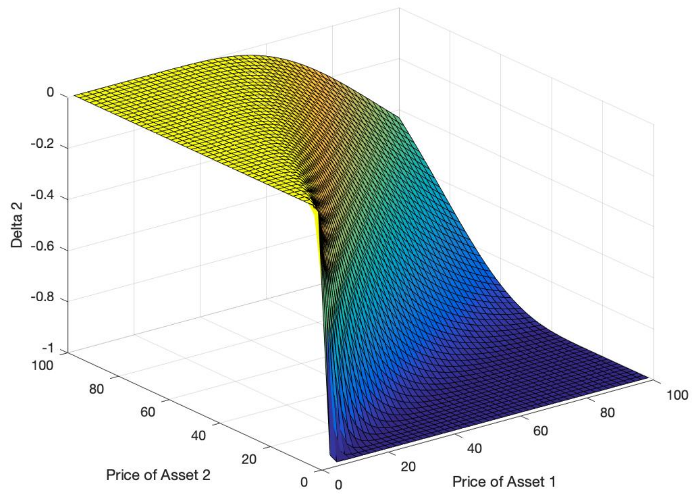


Figure 2.1.2 illustrates the behaviour of Δ_2 caused by prices of the two assets at initiation.

Figure 2.1.2: Exchange option Δ_2



From the two figures, Δ_1 has a positive relationship with $S_1(0)$ and a negative relationship with $S_2(0)$. Δ_2 has a negative relationship with $S_1(0)$ and a positive relationship with $S_2(0)$. This is not at all surprising because $N(x)$ is an monotonically increasing function.

Theta is commonly referred to as an option's time decay. Holding all other parameters at constant, Theta will decrease as option nears its maturity. This is because the option will gradually lose its utility as it matures. Let $\tau = T - t.$, then the Theta of Exchange option is defined as:

$$\Theta(t) = \frac{\partial V(t, s_1, s_2)}{\partial \tau}.$$

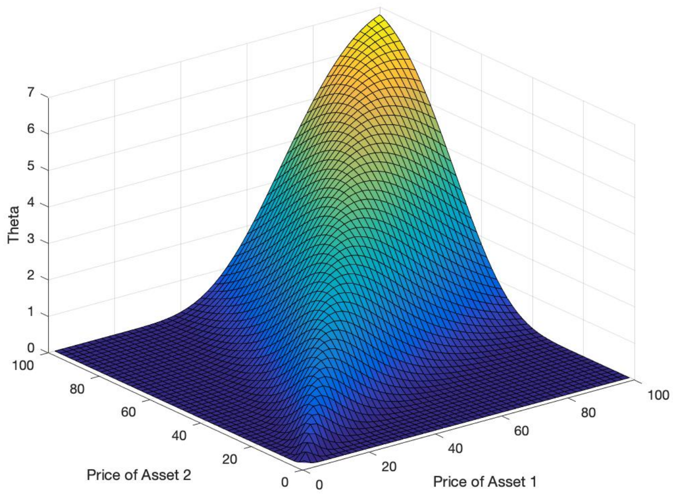
Theorem 2.2.2 (**Exchange option Theta**)**.**

$$\Theta(t) = \frac{\sigma}{2\sqrt{T-t}} s_1 N'(d_+) = -\frac{\sigma}{2\sqrt{T-t}} s_2 N'(d_-).$$

The proofs are available in Appendix 1 A.1.2.

Figure 2.1.3 illustrates the behaviour of Θ caused by changes in prices of the two assets at initiation.

Figure 2.1.3: Exchange option Θ



Onto the second-order Greeks, Gamma is sensitivity of Delta to change in the underlying assets. It takes on the largest value for at the money options. When an option is at the money, a slight price movement of the asset will cause it to be either in the money or out of the money. Thus, Delta of the option will have the greatest instantaneous change in magnitude. Exchange option has three Gammas, they are:

$$\Gamma_{11}(t) = \frac{\partial V^2(t, s_1, s_2)}{\partial s_1^2}, \quad \Gamma_{12}(t) = \frac{\partial V^2(t, s_1, s_2)}{\partial s_1 \partial s_2}, \quad \Gamma_{22}(t) = \frac{\partial V^2(t, s_1, s_2)}{\partial s_2^2}.$$

Theorem 2.2.3 (**Exchange option Gamma**).

$$\Gamma_{11}(t) = \frac{1}{\sigma\sqrt{T-t}} \frac{N'(d_+)}{s_1}, \quad \Gamma_{22}(t) = \frac{1}{\sigma\sqrt{T-t}} \frac{N'(d_-)}{s_2},$$
$$\Gamma_{12}(t) = -\frac{1}{\sigma\sqrt{T-t}} \frac{N'(d_+)}{s_2} = -\frac{1}{\sigma\sqrt{T-t}} \frac{N'(d_-)}{s_1} = \Gamma_{21}(t).$$

The proofs are available in Appendix 1 A.1.3.

Figures 2.1.4, 2.1.5 and 2.1.6 illustrates the behaviour of Gammas caused by changes in price of the two assets at initiation. One may observe a spike in Gamma values along the diagonal of the S_1 and S_2 plane. The option is at the money on the diagonal. Therefore the figures are consistent with the behaviour assertion of Gamma.

Figure 2.1.4: Exchange option Γ_{11}

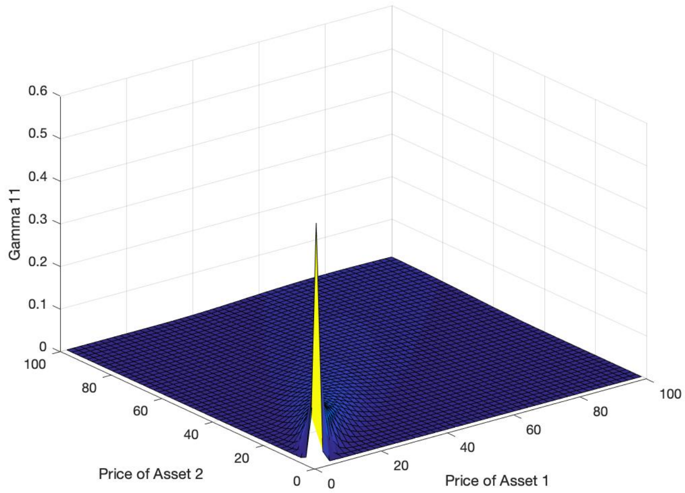


Figure 2.1.5: Exchange option Γ_{12}

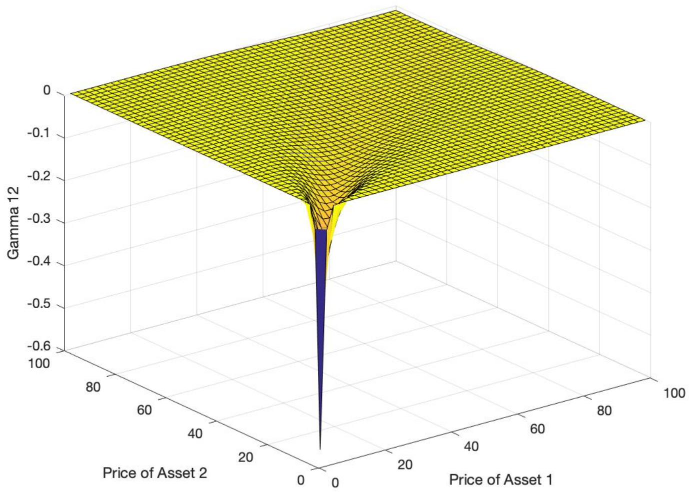
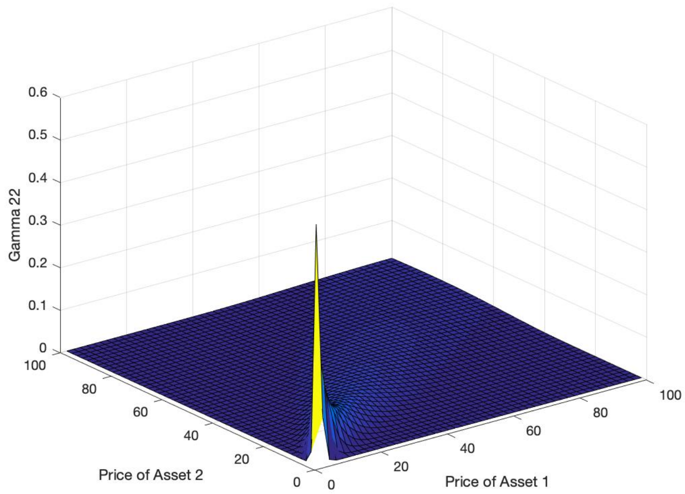


Figure 2.1.6: Exchange option Γ_{22}



Charm is sensitivity of Delta to time decay. It is often just referred to as Delta decay. Exchange option has two Charms, they are:

$$Charm_1(t) = \frac{\partial V^2(t, s_1, s_2)}{\partial s_1 \partial \tau}, \quad Charm_2(t) = \frac{\partial V^2(t, s_1, s_2)}{\partial s_2 \partial \tau}.$$

Theorem 2.2.4 (**Exchange option Charm**)**.**

$$Charm_1(t) = N'(d_+)\Big(- \frac{\log\left(\frac{s_1}{s_2}\right)}{2\sigma(T-t)^{\frac{3}{2}}} + \frac{\sigma}{4\sqrt{T-t}}\Big),$$
$$Charm_2(t) = N'(d_-)\Big(\frac{\log\left(\frac{s_1}{s_2}\right)}{2\sigma(T-t)^{\frac{3}{2}}} + \frac{\sigma}{4\sqrt{T-t}}\Big).$$

The proofs are available in Appendix 1 A.1.4.

Figures 2.1.7 and 2.1.8 illustrates the behaviour of Charms when the initiation price of the two assets changes.

Figure 2.1.7: Exchange option $Charm_1$

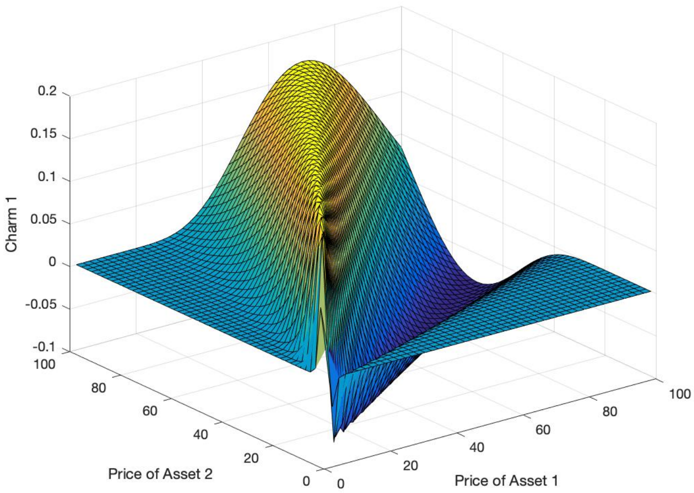
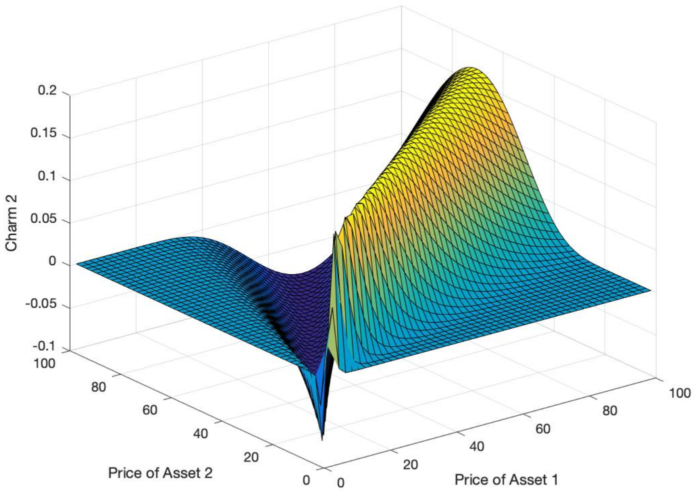


Figure 2.1.8: Exchange option $Charm_2$



For the third-order Greeks, Speed is sensitivity of Gamma to change in the underlying asset. Exchange option has four Speeds, they are:

$$Speed_{111}(t) = \frac{\partial V^3(t, s_1, s_2)}{\partial s_1^3}, \quad Speed_{112}(t) = \frac{\partial V^3(t, s_1, s_2)}{\partial s_1^2 \partial s_2},$$
$$Speed_{122}(t) = \frac{\partial V^3(t, s_1, s_2)}{\partial s_1 \partial s_2^2}, \quad Speed_{122}(t) = \frac{\partial V^3(t, s_1, s_2)}{\partial s_2^3}.$$

Theorem 2.2.5 (**Exchange option Speed**).

$$Speed_{111}(t) = -\frac{N'(d_+)}{\sigma s_1^2\sqrt{T-t}}\Big(\frac{2d_+}{\sigma s_1\sqrt{T-t}}+1\Big) = -\frac{\Gamma_{11}(t)}{s_1}\Big(\frac{2d_+}{\sigma s_1\sqrt{T-t}}+1\Big),$$
$$Speed_{222}(t) = -\frac{N'(d_-)}{\sigma s_2^2\sqrt{T-t}}\Big(\frac{2d_-}{\sigma s_2\sqrt{T-t}}+1\Big) = -\frac{\Gamma_{22}(t)}{s_2}\Big(\frac{2d_-}{\sigma s_2\sqrt{T-t}}+1\Big),$$
$$Speed_{112}(t) = Speed_{121}(t) = Speed_{211}(t) = -\frac{2d_+N'(d_+)}{\sigma^2(T-t)s_1s_2} = \frac{2d_+\Gamma_{12}(t)}{\sigma(T-t)^{\frac{1}{2}}s_1},$$
$$Speed_{221}(t) = Speed_{212}(t) = Speed_{122}(t) = -\frac{2d_-N'(d_-)}{\sigma^2(T-t)s_1s_2} = \frac{2d_-\Gamma_{21}(t)}{\sigma(T-t)^{\frac{1}{2}}s_2}.$$

The proofs are available in Appendix 1 A.1.5.

Colour is the sensitivity of Gamma to time decay. Exchange option has three Colours, they are:

$$Colour_{11}(t) = \frac{\partial V^3(t,s_1,s_2)}{\partial s_1^2\partial\tau}, \quad Colour_{12}(t) = \frac{\partial V^3(t,s_1,s_2)}{\partial s_1\partial s_2\partial\tau}, \quad Colour_{22}(t) = \frac{\partial V^3(t,s_1,s_2)}{\partial s_2^2\partial\tau}.$$

Theorem 2.2.6 (**Exchange option Colour**).

$$Colour_{11}(t) = -\frac{N'(d_+)}{2^3\sigma^3(T-t)^{\frac{5}{2}}s_1}\Big(\sigma^4(T-t)^2+4\sigma^2(T-t)-4\log^2\big(\frac{S_1}{S_2}(t)\big)\Big),$$
$$Colour_{22}(t) = -\frac{N'(d_-)}{2^3\sigma^3(T-t)^{\frac{5}{2}}s_2}\Big(\sigma^4(T-t)^2+4\sigma^2(T-t)-4\log^2\big(\frac{S_1}{S_2}(t)\big)\Big),$$
$$Colour_{12}(t) = \frac{N'(d_+)}{2^3\sigma^3(T-t)^{\frac{5}{2}}s_1}\Big(\sigma^4(T-t)^2+4\sigma^2(T-t)-4\log^2\big(\frac{S_1}{S_2}(t)\big)\Big).$$

The proofs are available in Appendix 1 A.1.6.

Currently, there are no formal names given to fourth order Greeks. The sensitivity of Speed due to change in the underlying asset will be referred as *Acceleration*. Exchange option has five Accelerations, they are:

$$Acceleration_{1111}(t) = \frac{\partial V^4(t,s_1,s_2)}{\partial s_1^4}, \quad Acceleration_{1112}(t) = \frac{\partial V^4(t,s_1,s_2)}{\partial s_1^3\partial s_2},$$
$$Acceleration_{1122}(t) = \frac{\partial V^4(t,s_1,s_2)}{\partial s_1^2\partial s_2^2}, \quad Acceleration_{1222}(t) = \frac{\partial V^4(t,s_1,s_2)}{\partial s_1\partial s_2^3},$$
$$Acceleration_{2222}(t) = \frac{\partial V^4(t,s_1,s_2)}{\partial s_2^4}.$$

Theorem 2.2.7 **(Exchange option Acceleration).**

$$\begin{aligned}
Acceleration_{1111}(t) &= -\frac{2\Gamma_{11}(t)}{\sigma^2(T-t)s_1^3}\Big(\frac{2d_+^2}{s_1}+1\Big),\\
Acceleration_{1112}(t) &= \frac{2\Gamma_{11}(t)}{\sigma\sqrt{T-t}s_1s_2}\Big(\frac{2d_+^2}{\sigma\sqrt{T-t}s_1}+d_+ + \frac{1}{\sigma\sqrt{T-t}s_1}\Big),\\
Acceleration_{1122}(t) &= \frac{2\Gamma_{12}}{\sigma^2(T-t)s_1s_2}\big(d_+d_- - 1\big),\\
Acceleration_{1222}(t) &= \frac{2\Gamma_{22}(t)}{\sigma\sqrt{T-t}s_1s_2}\Big(\frac{2d_-^2}{\sigma\sqrt{T-t}s_2}+d_- - \frac{1}{\sigma\sqrt{T-t}s_2}\Big)\\
Acceleration_{2222}(t) &= -\frac{2\Gamma_{22}(t)}{\sigma^2(T-t)s_2^3}\Big(\frac{2d_-^2}{s_2}+1\Big).
\end{aligned}$$

The proofs are available in Appendix 1 A.1.7.

All of the Exchange option Greeks examined above are summarized in the table below:

Table 2.1.1: Exchange option Greeks

First order Greek	
$\Delta_1(t)$	$N(d_+)$
$\Delta_2(t)$	$-N(d_-)$
$\Theta(t)$	$\frac{\sigma}{2\sqrt{T-t}} s_1 N'(d_+) = -\frac{\sigma}{2\sqrt{T-t}} s_2 N'(d_-)$
Second order Greek	
$\Gamma_{11}(t)$	$\frac{1}{\sigma\sqrt{T-t}} \frac{N'(d_+)}{s_1}$
$\Gamma_{22}(t)$	$\frac{1}{\sigma\sqrt{T-t}} \frac{N'(d_-)}{s_2}$
$\Gamma_{12}(t) = \Gamma_{21}(t)$	$-\frac{1}{\sigma\sqrt{T-t}} \frac{N'(d_+)}{s_2} = -\frac{1}{\sigma\sqrt{T-t}} \frac{N'(d_-)}{s_1}$
$Charm_1(t)$	$N'(d_+)\Big(-\frac{\log\left(\frac{s_1}{s_2}\right)}{2\sigma(T-t)^{\frac{3}{2}}} + \frac{\sigma}{4\sqrt{T-t}}\Big)$
$Charm_2(t)$	$N'(d_-)\Big(\frac{\log\left(\frac{s_1}{s_2}\right)}{2\sigma(T-t)^{\frac{3}{2}}} + \frac{\sigma}{4\sqrt{T-t}}\Big)$
Third order Greek	
$Speed_{111}(t)$	$-\frac{N'(d_+)}{\sigma s_1^2\sqrt{T-t}}\big(\frac{2d_+}{\sigma s_1\sqrt{T-t}} + 1\big)$
$Speed_{222}(t)$	$-\frac{N'(d_-)}{\sigma s_2^2\sqrt{T-t}}\big(\frac{2d_-}{\sigma s_2\sqrt{T-t}} + 1\big)$
$Speed_{112}(t)$	$-\frac{2d_+ N'(d_+)}{\sigma^2(T-t)s_1 s_2}$
$Speed_{221}(t)$	$-\frac{2d_- N'(d_-)}{\sigma^2(T-t)s_1 s_2}$
$Colour_{11}(t)$	$-\frac{N'(d_+)}{2^3\sigma^3(T-t)^{\frac{5}{2}} s_1}\Big(\sigma^4(T-t)^2 + 4\sigma^2(T-t) - 4\log^2\left(\frac{s_1}{s_2}\right)\Big)$
$Colour_{22}(t)$	$-\frac{N'(d_-)}{2^3\sigma^3(T-t)^{\frac{5}{2}} s_2}\Big(\sigma^4(T-t)^2 + 4\sigma^2(T-t) - 4\log^2\left(\frac{s_1}{s_2}\right)\Big)$
$Colour_{12}(t)$	$-\frac{N'(d_+)}{2^3\sigma^3(T-t)^{\frac{5}{2}} s_1}\Big(\sigma^4(T-t)^2 + 4\sigma^2(T-t) - 4\log^2\left(\frac{s_1}{s_2}\right)\Big)$
Fourth order Greek	
$Acceleration_{1111}(t)$	$-\frac{2\Gamma_{11}(t)}{\sigma^2(T-t)s_1^3}\big(\frac{2d_+^2}{s_1} + 1\big)$
$Acceleration_{1112}(t)$	$\frac{2\Gamma_{11}(t)}{\sigma\sqrt{T-t}s_1 s_2}\big(\frac{2d_+^2}{\sigma\sqrt{T-t}s_1} + d_+ + \frac{1}{\sigma\sqrt{T-t}s_1}\big)$
$Acceleration_{1122}(t)$	$\frac{2\Gamma_{12}}{\sigma^2(T-t)s_1 s_2}\big(d_+ d_- - 1\big)$
$Acceleration_{1222}(t)$	$\frac{2\Gamma_{22}(t)}{\sigma\sqrt{T-t}s_1 s_2}\big(\frac{2d_-^2}{\sigma\sqrt{T-t}s_2} + d_- - \frac{1}{\sigma\sqrt{T-t}s_2}\big)$
$Acceleration_{2222}(t)$	$-\frac{2\Gamma_{22}(t)}{\sigma^2(T-t)s_2^3}\big(\frac{2d_-^2}{s_2} + 1\big)$

These particular Margrabe Greeks will taken on an important role in the subsequent chapters of this thesis. More specifically, the Greeks will aid in the establishment of an existence and uniqueness requirement for the SDEs in a finite liquidity market model.

2.2 Spread option

Spread option is similar to the exchange option. It allows the holder the choice to exchange one asset for another, at a cost. There are spread option issued on many of the same assets as exchange option. However, they are primarily traded over-the-counter. Commodity spreads provide traders an unique way to have exposure of the production process. The most notable examples are the crack, crush, and spark spread options. They measure profits in the oil, soybean, and electricity markets respectively.

2.2.1 Pricing Formula

The risk-neutral pricing formula for spread option is:

$$V_{Spread}\big(t, S_1(t), S_2(t)\big) = \widetilde{\mathbb{E}}\big[e^{-r\tau}\big(S_1(T) - S_2(T) - K\big)^+ |\mathscr{F}(t)\big]. \qquad (2.2.1)$$

Unlike exchange option, there are no closed form pricing formulas for spread option. There are many numerical methods available for spread option such as finite difference, Monte Carlo and numerical integration. This thesis shall adopt a Fourier transform method developed by Hurd and Zhou (2010) [29]. The reason is because Fourier discretization have fast execution speed and can simultaneously compute a range of option prices. To summarize their method for the BSM model of (1.3.1), one should first consider transforming the payout function as $(s_1 - s_2 - K)^+ = K(e^{x_1} - e^{x_2} - 1)^+ = P(\mathbf{x})$. Then, $\big(X_1(T), X_2(T)\big)$ is a bi-variate normal with conditional characteristic function:

$$\Phi_{\mathbf{x}(t)}(\mathbf{u}, \tau) = \exp\big\{i\mathbf{u}'\big(\mathbf{x} + r\mathbf{1} - \frac{1}{2}(\sigma_1^2 + \sigma_2^2)\big)\tau - \frac{1}{2}\mathbf{u}'\Sigma\mathbf{u}\tau\big\},$$

where

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho \\ \sigma_1\sigma_2\rho & \sigma_2^2 \end{bmatrix}.$$

The pricing formula created by Hurd and Zhou for the GBM model (1.3.1) can be summarized by the following theorem:

Theorem 2.2.1 (Hurd and Zhou Spread option formula).

$$V_{Spread}\big(t, s_1, s_2\big) = \frac{1}{(2\pi)^2} e^{-r\tau} K \int\int_{\mathbb{R}^2 + i\epsilon} e^{i\mathbf{u}'\mathbf{X}(t)} \hat{P}(\mathbf{u}) \Phi(\mathbf{u}, \tau) d\mathbf{u}, \qquad (2.2.2)$$

where

$$\hat{P}(\mathbf{u}) = \frac{\Gamma\big(i(u_1 + u_2) - 1\big)\Gamma\big(-iu_2\big)}{\Gamma\big(iu_1 + 1\big)},$$

$$\Phi(\mathbf{u}, \tau) = \exp\big\{i\mathbf{u}'\big(r\mathbf{1} - \frac{1}{2}(\sigma_1^2 + \sigma_2^2)\big)\tau - \frac{1}{2}\mathbf{u}'\Sigma\mathbf{u}\tau\big\}.$$

This pricing formula is only valid for $\epsilon_2 > 0$ *and* $\epsilon_1 + \epsilon_2 < -1$.

Proof: There are few key ideas used in their pricing formula. The Fourier transformation of a PDF is the complex conjugate of its respective characteristic function. This property can be combined along with Parseval's identity to simplify pricing formula (2.2.1). This is an important step because evaluating expectations are often easier in the Fourier frequency space. Applying Parseval's identity C.1, one may obtain:

$$\begin{aligned} V_{Spread}\big(t, s_1, s_2\big) &= e^{-r\tau} K \int\int_{\mathbb{R}^2} P(\mathbf{x}(t)) N_2'(\mathbf{x}) d\mathbf{x} \\ &= \frac{e^{-r\tau}K}{(2\pi)^2} \int\int_{\mathbb{R}^2} \mathcal{F}\big[P\big(\mathbf{x}(t)\big)\big](\mathbf{u}) \overline{\mathcal{F}\big[N_2'\big(\mathbf{x}(t)\big)\big](\mathbf{u})} d\mathbf{u} \end{aligned} \tag{2.2.3}$$

The Fourier transform of the PDF's complex conjugate is:

$$\begin{aligned} \overline{\mathcal{F}\big[N_2'\big(\mathbf{x}(t)\big)\big](\mathbf{u})} :&= \overline{\int\int_{\mathbb{R}^2+i\epsilon} e^{-i\mathbf{x}'(t)\mathbf{u}} N_2'(\mathbf{x}) d\mathbf{x}} = \widetilde{\mathbb{E}}\big[\overline{e^{-i\mathbf{X}'(t)\mathbf{u}}} | \mathscr{F}(t)\big] \\ &= \widetilde{\mathbb{E}}\big[e^{i\mathbf{X}'(t)\mathbf{u}} | \mathscr{F}(t)\big] = \Phi_{\mathbf{x}(t)}(\mathbf{u}), \end{aligned} \tag{2.2.4}$$

The Fourier transform of $P\big(\mathbf{X}(t)\big)$ does not exist. For the Fourier inverse theorem to be applicable, $P\big(\mathbf{X}(t)\big)$ needs to be coupled with a dampening factor $e^{\mathbf{X}'(t)\epsilon}$. To show this, one can in fact check $e^{\mathbf{X}'(t)\epsilon} P\big(\mathbf{X}(t)\big)$ is in $\mathbb{L}(\mathbb{R}^2)$.

$$\mathcal{F}^{-1}\big[\hat{P}(\mathbf{u})\big]\big(\mathbf{x}(t)\big) := \frac{1}{(2\pi)^2} e^{-\mathbf{x}'(t)\epsilon} \int\int_{\mathbb{R}^2} e^{-i\mathbf{x}'(t)\bar{\mathbf{u}}} \bar{P}(\bar{\mathbf{u}}) d\bar{\mathbf{u}} = \frac{1}{(2\pi)^2} \int\int_{\mathbb{R}^2+i\epsilon} e^{-i\mathbf{x}'(t)\mathbf{u}} \hat{P}(\mathbf{u}) d\mathbf{u},$$

where $\mathbf{u} = \bar{\mathbf{u}} + i\epsilon$, and

$$\hat{P}(\mathbf{u}) = \mathcal{F}\big[P\big(\mathbf{x}(t)\big)\big](\mathbf{u}) := \int\int_{\mathbb{R}^2} e^{-i\mathbf{x}'(t)\mathbf{u}} \big(e^{x_1(t)} - e^{x_2(t)} - 1\big)^+ d\mathbf{x}. \tag{2.2.5}$$

To obtain integrability almost everywhere, one may restrict the integration region to $\mathcal{D} = \big\{\mathbf{x}(t) : x_1(t) > 0, x_2(t) < \log(e^{x_1(t)} - 1)\big\}$. However, precaution needs to be taken when dealing with values along the edges of $\mathcal{D}$. Expression (2.2.5) can now be evaluated as:

$$\begin{aligned} \hat{P}(\mathbf{u}) &= \int_0^\infty e^{-ix_1(t)u_1} \int_{-\infty}^{\log(e^{x_1(t)}-1)} \big(e^{-ix_2(t)u_2}(e^{x_1(t)} - 1) - e^{(1-iu_2)x_2(t)}\big) dx_2 dx_1 \\ &= \int_0^\infty e^{-ix_1(t)u_1} \Big(\big(\frac{e^{-ix_2(t)u_2}}{-iu_2} (e^{x_1(t)} - 1) - \frac{e^{(1-iu_2)x_2(t)}}{1 - iu_2} \big) \Big|_{-\infty}^{\log(e^{x_1(t)}-1)} \Big) dx_2 dx_1. \end{aligned}$$

It is important to make note that this double integral will only exist under certain conditions for ϵ. These conditions can be determined later on in the proof. In the main time, suppose the integral exists, then it will simplify to:

$$\hat{P}(\mathbf{u}) = \Big(\frac{1}{-iu_2} - \frac{1}{1 - iu_2}\Big) \int_0^\infty e^{-ix_1(t)u_1} (e^{x_1(t)} - 1)^{1-iu_2} dx_1.$$

Let $z = e^{-x_1(t)}$, then the expression for $\hat{P}(\mathbf{u})$ will become the integral form of a complex beta function, that is

$$\hat{P}(\mathbf{u}) = \frac{1}{-iu_2(1-iu_2)} \int_0^1 z^{i(u_1+u_2)-2}(z-1)^{1-iu_2} dx_1 = \frac{\Gamma\big(i(u_1+u_2)-1\big)\Gamma\big(-iu_2+2\big)}{-iu_2(1-iu_2)\Gamma\big(iu_1+1\big)}.$$

Furthermore, one may apply the recursive property of gamma function
$\Gamma(z+1) = z\Gamma(z)$ to obtain:

$$\hat{P}(\mathbf{u}) = \frac{\Gamma\big(i(u_1+u_2)-1\big)\Gamma\big(-iu_2\big)}{\Gamma\big(iu_1+1\big)}. \tag{2.2.6}$$

The complex Gamma function $\Gamma(z)$ is only define for $\Re(z) > 0$. Recall $u_1 = \bar{u}_1 + i\epsilon_1$ and $u_2 = \bar{u}_2 + i\epsilon_2$, then one may obtain the requirements of ϵ from (2.2.6):

$$\Re\big(i(u_1+u_2)-1\big) > 0 \Rightarrow \epsilon_1 + \epsilon_2 < -1,$$
$$\Re\big(-iu_2\big) > 0 \Rightarrow \epsilon_2 > 0,$$
$$\Re\big(iu_1+1\big) > 0 \Rightarrow \epsilon_1 > 1.$$

Therefore, $\hat{P}(\mathbf{u})$ is only defined on the contour $\mathbb{R}^2 + i\epsilon$, when $\epsilon_2 > 0$ and $\epsilon_1 + \epsilon_2 < -1$.

Substituting (2.2.4) and (2.2.6) into (2.2.3), it will become:

$$\begin{aligned} V_{Spread}\big(t, s_1, s_2\big) &= \frac{1}{(2\pi)^2} e^{-r\tau} K \int\int_{\mathbb{R}^2+i\epsilon} \hat{P}(\mathbf{u}) \Phi_{\mathbf{x}(t)}(\mathbf{u}) d\mathbf{u} \\ &= \frac{1}{(2\pi)^2} e^{-r\tau} K \int\int_{\mathbb{R}^2+i\epsilon} e^{i\mathbf{u}'\mathbf{X}(t)} \hat{P}(\mathbf{u}) \Phi(\mathbf{u}, \tau) d\mathbf{u}, \end{aligned}$$

where

$$\Phi(\mathbf{u}, \tau) = \exp\big\{i\mathbf{u}'\big(r\mathbf{1} - \frac{1}{2}(\sigma_1^2 + \sigma_2^2)\big)\tau - \frac{1}{2}\mathbf{u}'\Sigma\mathbf{u}\tau\big\}.$$

□

This concludes the proof.

2.2.2 Greeks of Spread option

Another advantage of Hurd and Zhou's method is its simplicity when it comes to the computation of Greeks. For instance, the spread option Deltas can be computed in the following manner:

$$\mathbf{\Delta}(t) = \begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix}(t) = \frac{\partial V_{spr}\big(t, \mathbf{s}\big)}{\partial \mathbf{s}} = \frac{\partial \mathbf{x}}{\partial \mathbf{s}} \frac{\partial V_{spr}\big(t, \mathbf{s}\big)}{\partial \mathbf{x}}.$$

Theorem 2.2.2 (Spread option Delta).

$$\mathbf{\Delta}(t) = (2\pi)^{-2} e^{-r\tau} K \mathcal{H} \bar{\mathbf{\Delta}}(t), \tag{2.2.1}$$

where

$$\bar{\mathbf{\Delta}}(t) = \int\int_{\mathbb{R}^2 + i\epsilon} i\mathbf{u} e^{i\mathbf{u}'\mathbf{X}(t)} \Phi(\boldsymbol{u}, \tau) \hat{P}(\mathbf{u}) d\boldsymbol{u},$$

$$\mathcal{H} = \begin{bmatrix} \frac{1}{s_1} & 0 \\ 0 & \frac{1}{s_2} \end{bmatrix}.$$

The proofs are available in Appendix 1 A.2.1.

To calculation the Greeks, a numerical scheme such as Fast Fourier Transformation has to be used. The FFT schemes used to generate the graphs are available in Section 5.2. The parameters used to generate all the Greek graphs are: $K = 4$, $T = 1$, $\sigma_1 = 0.4$, $\sigma_2 = 0.2$, $\rho = 0.5$ and $r = 0.05$.

Figure 2.2.1: Spread option Δ_1

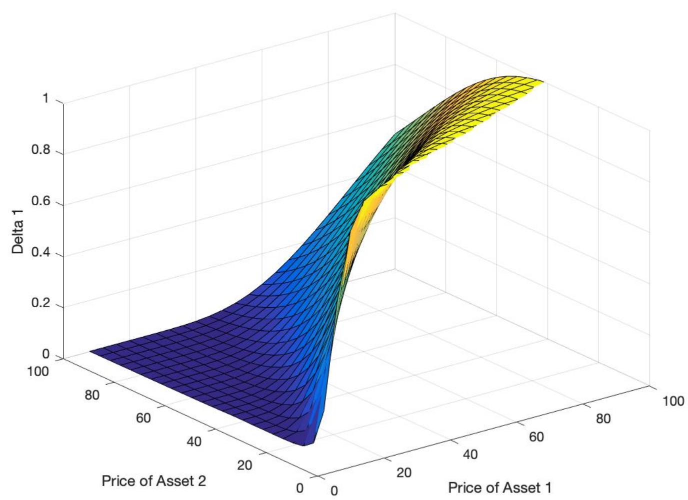
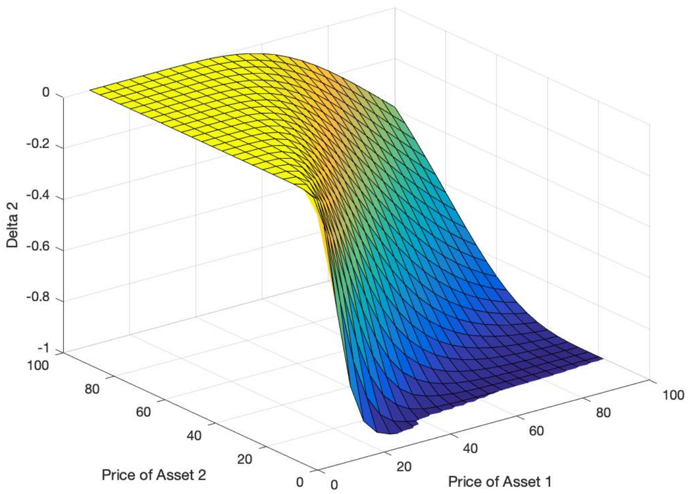


Figure 2.2.2: Spread option Δ_2



The spread option Theta is given by:

$$\Theta(t) = \frac{\partial V_{spr}\big(t, \mathbf{s}\big)}{\partial \tau}.$$

Theorem 2.2.3 (**Spread option Theta**).

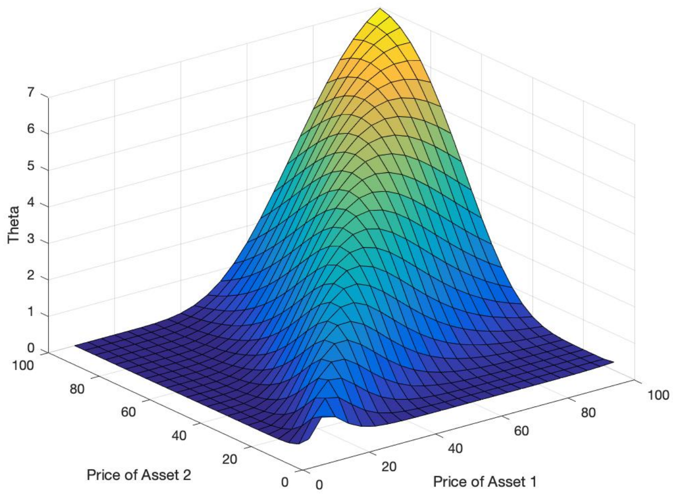
$$\Theta(t) = (2\pi)^{-2} K e^{-r\tau} \bar{\mathbf{\Theta}}(t), \tag{2.2.2}$$

where

$$\bar{\mathbf{\Theta}}(t) = \int\int_{\mathbb{R}^2+i\epsilon} \Big(i\mathbf{u}'\big(r\mathbf{1} - \frac{1}{2}(\sigma_1^2 + \sigma_2^2)\big) - \frac{1}{2}\mathbf{u}'\Sigma\mathbf{u} - r\Big) e^{i\mathbf{u}'\mathbf{X}(t)} \Phi(\boldsymbol{u}, \tau) \hat{P}(\mathbf{u}) d\boldsymbol{u}$$

The proofs are available in Appendix 1 A.2.2.

Figure 2.2.3: Spread option Θ



For the higher order Greeks, we shall adopt tensor notations for their expressions. The Spread option Gamma can be defined as:

$$\boldsymbol{\Gamma}(t) = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{21} & \Gamma_{22} \end{bmatrix}(t) = \frac{\partial \boldsymbol{\Delta}(\mathbf{t})}{\partial \mathbf{s}}.$$

where

Theorem 2.2.4 (Spread option Gamma).

$$\boldsymbol{\Gamma}(t) = -(2\pi)^{-2} e^{-r\tau} K \Big(\mathcal{T}^{(3)} \bar{\boldsymbol{\Delta}}(t) + \mathcal{H} \bar{\boldsymbol{\Gamma}}(t) \mathcal{H} \Big), \tag{2.2.3}$$

where

$$\bar{\boldsymbol{\Gamma}}(t) = \int\int_{\mathbb{R}^2 + i\epsilon} \mathbf{u} \bar{\otimes} \mathbf{u} e^{i\mathbf{u}'\mathbf{X}(t)} \Phi(\boldsymbol{u}, \tau) \hat{P}(\mathbf{u}) d\boldsymbol{u},$$

$$\mathcal{T}^{(3)} = \begin{bmatrix} \frac{1}{s_1^2} & 0 \\ 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{s_2^2} \end{bmatrix}.$$

In this subsection, ⊗ denotes the tensor product(see C.2.1), and $\bar{\otimes}$ is the outer product(see C.2.3). We also used some tensor multiplications rules, they are defined in C.2.2. The full proof for Gamma is available in Appendix 1 A.2.3.

Figure 2.2.4: Spread option Γ_{11}

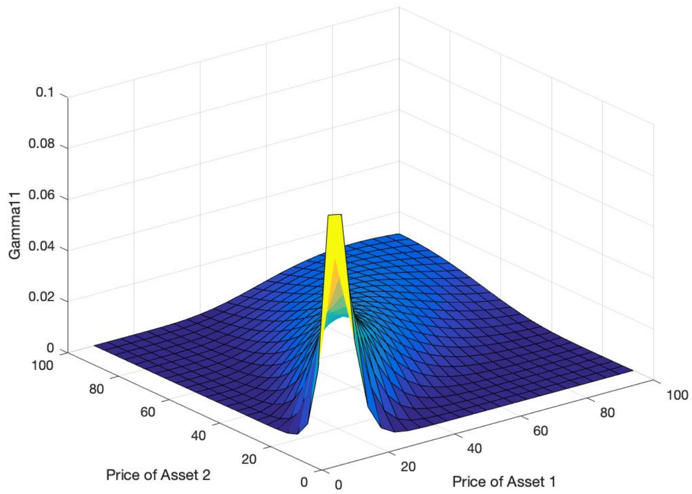


Figure 2.2.5: Spread option Γ_{12}

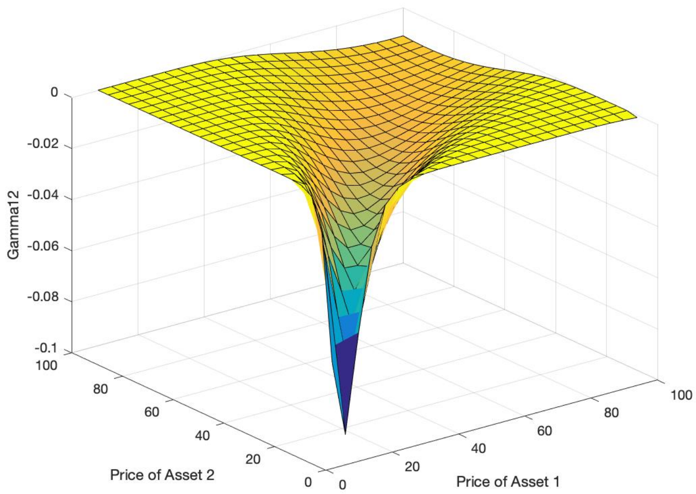
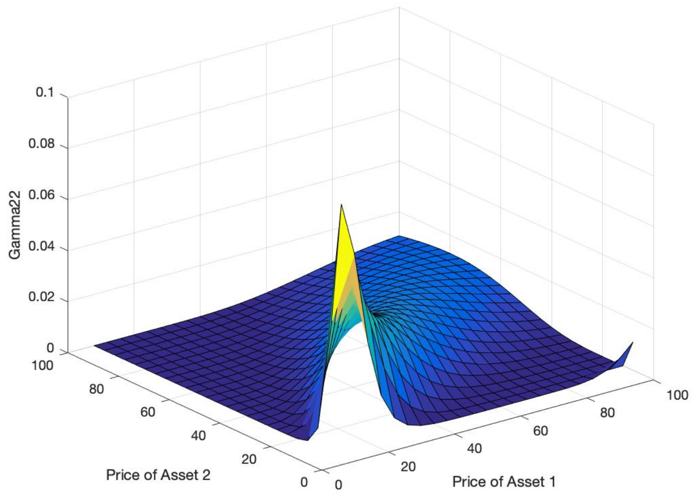


Figure 2.2.6: Spread option Γ_{22}



The Spread option Charms are defined as:

$$\mathbf{Charm}(t) = \begin{bmatrix} Charm_1 \\ Charm_2 \end{bmatrix}(t) = \frac{\partial \boldsymbol{\Delta}(t)}{\partial \tau}.$$

Theorem 2.2.5 (Spread option Charm).

$$\mathbf{Charm}(t) = (2\pi)^{-2} e^{-r\tau} K \mathcal{H} \overline{\mathbf{Charm}}(t), \tag{2.2.4}$$

where

$$\overline{\mathbf{Charm}}(t) = \int\int_{\mathbb{R}^2+i\epsilon} \Big(i\mathbf{u}'\big(r\mathbf{1} - \frac{1}{2}(\sigma_1^2 + \sigma_2^2)\big) - \frac{1}{2}\mathbf{u}'\Sigma\mathbf{u} - r\Big) i\mathbf{u} e^{i\mathbf{u}'\mathbf{X}(t)} \Phi(\boldsymbol{u}, \tau)\hat{P}(\mathbf{u}) d\boldsymbol{u}.$$

The proofs are available in Appendix 1 A.2.4.

Figure 2.2.7: Spread option $Charm_1$

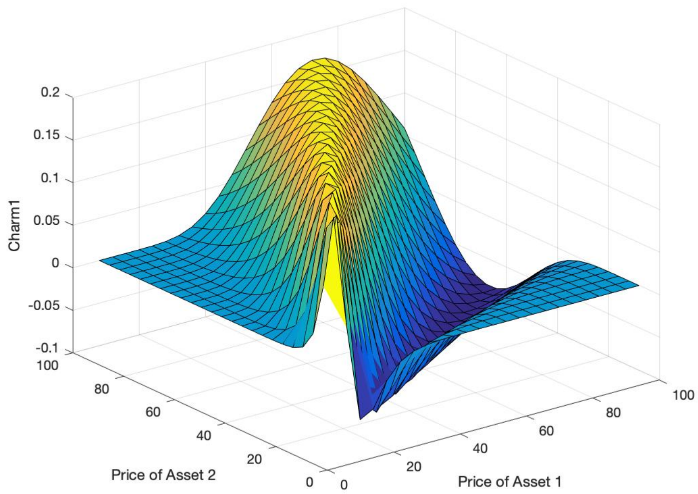
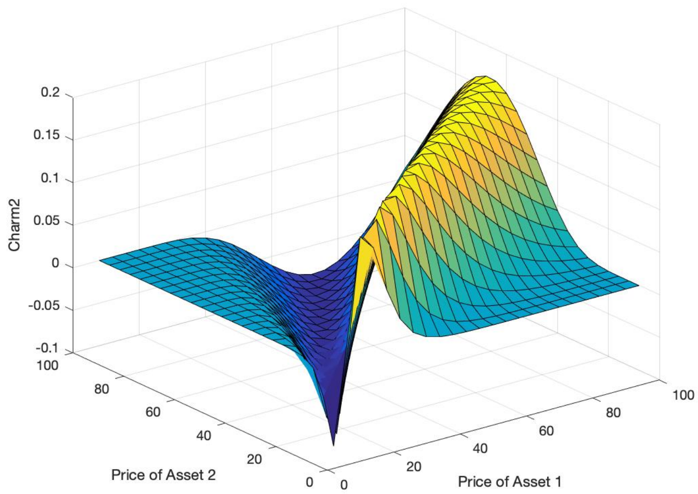


Figure 2.2.8: Spread option $Charm_2$



The Spread option Speed is defined as:

$$\mathbf{Speed} = \begin{bmatrix} Speed_{111} & Speed_{121} \\ Speed_{211} & Speed_{221} \end{bmatrix} \otimes \begin{bmatrix} Speed_{112} & Speed_{122} \\ Speed_{212} & Speed_{222} \end{bmatrix}(t) = \frac{\partial \boldsymbol{\Gamma}(t)}{\partial \mathbf{s}}.$$

Theorem 2.2.6 (**Spread option Speed**).

$$\mathbf{Speed}(t) = (2\pi)^{-2} e^{-r\tau} K\Big(2\mathcal{T}^{(4)}\bar{\boldsymbol{\Delta}}(t) - \mathcal{T}^{(3)}\big(\mathcal{H}\bar{\boldsymbol{\Gamma}}(t)\big) - \mathcal{H}^2\big(\overline{\mathbf{Speed}}(t)\mathcal{H}\big)\Big), \quad (2.2.5)$$

where

$$\overline{\mathbf{Speed}}(t) = \int\int_{\mathbb{R}^2+i\epsilon} i(\mathbf{u}\bar{\otimes}\mathbf{u}\bar{\otimes}\mathbf{u})e^{i\mathbf{u}'\mathbf{X}(t)}\Phi(\boldsymbol{u},\tau)\hat{P}(\mathbf{u})d\boldsymbol{u},$$

$$\mathcal{T}^{(4)} = \left(\begin{bmatrix}\frac{1}{s_1^3} & 0\\ 0 & 0\end{bmatrix}\otimes\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\right)\otimes\left(\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\otimes\begin{bmatrix}0 & 0\\ 0 & \frac{1}{s_2^3}\end{bmatrix}\right).$$

The proofs are available in Appendix 1 A.2.5.

The method required to determine higher order Greeks becomes redundant. It is clear we only need to take higher order partial derivatives of (2.2.2). The computational result for any higher order Greek will be a linear combination of contour integral, with each contour integral having the form of:

$$\overline{\mathbf{Greek}}(t, s_1, s_2) = \int\int_{\mathbb{R}^2+i\epsilon} f_{\otimes}(\mathbf{u})e^{i\mathbf{u}'\mathbf{X}(t)}\Phi(\mathbf{u},\tau)\hat{P}(\mathbf{u})d\mathbf{u} \tag{2.2.6}$$

where $f_{\otimes}(\mathbf{u})$ is some tensor polynomial function. For example, $\mathbf{\Gamma}(t)$ (2.2.3) is a linear combination of the contour integrals $\bar{\mathbf{\Delta}}(t)$ and $\bar{\mathbf{\Gamma}}(t)$, with their tensor polynomial functions being $i\mathbf{u}e^{i\mathbf{u}'\mathbf{X}(t)}$ and $\mathbf{u}\bar{\otimes}\mathbf{u}e^{i\mathbf{u}'\mathbf{X}(t)}$ respectively.(formula available in Appendix 1 A.2)

These terms will be important when establishing the existence and uniqueness requirement for the SDEs in a finite liquidity market framework.

The table below summarizes some Greeks of spread option.

Table 2.2.1: Spread option Greeks

First order Greek	
$\mathbf{\Delta}(t) = \begin{bmatrix}\Delta_1\\ \Delta_2\end{bmatrix}(t)$	$(2\pi)^{-2}e^{-r\tau}K\mathcal{H}(t)\bar{\mathbf{\Delta}}(t)$
$\Theta(t)$	$(2\pi)^{-2}e^{-r\tau}K\bar{\mathbf{\Theta}}(t)$
Second order Greek	
$\mathbf{\Gamma}(t) = \begin{bmatrix}\Gamma_{11} & \Gamma_{12}\\ \Gamma_{21} & \Gamma_{22}\end{bmatrix}(t)$	$-(2\pi)^{-2}e^{-r\tau}K\Big(\mathcal{T}^{(3)}(t)\bar{\mathbf{\Delta}}(t) + \mathcal{H}(t)\bar{\mathbf{\Gamma}}(t)\mathcal{H}(t)\Big)$
$\mathbf{Charm}(t) = \begin{bmatrix}Charm_1\\ Charm_2\end{bmatrix}(t)$	$(2\pi)^{-2}e^{-r\tau}K\mathcal{H}(t)\overline{\mathbf{Charm}}(t)$
$\mathbf{Speed}(t) = \begin{bmatrix}Speed_{111} & Speed_{112}\\ Speed_{121} & Speed_{211}\end{bmatrix}$ $\otimes\begin{bmatrix}Speed_{122} & Speed_{221}\\ Speed_{212} & Speed_{222}\end{bmatrix}(t)$	$(2\pi)^{-2}e^{-r\tau}K\Big(2\mathcal{T}^{(4)}(t)\bar{\mathbf{\Delta}}(t) - \mathcal{T}^{(3)}(t)\big(\mathcal{H}(t)\bar{\mathbf{\Gamma}}(t)\big)$ $-\mathcal{H}^2(t)\big(\overline{\mathbf{Speed}}(t)\mathcal{H}(t)\big)\Big)$

Chapter 3

Finite Liquidity Market Model

3.1 Model Setup

In the Black-Scholes model of (1.3.1) the market is assumed to be perfectly liquid, i.e. trading does not effect assets' price. BS model is an inaccurate representation of the real world because trading an asset will inevitably effect its traded price. In a market with finite liquidity, trading will have an impact on the price of assets. The model studied in this thesis will be a slight variation of the model in [1]. The risky assets have the following dynamics:

$$\begin{aligned} \frac{dS_1(t)}{S_1(t)} &= \mu_1 dt + \sigma_1 dW_1(t) + \lambda\big(t, S_1(t), S_2(t)\big) df\big(t, S_1(t), S_2(t)\big), \\ \frac{dS_2(t)}{S_2(t)} &= \mu_2 dt + \sigma_2 \rho dW_1(t) + \sigma_2 \sqrt{1-\rho^2} dW_2(t), \\ \frac{dD(t)}{D(t)} &= -r dt. \end{aligned} \tag{3.1.1}$$

The additional $\lambda\big(t, S_1(t), S_2(t)\big) df\big(t, S_1(t), S_2(t)\big)$ term on asset 1 represents the price impact of a trading strategy. Under this particular finite liquidity model, the future price of the assets are determined by the following factors:

- drift constants μ_1 and μ_2,
- volatility constants σ_1 and σ_2,
- a correlation constant ρ that has value between -1 and 1,
- a risk-free interest rate r,
- a price impact term $\lambda\big(t, S_1(t), S_2(t)\big) df\big(t, S_1(t), S_2(t)\big)$ of a large trader's strategy $f\big(t, S_1(t), S_2(t)\big)$ on asset one, and an impact function $\lambda(t, s_1, s_2)$.

The next step is to obtain the SDE of asset 1. The differential of $f\big(t, S_1(t), S_2(t)\big)$ according to Itô's formula is:

$$df\big(t, S_1(t), S_2(t)\big) = \frac{\partial f}{\partial t}dt + \frac{\partial f}{\partial s_1}dS_1(t) + \frac{\partial f}{\partial s_2}dS_2(t) + \frac{\partial^2 f}{\partial s_1 \partial s_2}dS_1(t)dS_2(t) + \frac{1}{2}\frac{\partial^2 f}{\partial s_1^2}dS_1(t) + \frac{1}{2}\frac{\partial^2 f}{\partial s_2^2}dS_2(t)dS_2(t). \tag{3.1.2}$$

Next, substitute (3.1.2) into $dS_1(t)$ of (3.1.1) to get:

$$dS_1(t) = \mu_1 S_1(t)dt + \sigma_1 S_1(t)dW_1(t) + \lambda\big(t, S_1(t), S_2(t)\big)\Big(\frac{\partial f}{\partial t}dt + \frac{\partial f}{\partial s_1}dS_1(t) + \frac{\partial f}{\partial s_2}dS_2(t) + \frac{\partial^2 f}{\partial s_1 \partial s_2}dS_1(t)dS_2(t) + \frac{1}{2}\frac{\partial^2 f}{\partial s_1^2}dS_1(t)dS_1(t) + \frac{1}{2}\frac{\partial^2 f}{\partial s_2^2}dS_2(t)dS_2(t)\Big). \tag{3.1.3}$$

The next goal is to rearrange (3.1.3) for $dS_1(t)$. Simple algebra applied to the expression for $dS_1(t)$ leads to:

$$dS_1(t) = \frac{1}{1 - \lambda\big(t, S_1(t), S_2(t)\big)\frac{\partial f}{\partial s_1}}\Big(\mu_1 S_1(t)dt + \sigma_1 S_1(t)dW_1(t) + \lambda\big(t, S_1(t), S_2(t)\big)\big(\frac{\partial f}{\partial t}dt + \frac{\partial f}{\partial s_2}dS_2(t) + \frac{\partial^2 f}{\partial s_1 \partial s_2}dS_1(t)dS_2(t) + \frac{1}{2}\frac{\partial^2 f}{\partial s_1^2}dS_1(t)dS_1(t) + \frac{1}{2}\frac{\partial^2 f}{\partial s_2^2}dS_2(t)dS_2(t)\big)\Big). \tag{3.1.4}$$

To further simplify $dS_1(t)$, we are required to compute the quadratic variations in (3.1.4):

$$dS_1(t)dS_1(t) = \frac{1}{\Big(1 - \lambda\big(t, S_1(t), S_2(t)\big)\frac{\partial f}{\partial s_1}\Big)^2}\Big(\sigma_1^2 S_1^2(t) + \lambda^2\big(t, S_1(t), S_2(t)\big)\big(\frac{\partial f}{\partial s_2}\big)^2\sigma_2^2 S_2^2(t) + 2\lambda\big(t, S_1(t), S_2(t)\big)\frac{\partial f}{\partial s_2}\rho\sigma_1\sigma_2 S_1(t)S_2(t)\Big)dt,$$

$$dS_1(t)dS_2(t) = \frac{1}{1 - \lambda\big(t, S_1(t), S_2(t)\big)\frac{\partial f}{\partial s_1}}\Big(\rho\sigma_1\sigma_2 S_1(t)S_2(t) + \sigma_2^2 S_2^2(t)\lambda\big(t, S_1(t), S_2(t)\big)\frac{\partial f}{\partial s_2}\Big)dt,$$

$$dS_2(t)dS_2(t) = \sigma_2^2 S_2^2(t)dt.$$

By substituting the quadratic variations terms into the expression of $dS_1(t)$, this leads to:

$$
\begin{aligned}
&dS_1(t)\\
&= \frac{1}{1-\lambda\big(t,S_1(t),S_2(t)\big)\frac{\partial f}{\partial s_1}}\Big\{\big(\mu_1 S_1(t) + \lambda\big(t,S_1(t),S_2(t)\big)\frac{\partial f}{\partial t} + \mu_2 S_2(t)\lambda\big(t,S_1(t),S_2(t)\big)\frac{\partial f}{\partial s_2}\\
&+ \frac{1}{2}\frac{\partial^2 f}{\partial S_1^2}\frac{1}{\big(1-\lambda\big(t,S_1(t),S_2(t)\big)\frac{\partial f}{\partial s_1}\big)^2}\Big(\sigma_1^2 S_1^2 + \sigma_2^2 S_2^2 \lambda^2\big(t,S_1(t),S_2(t)\big)\big(\frac{\partial f}{\partial s_2}\big)^2\\
&+ 2\rho\sigma_1\sigma_2 S_1(t)S_2(t)\lambda\big(t,S_1(t),S_2(t)\big)\frac{\partial f}{\partial s_2}\Big)\\
&+ \frac{\partial^2 f}{\partial s_1 \partial s_2}\frac{1}{1-\lambda\big(t,S_1(t),S_2(t)\big)\frac{\partial f}{\partial s_1}}\Big(\rho\sigma_1\sigma_2 S_1 S_2 + \sigma_2^2 S_2^2\lambda\big(t,S_1(t),S_2(t)\big)\frac{\partial f}{\partial s_2}\Big) + \frac{1}{2}\sigma_2^2 S_2^2\frac{\partial^2 f}{\partial s_2^2}\}dt\\
&+ \sigma_1 S_1(t)dW_1(t) + \sigma_2 S_2(t)\lambda\big(t,S_1(t),S_2(t)\big)\frac{\partial f}{\partial s_2}dW_2(t)\Big\}.
\end{aligned}
$$

To further simplify, we may define the functions:

$$
\begin{aligned}
\bar{\sigma}_{11}(t,s_1,s_2) &= \frac{\sigma_1}{1-\lambda(t,s_1,s_2)\frac{\partial f}{\partial s_1}},\\
\bar{\sigma}_{12}(t,s_1,s_2) &= \frac{\sigma_2\frac{s_2}{s_1}\lambda(t,s_1,s_2)\frac{\partial f}{\partial s_2}}{1-\lambda(t,s_1,s_2)\frac{\partial f}{\partial s_1}},\\
\bar{\mu}_1(t,s_1,s_2) &= \frac{1}{1-\lambda(t,s_1,s_2)\frac{\partial f}{\partial s_1}}\Big(\mu_1 + \frac{1}{s_1}\lambda(t,s_1,s_2)\frac{\partial f}{\partial t} + \mu_2\frac{s_2}{s_1}\lambda(t,s_1,s_2)\frac{\partial f}{\partial s_2}\\
&+ \frac{1}{2}\frac{\partial^2 f}{\partial s_1^2}\frac{1}{\big(1-\lambda(t,s_1,s_2)\frac{\partial f}{\partial s_1}\big)^2}\big(\sigma_1^2 s_1 + \sigma_2^2\frac{s_2^2}{s_1}\lambda^2(t,s_1,s_2)\big(\frac{\partial f}{\partial s_2}\big)^2 + 2\rho\sigma_1\sigma_2 s_2\lambda(t,s_1,s_2)\frac{\partial f}{\partial s_2}\big)\\
&+ \frac{\partial^2 f}{\partial s_1\partial s_2}\frac{1}{1-\lambda(t,s_1,s_2)\frac{\partial f}{\partial s_1}}\big(\rho\sigma_1\sigma_2 s_2 + \sigma_2^2\frac{s_2^2}{s_1}\lambda(t,s_1,s_2)\frac{\partial f}{\partial s_2}\big) + \frac{1}{2}\sigma_2^2\frac{s_2^2}{s_1}\frac{\partial^2 f}{\partial s_2^2}\Big).
\end{aligned}
$$

The simplified result of finite liquidity model takes on the form:

$$
\begin{aligned}
\frac{dS_1(t)}{S_1(t)} &= \bar{\mu}_1\big(t,S_1(t),S_2(t)\big)dt + \bar{\sigma}_{11}\big(t,S_1(t),S_2(t)\big)dW_1(t) + \bar{\sigma}_{12}\big(t,S_1(t),S_2(t)\big)dW_2(t),\\
\frac{dS_2(t)}{S_2(t)} &= \mu_2 dt + \sigma_2\rho dW_1(t) + \sigma_2\sqrt{1-\rho^2}dW_2(t), \qquad (3.1.5)\\
\frac{dD(t)}{D(t)} &= -rdt.
\end{aligned}
$$

The above SDE system can also be expressed in its canonical form:

$$
\begin{aligned}
dS_1(t) &= \bar{\mu}_1\big(t,\mathbf{S}(t)\big)S_1(t)dt + \bar{\sigma}_{11}\big(t,\mathbf{S}(t)\big)S_1(t)dW_1(t) + \bar{\sigma}_{12}\big(t,\mathbf{S}(t)\big)S_1(t)dW_2(t),\\
dS_2(t) &= \mu_2 S_2(t)dt + \sigma_2\rho S_2(t)dW_1(t) + \sigma_2\sqrt{1-\rho^2}S_2(t)dW_2(t), \qquad (3.1.6)\\
dD(t) &= -rD(t)dt.
\end{aligned}
$$

In the subsequent sections, we will discuss the establishment of a risk neutral measure as well as existence and uniqueness theorems for our market model.

3.2 Finite Liquidity Risk Neutral Measure

To price options with the finite liquidity model of (3.1.5), one needs to discover a risk-neutral measure. It turns out there is a risk-neutral measure. The results are summarized in the theorem below with an unconventional proof provided.

Theorem 3.3 (**Finite Liquidity Risk-Neutral Measure**). *There exists a unique risk-neutral measure* $\widetilde{\mathbb{P}}$ *for the finite liquidity model given by the vector-valued market price of risk generator process:*

$$\mathbf{\Theta}\big(t, S_1(t), S_2(t)\big) = \frac{1}{\sigma_2(\sqrt{1-\rho^2}\bar{\sigma}_{11} - \rho\bar{\sigma}_{12})}\begin{bmatrix}\sqrt{1-\rho^2}\sigma_2 & -\bar{\sigma}_{12} \\ -\rho\sigma_2 & \bar{\sigma}_{11}\end{bmatrix}\begin{bmatrix}\bar{\mu}_1 - r \\ \mu_2 - r\end{bmatrix}.$$

Proof: Suppose there exists an equivalent measure $\widetilde{\mathbb{P}}$ generated by some process $\mathbf{\Theta}(t)$ such that under $\widetilde{\mathbb{P}}$, $dS_1(t)$ and $dS_2(t)$ has drifts of r. Let the density process of $\widetilde{\mathbb{P}}$ with respect to $\mathbb{P}$ be:

$$Z(t) = \exp\Big(-\int_0^t \langle \mathbf{\Theta}(u), d\mathbf{W}(t)\rangle - \frac{1}{2}\int_0^t ||\mathbf{\Theta}(u)||^2 du\Big).$$

Then under $\widetilde{\mathbb{P}}$

$$d\widetilde{W}_1(t) = dW_1(t) + \Theta_1(t)dt, \quad \text{and}$$
$$d\widetilde{W}_2(t) = dW_2(t) + \Theta_2(t)dt.$$

Under $\widetilde{\mathbb{P}}$ we have the following dynamics:

$$\begin{aligned} dS_1(t) &= \Big(\bar{\mu}_1\big(t, S_1(t), S_2(t)\big) - \bar{\sigma}_{11}\big(t, S_1(t), S_2(t)\big)\Theta_1(t) \\ &\quad - \bar{\sigma}_{12}\big(t, S_1(t), S_2(t)\big)\Theta_2(t)\Big)\Big)S_1(t)dt \\ &\quad + \bar{\sigma}_{11}\big(t, S_1(t), S_2(t)\big)S_1(t)d\widetilde{W}_1(t) + \bar{\sigma}_{12}\big(t, S_1(t), S_2(t)\big)S_1(t)d\widetilde{W}_2(t), \qquad (3.2.1) \\ dS_2(t) &= \Big(\mu_2 - \rho\sigma_2\Theta_1(t) - \sqrt{1-\rho^2}\sigma_2\Theta_2(t)\Big)S_2(t)dt + \rho\sigma_2 S_2(t)d\widetilde{W}_1(t) \\ &\quad + \sqrt{1-\rho^2}\sigma_2 S_2(t)d\widetilde{W}_2(t). \end{aligned}$$

Imposing the risk-less return rate under $\widetilde{\mathbb{P}}$ leads to the following linear system:

$$\begin{bmatrix}\bar{\sigma}_{11} & \bar{\sigma}_{12} \\ \sigma_2\rho & \sigma_2\sqrt{1-\rho^2}\end{bmatrix}\begin{bmatrix}\Theta_1(t) \\ \Theta_1(t)\end{bmatrix} = \begin{bmatrix}\bar{\mu}_1 - r \\ \mu_2 - r\end{bmatrix}.$$

The system has a unique solution $\widetilde{\mathbb{P}}$ almost surely when the determinant is not zero, that is $\bar{\sigma}_{11}(t)\sigma_2\sqrt{1-\rho^2} - \bar{\sigma}_{12}(t)\sigma_2\rho \neq 0$, $\widetilde{\mathbb{P}} \otimes dt$ almost surely. Due to continuity of our processes, it will be sufficient to conclude for all t we have $\bar{\sigma}_{11}(t)\sigma_2\sqrt{1-\rho^2} - \bar{\sigma}_{12}(t)\sigma_2\rho \neq 0$, $\widetilde{\mathbb{P}}$ almost surely. Therefore, a necessary condition for the finite liquid market model to be complete is:

$$\frac{S_1(t)}{S_2(t)} \neq \frac{\sigma_2\rho}{\sigma_1\sqrt{1-\rho^2}}\lambda(t, S_1(t), S_2(t)) f_{s_2}, \quad \widetilde{\mathbb{P}} \text{ almost surely.}$$

This condition is met in light of the continuous distribution of our processes. □

Under the risk-neutral measure, the SDEs of finite liquidity market model has the following dynamics:

$$\begin{aligned}
\frac{dS_1(t)}{S_1(t)} &= rdt + \bar{\sigma}_{11}\big(t, S_1(t), S_2(t)\big)d\widetilde{W}_1(t) + \bar{\sigma}_{12}\big(t, S_1(t), S_2(t)\big)d\widetilde{W}_2(t), \\
\frac{dS_2(t)}{S_2(t)} &= rdt + \sigma_2\rho d\widetilde{W}_1(t) + \sigma_2\sqrt{1-\rho^2}d\widetilde{W}_2(t), \\
\frac{dD(t)}{D(t)} &= -rdt.
\end{aligned} \tag{3.2.2}$$

Or in its canonical form:

$$\begin{aligned}
dS_1(t) &= rS_1(t)dt + \bar{\sigma}_{11}\big(t, \mathbf{S}(t)\big)S_1(t)d\widetilde{W}_1(t) + \bar{\sigma}_{12}\big(t, \mathbf{S}(t)\big)S_1(t)d\widetilde{W}_2(t), \\
dS_2(t) &= rS_2(t)dt + \sigma_2\rho S_2(t)d\widetilde{W}_1(t) + \sigma_2\sqrt{1-\rho^2}S_2(t)d\widetilde{W}_2(t), \\
dD(t) &= -rD(t)dt.
\end{aligned} \tag{3.2.3}$$

3.3 Existence and Uniqueness of Solutions

Itô's result (1979) [30] established a method of guaranteeing the existence of a unique solution for Itô processes.

Theorem 3.1 (Itô's Existence and Uniqueness).

For $T > 0$, let $\boldsymbol{\mu}(t, \mathbf{s}) : [0, T] \times \mathbb{R}^2 \to \mathbb{R}^2$ and $\boldsymbol{\sigma}(t, \mathbf{s}) : [0, T] \times \mathbb{R}^2 \to \mathbb{R}^{2\times 2}$ be vector valued uniformly Lipschitz functions that satisfies the linear growth property. That is for some $c > 0$, we have

$$\begin{aligned}
&|\boldsymbol{\mu}(t, \mathbf{s}) - \boldsymbol{\mu}(t, \tilde{\mathbf{s}})| + |\boldsymbol{\sigma}(t, \mathbf{s}) - \boldsymbol{\sigma}(t, \tilde{\mathbf{s}})| \leqslant c|\mathbf{s} - \tilde{\mathbf{s}}|, \quad && \forall \mathbf{s}, \tilde{\mathbf{s}} \in \mathbb{R}^2, t \in [0, T], \\
&|\boldsymbol{\mu}(t, \mathbf{s})| + |\boldsymbol{\sigma}(t, \mathbf{s})| \leqslant c(1 + |\mathbf{s}|), && \forall \mathbf{s} \in \mathbb{R}^2, t \in [0, T],
\end{aligned}$$

where $|\cdot|$ *is the Euclidean norm. Then the SDE system*

$$dS_1(t) = \mu_1\big(t, \mathbf{S}(t)\big)dt + \sigma_{11}\big(t, \mathbf{S}(t)\big)dW_1(t) + \sigma_{12}\big(t, \mathbf{S}(t)\big)dW_2(t),$$
$$dS_2(t) = \mu_2\big(t, \mathbf{S}(t)\big)dt + \sigma_{21}\big(t, \mathbf{S}(t)\big)dW_1(t) + \sigma_{22}\big(t, \mathbf{S}(t)\big)dW_2(t),$$

has a unique solution with the initial value $\mathbf{S}(0) = \bar{\mathbf{s}}$.

In Chapter 5 of Oksendal (1992) [37], he mentioned when the Brownian Motion is given in advance of the solution, then the solution is call a *strong* solution. Otherwise, when we have to search for a alternative measure $\widetilde{\mathbb{P}}$ for such that Theorem 3.1 holds, then it is a *weak* solution in the $\mathbb{P}$ sense and strong in the $\widetilde{\mathbb{P}}$ sense.

The following theorem presents existence and uniqueness condition for the SDEs of our FLMM (3.1.5).

Theorem 3.3 (**Existence and Uniqueness of Finite Liquidity Market Model SDE I**)**.**

1. *Under the assumptions* (1) *to* (6) *of Section B.1, the SDE system* (3.1.5) *has a unique solution.*

2. *Under the assumptions* (1) *to* (3) *of Section B.1, the SDE system* (3.2.2) *has a unique solution.*

Proof: Please refer to B.1. □

3.4 Replicating Portfolio of the Finite Liquidity Market Model

We want to establish the methodology of option price with this model. Let $V\big(t, S_1(t), S_2(t)\big)$ be the option price at t of an option with a given payoff $V\big(T, S_1(T), S_2(T)\big)$ at the maturity T. Our goal is to characterize the function $V\big(t, s_1, s_2\big)$ through a PDE. Let $X(t)$ be a self-financing portfolio established in similar manner as section 1.4. Then, the differential of investor's portfolio is:

$$
\begin{aligned}
& dX(t) \\
& = \Delta_1(t)dS_1(t) + \Delta_2(t)dS_2(t) + r\big(X(t) - \Delta_1(t)S_1(t) - \Delta_2(t)S_2(t)\big)dt \\
& = \Delta_1(t)S_1(t)\Big(\bar{\mu}_1\big(t, S_1(t), S_2(t)\big)dt + \bar{\sigma}_{11}\big(t, S_1(t), S_2(t)\big)dW_1(t) \\
& + \bar{\sigma}_{12}\big(t, S_1(t), S_2(t)\big)dW_2(t)\Big) + \Delta_2(t)S_2(t)\Big(\mu_2 dt + \sigma_2\rho dW_1(t) + \sigma_2\sqrt{1-\rho^2}dW_2(t)\Big) \\
& + r\big(X(t) - \Delta_1(t)S_1(t) - \Delta_2(t)S_2(t)\big)dt \\
& = rX(t)dt + \Delta_1(t)\Big(\bar{\mu}_1\big(t, S_1(t), S_2(t)\big) - r\Big)S_1(t)dt + \Delta_2(t)\Big(\mu_2 - r\Big)S_2(t)dt \\
& + \Big(\Delta_1(t)\bar{\sigma}_{11}\big(t, S_1(t), S_2(t)\big)S_1(t) + \Delta_2(t)\sigma_2\rho S_2(t)\Big)dW_1(t) \\
& + \Big(\Delta_1(t)\bar{\sigma}_{12}\big(t, S_1(t), S_2(t)\big)S_1(t) + \Delta_2(t)\sigma_2\sqrt{1-\rho^2}S_2(t)\Big)dW_2(t).
\end{aligned}
$$

The differential of the discounted portfolio will become:

$$
\begin{aligned}
& d\big(D(t)X(t)\big) \\
& = D(t)\Big\{\Delta_1(t)\Big(\bar{\mu}_1\big(t, S_1(t), S_2(t)\big) - r\Big)S_1(t)dt + \Delta_2(t)\Big(\mu_2 - r\Big)S_2(t)dt \\
& + \Big(\Delta_1(t)\bar{\sigma}_{11}\big(t, S_1(t), S_2(t)\big)S_1(t) + \Delta_2(t)\sigma_2\rho S_2(t)\Big)dW_1(t) \\
& + \Big(\Delta_1(t)\bar{\sigma}_{12}\big(t, S_1(t), S_2(t)\big)S_1(t) + \Delta_2(t)\sigma_2\sqrt{1-\rho^2}S_2(t)\Big)dW_2(t)\Big\}.
\end{aligned}
$$

On the other hand, the differential of the discounted option price is:

$$
\begin{aligned}
&d\Big(D(t)V\big(t, S_1(t), S_2(t)\big)\Big) \\
&= -rD(t)V\big(t, S_1(t), S_2(t)\big)dt + D(t)\Big\{V_t dt + V_{s_1}S_1(t)\Big(\bar{\mu}_1\big(t, S_1(t), S_2(t)\big)dt \\
&+ \bar{\sigma}_{11}\big(t, S_1(t), S_2(t)\big)dW_1(t) + \bar{\sigma}_{12}\big(t, S_1(t), S_2(t)\big)dW_2(t)\Big) \\
&+ V_{s_2}S_2(t)\Big(\mu_2 dt + \sigma_2\rho dW_1(t) + \sigma_2\sqrt{1-\rho^2}dW_2(t)\Big) \\
&+ \frac{1}{2}V_{s_1s_1}\frac{1}{\big(1-\lambda\big(t, S_1(t), S_2(t)\big)\frac{\partial f}{\partial S_1}\big)^2}\Big(\sigma_1^2S_1^2(t) + \lambda^2(t, S_1(t), S_2(t))\big(\frac{\partial f}{\partial S_2}\big)^2\sigma_2^2S_2^2(t) \\
&+ 2\lambda\big(t, S_1(t), S_2(t)\big)\frac{\partial f}{\partial S_2}\rho\sigma_1\sigma_2S_1(t)S_2(t)\Big)dt + \frac{1}{2}V_{s_2s_2}\sigma_2^2S_2^2(t)dt \\
&+ V_{s_1s_2}\frac{1}{1-\lambda\big(t, S_1(t), S_2(t)\big)\frac{\partial f}{\partial S_1}}\Big(\rho\sigma_1\sigma_2S_1(t)S_2(t) + \sigma_2^2S_2^2(t)\lambda\big(t, S_1(t), S_2(t)\big)\frac{\partial f}{\partial S_2}\Big)dt\Big\} \\
&= D(t)\Big\{-rV(t, S_1(t), S_2(t)) + V_t + V_{s_1}\bar{\mu}_1\big(t, S_1(t), S_2(t)\big)S_1(t) + V_{s_2}\mu_2S_2(t) \\
&+ \frac{1}{2}V_{s_1s_1}\frac{1}{\big(1-\lambda\big(t, S_1(t), S_2(t)\big)\frac{\partial f}{\partial S_2}\big)^2}\Big(\sigma_1^2S_1^2 + \lambda^2\big(t, S_1(t), S_2(t)\big)\big(\frac{\partial f}{\partial S_1}\big)^2\sigma_2^2S_2^2(t) \\
&+ 2\lambda\big(t, S_1(t), S_2(t)\big)\frac{\partial f}{\partial S_2}\rho\sigma_1\sigma_2S_1(t)S_2(t)\Big) + \frac{1}{2}V_{s_2s_2}\sigma_2^2S_2^2(t) \\
&+ V_{s_1s_2}\frac{1}{1-\lambda\big(t, S_1(t), S_2(t)\big)\frac{\partial f}{\partial S_1}}\Big(\rho\sigma_1\sigma_2S_1(t)S_2(t) + \sigma_2^2S_2^2(t)\lambda\big(t, S_1(t), S_2(t)\big)\frac{\partial f}{\partial S_2}\Big)\Big\}dt \\
&+ D(t)\Big(V_{s_1}\bar{\sigma}_{11}\big(t, S_1(t), S_2(t)\big)S_1(t) + V_{s_2}\sigma_2\rho S_2(t)\Big)dW_1(t) \\
&+ D(t)\Big(V_{s_1}\bar{\sigma}_{12}\big(t, S_1(t), S_2(t)\big)S_1(t) + V_{s_2}\sigma_2\sqrt{1-\rho^2}S_2(t)\Big)dW_2(t).
\end{aligned}
$$

By equating the diffusion terms of $d\big(D(t)X(t)\big)$ and $d\Big(D(t)V\big(t, S_1(t), S_2(t)\big)\Big)$, the Deltas for hedging can be concluded to be $\Delta_1(t) = V_{s_1}$ and $\Delta_2(t) = V_{s_2}$.

Next, equate the drift terms of $d\big(D(t)X(t)\big)$ and $d\Big(D(t)V\big(t, S_1(t), S_2(t)\big)\Big)$ to obtain the PDE of the finite liquidity market model:

$$
\begin{aligned}
rV(t, s_1, s_2) = V_t + rs_1V_{s_1} + rs_2V_{s_2} &+ \frac{1}{2}V_{s_1s_1}\Big(\frac{1}{\big(1-\lambda(t, s_1, s_2)\frac{\partial f}{\partial s_1}\big)^2}\Big)\Big(\sigma_1^2s_1^2 \\
&+ \lambda^2(t, s_1, s_2)\big(\frac{\partial f}{\partial s_2}\big)^2\sigma_2^2s_2^2 + 2\lambda(t, s_1, s_2)\frac{\partial f}{\partial s_2}\rho\sigma_1\sigma_2s_1s_2\Big) \qquad (3.4.1) \\
&+ V_{s_1s_2}\Big(\frac{1}{1-\lambda(t, s_1, s_2)\frac{\partial f}{\partial s_1}}\Big)\Big(\rho\sigma_1\sigma_2s_1s_2 + \lambda(t, s_1, s_2)\frac{\partial f}{\partial s_2}\sigma_2^2s_2^2\Big) + \frac{1}{2}V_{s_2s_2}\sigma_2^2s_2^2,
\end{aligned}
$$

with $0 < s_1, s_2 < \infty$, $0 \leqslant t \leqslant T$, and the terminal condition:

$$V(T, s_1, s_2) = h(s_1, s_2),$$

where $h(s_1, s_2)$ is the pay off function of any Two-Asset option.

Just like the BS PDE in (1.4.3), the FLMM PDE of (3.4.1) is also parabolic in nature. Once again we can refer to Chapter 4 of Friedman [18] for existence and uniqueness of this family of PDEs.

3.5 Feynman-Kac Formula of Finite Liquidity Market Model

The Feynman-Kac formula of finite liquidity market is similar to the one in section 1.6. For any payoff function $h()$ of an option, given the stock values at time t, $S_1(t) = s_1$ and $S_2(t) = s_2$ where $t \in [0, T]$. Let $V(t, s_1, s_2)$ be a solution of the finite liquidity partial differential equation (3.4.1) with the terminal condition $V(T, s_1, s_2) = h(s_1, s_1)$. Then $V(t, s_1, s_2)$ will satisfy the the conditional expectation:

$$V(t, s_1, s_2) = \widetilde{\mathbb{E}}[e^{-r(T-t)}h\big(S_1(T), S_2(T)\big)|S_1(T) = s_1, S_2(T) = s_2].$$

In this case, $e^{-rt}V\big(t, S_1(t), S_2(t)\big)$ is a martingale.

3.6 Risk-Neutral Formula of Finite Liquidity Market Model

In the previous sections, the delta hedging strategy as well as Feynman-Kac formula was established for the finite liquidity model. Now, define the risk-neutral pricing formula to be:

$$V\big(t, s_1, s_2\big) = \widetilde{\mathbb{E}}[e^{-r(T-t)}V\big(T, S_1(T), S_2(T)\big)|\mathscr{F}(t)]. \tag{3.6.1}$$

Moreover, $V\big(t, s_1, s_2\big)$ must satisfy the finite liquidity PDE (3.4.1). This shows the finite liquidity PDE correctly prices any option that is a function of $S_1(t)$ and $S_2(t)$, for all $t \in [0, T]$.

Chapter 4

Numerical Methods I - Exchange Option

4.1 Impact of Delta Hedgers, Existence and Uniqueness

Delta hedging is a strategy adopted by many big financial institutions to reduce their option portfolio's exposure against movements in the underlying assets. By assuming majority of the market participates are delta hedgers, then the trading strategy function of (3.1.1) would be equal to the Exchange Option's Delta of asset one. That is, $f(t, s_1, s_2) = \Delta_1(t) = N(d_+)$. As a result, the finite liquidity market model will have the following dynamics:

$$\begin{aligned}
\frac{dS_1(t)}{S_1(t)} &= \bar{\mu}_1\big(t, S_1(t), S_2(t)\big)dt + \bar{\sigma}_{11}\big(t, S_1(t), S_2(t)\big)dW_1(t) + \bar{\sigma}_{12}\big(t, S_1(t), S_2(t)\big)dW_2(t),\\
\frac{dS_2(t)}{S_2(t)} &= \mu_2 dt + \bar{\sigma}_{21} dW_1(t) + \bar{\sigma}_{21} dW_2(t), \qquad (4.1.1)\\
\frac{dD(t)}{D(t)} &= -r dt,
\end{aligned}$$

where

$$\begin{aligned}
\bar{\mu}_1(t, s_1, s_2) &= \frac{1}{1-\lambda(t,s_1)\Gamma_{11}(t)}\Big(\mu_1(t) + \frac{1}{s_1}\lambda\big(t,s_1\big)\Theta(t) + \mu_2(t)\frac{s_2}{s_1}\lambda(t,s_1)\Gamma_{12}(t)\\
&+ \frac{1}{2}Speed_{111}(t)\frac{1}{\big(1-\lambda(t,s_1)\Gamma_{11}(t)\big)^2}\Big(\sigma_1^2 s_1 + \sigma_2^2\frac{s_2^2}{s_1}\lambda^2\big(t,s_1\big)\big(\Gamma_{12}(t)\big)^2 + 2\rho\sigma_1\sigma_2 s_2\lambda(t,s_1)\Gamma_{12}(t)\Big)\\
&+ Speed_{112}(t)\frac{1}{1-\lambda(t,s_1)\Gamma_{11}(t)}\big(\rho\sigma_1\sigma_2 s_2 + \sigma_2^2\frac{s_2^2}{s_1}\lambda(t,s_1)\Gamma_{12}(t)\big) + \frac{1}{2}\sigma_2^2\frac{s_2^2}{s_1}Speed_{122}(t)\Big),
\end{aligned}$$

$$\bar{\sigma}_{11}(t,s_1,s_2) = \frac{\sigma_1}{1-\lambda(t,s_1)\Gamma_{11}(t)}, \qquad \bar{\sigma}_{12}(t,s_1,s_2) = \frac{\sigma_2 s_2 \lambda(t,s_1)\Gamma_{12}(t)}{s_1\Big(1-\lambda(t,s_1)\Gamma_{11}(t)\Big)},$$
$$\bar{\sigma}_{21} = \sigma_2\rho, \qquad \bar{\sigma}_{22} = \sigma_2\sqrt{1-\rho^2}.$$

There had been many past studies done to model the price impact from trading. For example, Liu and Yong (2005) [33] studied a price impact model for a FBSDE system. Pirvu et al. (2014) [1] also studied a price impact model for spread option. This thesis will adopt a slightly different price impact model. The price impact function will approximately be:

$$\bar{\lambda}\big(t,s_1\big) = \begin{cases} \epsilon\big(1-e^{-\beta(T-t)^{\frac{3}{2}}}\big) & \text{if} \quad \underline{S_1} \leqslant s_1 \leqslant \overline{S}_1, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1.2}$$

In this impact function, $\underline{S_1}$ and $\overline{S}_1$ represents trading floor and cap of the asset respectively. Only within this range there will be trading price impact. ϵ is the price impact per share, and β is a decaying constant. For the rest of this chapter, $\underline{S_1}$ and $\overline{S}_1$ will be set to 60% and 140% of the current fair market value of the asset. ϵ will be set to 0.04, and β will be set to 100. It is important to emphasize $\bar{\lambda}\big(t,S_1(t)\big)$ will be employed for numerical approximation. The theoretical $\lambda\big(t,S_1(t)\big)$ should be a function with bounded derivative, that is obtained through a standard mollifying of $\bar{\lambda}\big(t,S_1(t)\big)$. For the remainder of this chapter, the trading strategy and price impact functions will be assumed to be $\Delta_1(t)$ and $\lambda\big(t,S_1(t)\big)$ respectively.

Theorem 4.1 (**Existence and Uniqueness of Finite Liquidity Market Model SDE II**).

The SDE system (4.1.1) *has a strong solution.*

Proof: Please refer to Appendix B.2 for the proof. □

After establishing existence and uniqueness with the above theorem, we can easily observe the risk-neutral dynamics of FLMM:

$$\begin{aligned} dS_1(t) &= rS_1(t)dt + \widetilde{\sigma}_{11}\big(t,S_1(t),S_2(t)\big)d\widetilde{W}_1(t) + \widetilde{\sigma}_{12}\big(t,S_1(t),S_2(t)\big)d\widetilde{W}_2(t), \\ dS_2(t) &= rS_2(t)dt + \widetilde{\sigma}_{21}(t)d\widetilde{W}_1(t) + \widetilde{\sigma}_{22}(t)d\widetilde{W}_2(t), \\ \frac{dD(t)}{D(t)} &= -rdt, \end{aligned} \tag{4.1.3}$$

where as a new naming convention, we have:

$$\widetilde{\sigma}_{11}(t,s_1,s_2) = \frac{\sigma_1 s_1}{1-\lambda\Gamma_{11}}, \qquad \widetilde{\sigma}_{12}(t,s_1,s_2) = \frac{\sigma_2 s_2 \lambda\Gamma_{12}}{1-\lambda\Gamma_{11}}.$$
$$\widetilde{\sigma}_{21}(t) = \sigma_2 s_2 \rho, \qquad \widetilde{\sigma}_{22}(t) = \sigma_2 s_2\sqrt{1-\rho^2}.$$

4.2 Numerical Solutions of Finite Liquidity Market Model

In financial mathematics, we often use numerical method for simulation of SDEs. Sometimes we need to simulate the processes' entire path, this is the case for path dependent options. In our case, it is because the FLMM SDE do not have a closed form solution. In the subsequent sections, we shall employ two different simulation methods.

4.2.1 Two-Dimensional Euler-Maruyama Simulation

Euler-Maruyama is a method designed for solving stochastic differential equations numerically. It is named after Leonhard Euler and Gisiro Maruyama. E-M method has ample application in the field of financial mathematics. The motivation behind the method is to use Iȋto Taylor expansion, then only use the first order terms in the discretization. Let a system of stochastic differential equation be of the form:

$$\begin{aligned} dX_1(t) &= \mu_1(t, X_1, X_2)dt + \sigma_{11}(t, X_1, X_2)dW_1(t) + \sigma_{12}(t, X_1, X_2)dW_2(t), \\ dX_2(t) &= \mu_2(t, X_1, X_2)dt + \sigma_{21}(t, X_1, X_2)dW_1(t) + \sigma_{22}(t, X_1, X_2)dW_2(t). \end{aligned} \tag{4.2.1}$$

Suppose the system of SDEs has initial conditions $X_1(t_0) = x_1$ and $X_2(t_0) = x_2$ on the time interval $[t_0, T]$. Integrating both side of the system of SDEs (4.2.1):

$$\begin{aligned} X_1(t) &= X_1(t_0) + \int_{t_0}^{t} \mu_1(u, X_1, X_2)du + \int_{t_0}^{t} \sigma_{11}(u, X_1, X_2)dW_1(u) \\ &\quad + \int_{t_0}^{t} \sigma_{12}(u, X_1, X_2)dW_2(u), \\ X_2(t) &= X_2(t_0) + \int_{t_0}^{t} \mu_2(u, X_1, X_2)du + \int_{t_0}^{t} \sigma_{21}(u, X_1, X_2)dW_1(u) \\ &\quad + \int_{t_0}^{t} \sigma_{22}(u, X_1, X_2)dW_2(u). \end{aligned} \tag{4.2.2}$$

The Euler-Maruyama approximation is the first order discretization of the stochastic integral. Let $Y_1(m)$ and $Y_2(m)$ be the discretization of (4.2.2), then E-M can be set up through the following these procedures:

1. Partition $[t_0, T]$ into M equivalent intervals of length $\Delta t = \frac{T-t_0}{M}$.

2. Set $Y_1(0) = x_1$ and $Y_2(0) = x_2$.

3. Sample $\{\Delta W_i(j)\}_{i=1,2;j=1,2,\dots M}$, where each $\Delta W_i(j) \sim N(0, \Delta t)$.

4. Recursively define:

$$\begin{aligned}
Y_1(m+1) &= Y_1(m) + \mu_1\big(m, Y_1(m), Y_2(m)\big)\Delta t + \sigma_{11}\big(m, Y_1(m), Y_2(m)\big)\Delta W_1(m+1) \\
&\quad + \sigma_{12}\big(m, Y_1(m), Y_2(m)\big)\Delta W_2(m+1), \\
Y_2(m+1) &= Y_2(m) + \mu_2\big(m, Y_1(m), Y_2(m)\big)\Delta t + \sigma_{21}\big(m, Y_1(m), Y_2(m)\big)\Delta W_1(m+1) \\
&\quad + \sigma_{22}\big(m, Y_1(m), Y_2(m)\big)\Delta W_2(m+1).
\end{aligned}$$

Of course, M has to be sufficiently large for $Y_1(i)$ and $Y_2(i)$ to converge to $X_1(t)$ and $X_2(t)$ respective. Platen (1999) [39] defined strong and weak convergence for simulated approximation of SDEs.

Let $Y(i)$ be a discrete time simulation of the exact solution $X(t)$ of a SDE on $[t_0, T]$. $Y(t)$ is convergent to $X(t)$ with order α if there exists a constant C or C_p, such that:

$$\begin{aligned}
&\mathbb{E}\Big[\sup_{t_0 \leqslant t_i \leqslant T} \big|X(t_i) - Y(i)\big|\Big] \leqslant C(\Delta t)^{\alpha}, \qquad \text{(strong sense)} \\
&\Big|\mathbb{E}\big[p\big(X(T)\big)\big] - \mathbb{E}\big[p\big(Y(M)\big)\big]\Big| \leqslant C_p(\Delta t)^{\alpha}, \qquad \text{(weak sense)}
\end{aligned}$$

here $p(x)$ is any polynomial.

Desmond (2015) [25] mentioned that the E-M algorithm is strongly convergent with order $\frac{1}{2}$ and weakly convergent with order 1. If 2ϵ is the width of the confidence interval where the true parameter of a Monte Carlo estimation resides, then this accuracy may be achieved at a combined complexity cost of $\mathcal{O}(\epsilon^{-3})$.

Going back to the finite liquidity market model (4.1.3), the asset processes can be simulated through E-M with the following algorithm:

1. Partition $[t, T]$ into M equivalent intervals of length $\Delta t = \frac{T-t}{M}$.

2. Set the initial values as $S_1(0) = s_1$ and $S_2(0) = s_2$.

3. Sample $\{\Delta W_1(j), \Delta W_2(j)\}_{j=1,2,\dots M}$, where each $\{\Delta W_1(j), \Delta W_2(j)\} \sim \mathcal{N}_2(\mathbf{0}, \Delta t I_2)$.

4. Recursively define:

$$S_1(m+1) = S_1(m) + rS_1(m)\Delta t + \sum_{i=1}^{2} \widetilde{\sigma}_{1i}\big(\mathbf{S}(m)\big)\Delta W_i(m+1), \tag{4.2.3}$$

$$S_2(m+1) = S_2(m) + rS_2(m)\Delta t + \sum_{i=1}^{2} \widetilde{\sigma}_{2i}\big(\mathbf{S}(m)\big)\Delta W_i(m+1).$$

We may redefine a matrix recursion version of the Milstein Scheme. Consider the following evolutionary dynamic of $\mathbf{S}(t)$:

$$\mathbf{S}(m+1) = \mathbf{B}(m)\mathbf{S}(m). \tag{4.2.4}$$

The matrix $\mathbf{B}(m)$ consists of the first order approximation and the vector $\mathbf{b}(m)$ is the second order approximation. For our SDE system (4.1.3), $\mathbf{B}(m)$ and $\mathbf{b}(m)$ can be defined as follows:

$$\mathbf{B}(m) = \begin{bmatrix} 1 + r\Delta t + \widetilde{\sigma}_{11}\big(\mathbf{S}(m)\big)\Delta W_1(m+1) & \widetilde{\sigma}_{12}\big(\mathbf{S}(m)\big)\Delta W_2(m+1) \\ \widetilde{\sigma}_{21}\big(\mathbf{S}(m)\big)\Delta W_1(m+1) & 1 + r\Delta t + \widetilde{\sigma}_{21}\big(\mathbf{S}(m)\big)\Delta W_2(m+1) \end{bmatrix}.$$

Algorithm 4.2.1 Euler-Maruyama

Initialize Values $\mathbf{S}(t) = \mathbf{s}$
Define $\Delta t =: \frac{T-t}{M}$
for $m = 0$ **to** $M-1$ **do**
  $\mathbf{\Delta W}(m) = \big(\Delta w_1(m), \Delta w_2(m)\big) \sim \mathcal{N}_2(0, \Delta t I_2)$
  Set $\mathbf{B}(m)$
  $\mathbf{S}(m+1) = \mathbf{B}(m)\mathbf{S}(m)$
end for
return $\mathbf{S}(M)$

The complete code of all algorithms of this chapter are made available at: https://github.com/ShuAiii/FLMMExchange.

4.2.2 Two-Dimensional Milstein Simulation

E-M algorithm from the previous section is a first order approximation of SDE's solution. Mil'shtein G. N. (1975) [36] created the Milstein method which is a second order approximation, that is the second order terms from Itô Taylor expansion will be kept in the discretization scheme. Consider an approximation of the diffusion function term:

$$\sigma_{ij}\big(t, X_1(t), X_2(t)\big) \approx \sigma_{ij}\big(t_0, X_1(t_0), X_2(t_0)\big) + \sum_{k,l=1}^{2} \frac{\partial \sigma_{ij}}{\partial x_l} \sigma_{lk}\big(t_0, X_1(t_0), X_2(t_0)\big) W_k(t),$$

substituting the above approximation into (4.2.2) yields:

$$
\begin{aligned}
X_1(t) &\approx X_1(t_0) + \int_{t_0}^{t} \mu_1(u, X_1, X_2)du + \int_{t_0}^{t} \sigma_{11}(u, X_1, X_2)dW_1(u) \\
&\quad + \int_{t_0}^{t} \sigma_{12}(u, X_1, X_2)dW_2(u) + \sum_{j,k,l=1}^{2} \frac{\partial \sigma_{1j}}{\partial x_l}\sigma_{lk}\big(t_0, X_1(t_0), X_2(t_0)\big) \int_{t_0}^{t} W_k(u)dW_j(u), \\
X_2(t) &\approx X_2(t_0) + \int_{t_0}^{t} \mu_2(u, X_1, X_2)du + \int_{t_0}^{t} \sigma_{21}(u, X_1, X_2)dW_1(u) \\
&\quad + \int_{t_0}^{t} \sigma_{22}(u, X_1, X_2)dW_2(u) + \sum_{j,k,l=1}^{2} \frac{\partial \sigma_{2j}}{\partial x_l}\sigma_{lk}\big(t_0, X_1(t_0), X_2(t_0)\big) \int_{t_0}^{t} W_k(u)dW_j(u).
\end{aligned}
$$

By using integral Itô product rule, we have:

$$
W_j(t)W_k(t) - W_j(t_0)W_k(t_0) = \int_{t_0}^{t} W_j(u)dW_k(u) + \int_{t_0}^{t} W_k(u)dW_j(u) + \rho_{jk}(t - t_0).
$$

If we let

$$
A_{ij}(t, t_0) = \int_{t_0}^{t} \Big(\big(W_i(u) - W_i(t_0)\big)dW_j(u) - \big(W_j(u) - W_j(t_0)\big)dW_i(u)\Big), \tag{4.2.1}
$$

then

$$
\int_{t_0}^{t} W_k(u)dW_j(u) = \frac{1}{2}\big(W_j(t)W_k(t) - W_j(t_0)W_k(t_0) + \rho_{jk}(t - t_0) - A_{ij}(t, t_0)\big).
$$

Substituting the above expression into (4.2.2) yields

$$
\begin{aligned}
X_1(t) &\approx X_1(t_0) + \int_{t_0}^{t} \mu_1(u, X_1, X_2)du + \int_{t_0}^{t} \sigma_{11}(u, X_1, X_2)dW_1(u) + \int_{t_0}^{t} \sigma_{12}(u, X_1, X_2)dW_2(u) \\
&\quad + \frac{1}{2}\sum_{j,k,l=1}^{2} \frac{\partial \sigma_{1j}}{\partial x_l}\sigma_{lk}\big(t_0, X_1, X_2\big)\big(W_j(t)W_k(t) - W_j(t_0)W_k(t_0) + \rho_{jk}(t - t_0) - A_{ij}(t, t_0)\big), \\
X_2(t) &\approx X_2(t_0) + \int_{t_0}^{t} \mu_2(u, X_1, X_2)du + \int_{t_0}^{t} \sigma_{21}(u, X_1, X_2)dW_1(u) + \int_{t_0}^{t} \sigma_{22}(u, X_1, X_2)dW_2(u) \\
&\quad + \frac{1}{2}\sum_{j,k,l=1}^{2} \frac{\partial \sigma_{2j}}{\partial x_l}\sigma_{lk}\big(t_0, X_1, X_2\big)\big(W_j(t)W_k(t) - W_j(t_0)W_k(t_0) + \rho_{jk}(t - t_0) - A_{ij}(t, t_0)\big).
\end{aligned}
$$

The expression (4.2.1) is the Lévy Area between two BMs [21]. There are many techniques to generate the Lévy Area, one of the easiest method is to just generate the bi-linear form piece by piece. This thesis will provide an algorithm that can be found in a paper by Scheicher (2007) [43].

Algorithm 4.2.2 Lévy Area

Define sub-partition length $\Delta^2 t := \frac{\Delta t}{K}$
Generate $\mathbf{z}_1, \mathbf{z}_2 \sim \mathcal{N}_K(\mathbf{0}, I_K)$.
Generate lower triangular matrix of 1s T, set $R := \Delta^2 t T$
Generate lower and upper diagonal matrices of 1s L and U of 1s.
Set $\mathbf{B}_1 := R\mathbf{z}_1$ and $\mathbf{B}_2 =: R\mathbf{z}_2$
$A = \mathbf{b_1}^T(U-L)\mathbf{b_2}$
return A

According to Scheicher, this algorithm for Lévy Area has complexity of $\mathcal{O}(K)$.

The Milstein approximation for the system of SDEs in (4.2.1) can be set up by following these procedures:

1. Partition $[t_0, T]$ into M equivalent intervals of length $\Delta t = \frac{T-t_0}{M}$.
2. Set $Y_1(0) = x_1$ and $Y_2(0) = x_2$.
3. Sample $\{\Delta W_i(j)\}_{i=1,2;j=1,2,\ldots M}$, where each $\Delta W_i(j) \sim N(0, \Delta t)$.
4. Generate $A_{ij}(T,t)$ with $\{\Delta W_i(j)\}_{i=1,2;j=1,2,\ldots M}$.
5. Recursively define:

$$\begin{aligned}
Y_1(m+1) &= Y_1(m) + \mu_1\big(m, Y_1(m), Y_2(m)\big)\Delta t + \sigma_{11}\big(m, Y_1(m), Y_2(m)\big)\Delta W_1(m+1) \\
&\quad + \sigma_{12}\big(m, Y_1(m), Y_2(m)\big)\Delta W_2(m+1) + \frac{1}{2}\sum_{i,j,k=1}^{2} \frac{\partial \sigma_{1i}}{\partial y_k}\sigma_{kj}\big(m, Y_1(m), Y_2(m)\big) \\
&\quad \times \big(\Delta W_i(m+1)\Delta W_j(m+1) - \mathbb{1}_{(i=j)}\Delta t - A_{ij}(0, \Delta t)\big), \\
Y_2(m+1) &= Y_2(m) + \mu_2\big(m, Y_1(m), Y_2(m)\big)\Delta t + \sigma_{21}\big(m, Y_1(m), Y_2(m)\big)\Delta W_1(m+1) \\
&\quad + \sigma_{22}\big(m, Y_1(m), Y_2(m)\big)\Delta W_2(m+1) + \frac{1}{2}\sum_{i,j,k=1}^{2} \frac{\partial \sigma_{2i}}{\partial y_k}\sigma_{kj}\big(m, Y_1(m), Y_2(m)\big) \\
&\quad \times \big(\Delta W_i(m+1)\Delta W_j(m+1) - \mathbb{1}_{(i=j)}\Delta t - A_{ij}(0, \Delta t)\big).
\end{aligned}$$

Desmond (2015) [25] mentioned that the Milstein has the same weak form convergence rate as E-M. On the other hand, Milstein has a strong form convergence rate of $\alpha = 1$, which is superior than E-M. Desmond also mentioned, for a 2ϵ confidence interval, the MC estimator that use Milstein simulation will have an $\mathcal{O}(\epsilon^{-2})$ complexity cost due to its faster convergence rate. For the system of GBMs in the standard BSM model of (1.3.1), a comparison of E-M and Milstein is available in 4.2.1 and 4.2.2. The simulation algorithms are available at 4.2.1 and 4.2.3.

Figure 4.2.1: E-M vs Milstein for $M = 100$

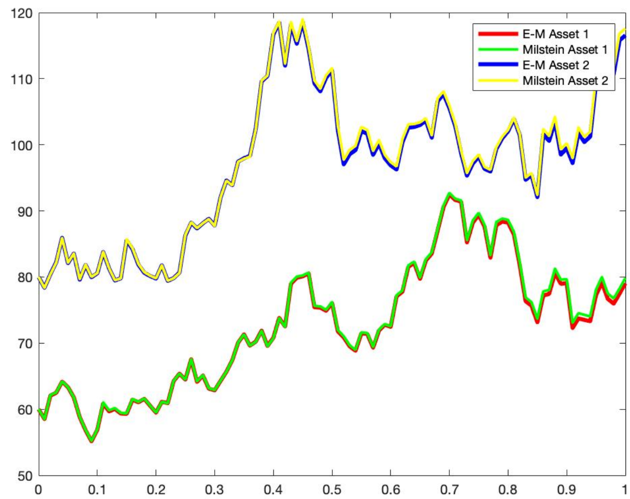
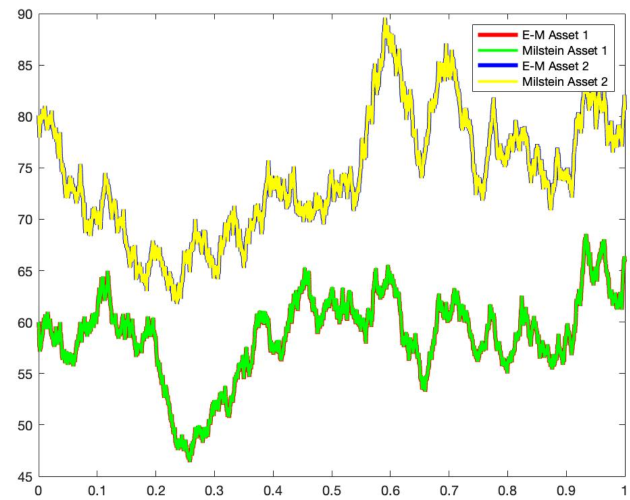


Figure 4.2.2: E-M vs Milstein for $M = 10000$



Note: the GBM parameters used are $S_1(0) = 60$, $S_2(0) = 80$, $T = 1$, $\sigma_1 = 0.3$, $\sigma_2 = 0.3$, $\rho = 0.5$ and $r = 0.05$.

Going back to the FLMM under risk-neutral measure (4.1.3), its Milstein approximation can be set up by following these procedures:

1. Partition $[t, T]$ into M equivalent intervals of length $\Delta t = \frac{T-t}{M}$.

2. Set the initial values as $S_1(0) = s_1$ and $S_2(0) = s_2$.

3. Sample $\{\Delta W_1(j), \Delta W_2(j)\}_{j=1,2,\ldots M}$, where each $\{\Delta W_1(j), \Delta W_2(j)\} \sim \mathcal{N}_2(\mathbf{0}, \Delta t I_2)$.

4. Generate Lévy Areas $\mathcal{A}_{ij}(0, \Delta t)$.

5. Recursively define:

$$\begin{aligned} S_1(m+1) &= S_1(m) + rS_1(m)\Delta t + \sum_{i=1}^{2} \widetilde{\sigma}_{1i}\big(\mathbf{S}(m)\big)\Delta W_i(m+1) + \frac{1}{2}\sum_{i,j,k=1}^{2}\frac{\partial \widetilde{\sigma}_{1i}}{\partial s_k} \\ &\quad \times \widetilde{\sigma}_{kj}\big(\mathbf{S}(m)\big)\big(\Delta W_i(m+1)\Delta W_j(m+1) - \mathbb{1}_{(i=j)}\Delta t - \mathcal{A}_{ij}\big), \qquad (4.2.2) \\ S_2(m+1) &= S_2(m) + rS_2(m)\Delta t + \sum_{i=1}^{2} \widetilde{\sigma}_{2i}\big(\mathbf{S}(m)\big)\Delta W_i(m+1) + \frac{1}{2}\sum_{i,j,k=1}^{2}\frac{\partial \widetilde{\sigma}_{2i}}{\partial s_k} \\ &\quad \times \widetilde{\sigma}_{kj}\big(\mathbf{S}(m)\big)\big(\Delta W_i(m+1)\Delta W_j(m+1) - \mathbb{1}_{(i=j)}\Delta t - \mathcal{A}_{ij}\big). \end{aligned}$$

We may redefine a matrix recursion version of the Milstein Scheme. Consider the following evolutionary dynamic of $\mathbf{S}(t)$:

$$\mathbf{S}(m+1) = \mathbf{B}(m)\mathbf{S}(m) + \frac{1}{2}\mathbf{b}(m). \qquad (4.2.3)$$

The matrix $\mathbf{B}(m)$ consists of the first order approximation and the vector $\mathbf{b}(m)$ is the second order approximation. For our SDE system (4.1.3), $\mathbf{B}(m)$ and $\mathbf{b}(m)$ can be defined as follows:

$$\begin{aligned} \mathbf{B}(m) &= \begin{bmatrix} 1 + r\Delta t + \widetilde{\sigma}_{11}\big(\mathbf{S}(m)\big)\Delta W_1(m+1) & \widetilde{\sigma}_{12}\big(\mathbf{S}(m)\big)\Delta W_2(m+1) \\ \widetilde{\sigma}_{21}\big(\mathbf{S}(m)\big)\Delta W_1(m+1) & 1 + r\Delta t + \widetilde{\sigma}_{21}\big(\mathbf{S}(m)\big)\Delta W_2(m+1) \end{bmatrix}, \\ \mathbf{b}(m) &= \begin{bmatrix} \mathbf{W}^T(m+1)J_1\Sigma\mathbf{W}(m+1) - tr(J_1\Sigma) - \mathbf{1}^T(J_1\Sigma \circ \mathcal{A})\mathbf{1} \\ \mathbf{W}^T(m+1)J_2\Sigma\mathbf{W}(m+1) - tr(J_2\Sigma) - \mathbf{1}^T(J_2\Sigma \circ \mathcal{A})\mathbf{1} \end{bmatrix}. \end{aligned}$$

Here J_i is the Jacobi matrix of the i-th asset's diffusion functions at the m-th step. Matrix Σ encapsulates diffusion functions of all assets, also at m-th step. They are of the form:

$$J_i = \begin{bmatrix} \frac{\partial \widetilde{\sigma}_{i1}}{\partial s_1} & \frac{\partial \widetilde{\sigma}_{i1}}{\partial s_2} \\ \frac{\partial \widetilde{\sigma}_{i2}}{\partial s_1} & \frac{\partial \widetilde{\sigma}_{i2}}{\partial s_2} \end{bmatrix}, \qquad \Sigma = \begin{bmatrix} \widetilde{\sigma}_{11} & \widetilde{\sigma}_{12} \\ \widetilde{\sigma}_{21} & \widetilde{\sigma}_{22} \end{bmatrix}.$$

$\mathcal{A}$ is the matrix of Lévy Areas at step m, it has the form:

$$\mathcal{A} = \begin{bmatrix} 0 & \mathcal{A}_{12} \\ \mathcal{A}_{21} & 0 \end{bmatrix},$$

notice $\mathcal{A}$ is an off diagonal matrix, this is because the stochastic integral (4.2.1) is 0 when $i = j$.

Algorithm 4.2.3 Milstein

Initialize Values $\mathbf{S}(t) = \mathbf{s}$
Define $\Delta t =: \frac{T-t}{M}$
for $m = 0$ **to** $M - 1$ **do**
  $\mathbf{\Delta W}(m) = \big(\Delta w_1(m), \Delta w_2(m)\big) \sim \mathcal{N}_2(0, \Delta t I_2)$
  Set $\mathbf{B}(m)$, $\mathbf{b}(m)$, J_i, Σ and $\mathcal{A}$
  $\mathbf{S}(m+1) = \mathbf{B}(m)\mathbf{S}(m) + \frac{1}{2}\mathbf{b}(m)$
end for
return $\mathbf{S}(M)$

4.3 Monte Carlo Methods for Exchange Option Value

Monte Carlo is a method that takes advantage of a large quantity of random samples for the purpose of estimation. In the reminder of this chapter, we will use various Monte Carlo schemes to estimate the price of Exchange Option under FLMM. We will also deploy various variance reduction techniques to get the best variance vs compute time trade off. The objective function is:

$$V(t, s_1, s_2) = \widetilde{\mathbb{E}}[e^{-r(T-t)}\big(S_1(T) - S_2(T)\big)^+ | S_1(T) = s_1, S_2(T) = s_2],$$

This is the result of the discounted Feynman-Kac formula of Section 3.5, with the payoff function $h(s_1, s_2) = (s_1 - s_2)^+$.

4.3.1 Naive Estimator

For any option that has payout $V\big(T, S_1(T), S_2(T)\big)$ with current asset prices $S_1(t) = s_1$ and $S_2(t) = s_2$, define a naive Monte Carlo estimator for the current option value to be:

$$\theta_n^{(ex)} = \frac{e^{-r\tau}}{N}\sum_{i=1}^{N}\big(S_1^{(i)}(T) - S_2^{(i)}(T)\big)^+. \tag{4.3.1}$$

Of course, this estimator is unbiased and converge to $V(t, s_1, s_2)$ almost surely when $n \to \infty$. There is a trade off between variance and unbiasedness when building an estimator. Ideally, a Monte Carlo estimator should chosen to be unbiased and then optimized for its variance and sampling complexity.

Under the finite liquidity framework of (4.1.1), exchange option can be priced by applying the MC estimator of (4.3.1). The following algorithm computes the mean, variance and confidence interval of the estimator:

Algorithm 4.3.1 Naive Monte Carlo

Sample N paths of $\mathbf{S}_1$ and $\mathbf{S}_2$
$\mathbf{S}_1, \mathbf{S}_2 = [\vec{S}_1^{(1)}, \vec{S}_1^{(2)}, ..., \vec{S}_1^{(N)}; \vec{S}_2^{(1)}, \vec{S}_2^{(2)}, ..., \vec{S}_2^{(N)}]$
Estimate the Mean:
$\bar{\mu}_n^{(ex)} = \frac{1}{N}\sum_{i=1}^{N}\left(\left(S_1^{(i)}(T) - S_2^{(i)}(T)\right)^+\right)$
Option Value:
$\theta_n^{(ex)} = e^{-r\tau}\bar{\mu}_n^{(ex)}$
Variance of $\theta_n^{(ex)}$
$\bar{\sigma}^2 = \frac{e^{-2r\tau}}{N}\sum_{i=1}^{N}\left(\left(S_1^{(i)}(T) - S_2^{(i)}(T)\right)^+ - \bar{\mu}_n^{(ex)}\right)^2$
$1-\alpha$ Confidence Interval of the Value
$CI^{(ex)} = \left[\theta_n^{(ex)} - z_\alpha\frac{\bar{\sigma}}{\sqrt{n}}, \theta_n^{(ex)} + z_\alpha\frac{\bar{\sigma}}{\sqrt{n}}\right]$

The next objective of this thesis is to compare E-M and Milstein in terms of variance and compute time. In the experiment, the hardware specifications are:

- Hardware processor: 2.8 GHz Intel Core i7.
- Memory: 16 GB 1600 MHz DDR3.

Table 4.3.1: Naive MC of Exchange option by E-M sampling

N	**M**	θ_n	**Variance of** θ_n	95% **CI of** θ_n	99% **CI of** θ_n	**Compute Time**
1000	1000	1.1673	20.1567	$(0.8891, 1.4456)$	$(0.8016, 1.5331)$	4.258944
1000	10000	1.0804	22.0730	$(0.7892, 1.3716)$	$(0.6977, 1.4631)$	40.60171
1000	100000	0.9292	16.9776	$(0.6739, 1.1846)$	$(0.5936, 1.2649)$	392.1904
10000	1000	0.9284	17.2189	$(0.8470, 1.0097)$	$(0.8215, 1.0353)$	42.56745
10000	10000	1.0576	21.3853	$(0.9670, 1.1483)$	$(0.9385, 1.1768)$	425.4602
10000	100000	0.9614	17.5385	$(0.8793, 1.0435)$	$(0.8536, 1.0693)$	4057.075
100000	1000	0.9973	19.0405	$(0.9703, 1.0244)$	$(0.9618, 1.0329)$	425.5349
100000	10000	1.0002	18.8249	$(0.9733, 1.0271)$	$(0.9648, 1.0355)$	4358.332
100000	100000	1.0104	19.4680	$(0.9831, 1.0378)$	$(0.9745, 1.0464)$	41624.19

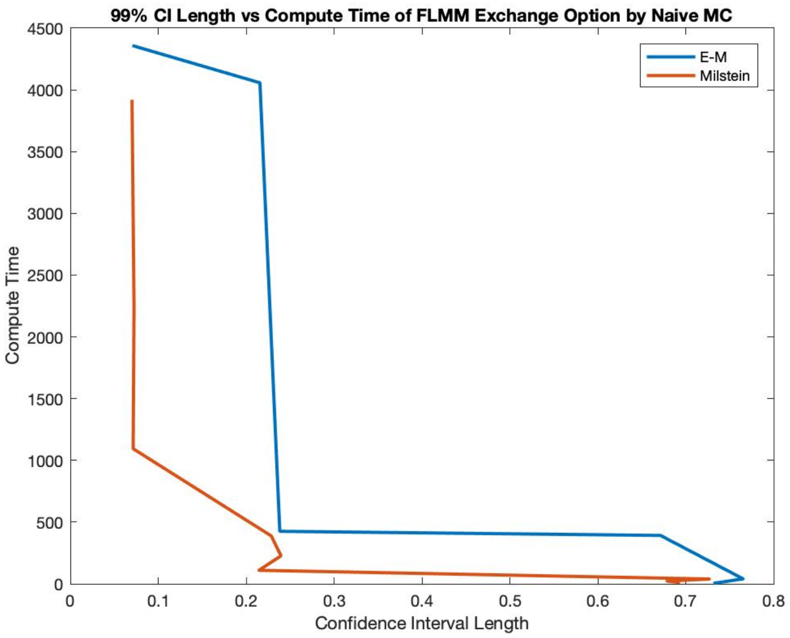
Note: the option parameters used are $S_1(0) = 60$, $S_2(0) = 80$, $T = 0.5$, $t = 0$, $\sigma_1 = 0.4$, $\sigma_2 = 0.2$, $\rho = 0.5$ and $r = 0.05$.

Table 4.3.2: Naive MC of Exchange option by Milstein sampling

N	**M**	θ_n	**Variance of** θ_n	95% **CI of** θ_n	99% **CI of** θ_n	**Compute Time**
1000	100	0.9782	18.1167	$(0.7144, 1.2420)$	$(0.6315, 1.3250)$	11.300327
1000	150	0.9797	17.3380	$(0.7216, 1.2378)$	$(0.6405, 1.3189)$	22.556142
1000	200	1.0290	19.9200	$(0.7523, 1.3056)$	$(0.6654, 1.3926)$	39.1799
10000	100	0.9440	17.3338	$(0.8623, 1.0256)$	$(0.8367, 1.0512)$	109.802010
10000	150	1.0763	21.6443	$(0.9852, 1.1675)$	$(0.9565, 1.1962)$	226.998594
10000	200	0.9707	19.6852	$(0.8837, 1.0576)$	$(0.8564, 1.0850)$	388.166524
100000	100	1.0081	19.2850	$(0.9809, 1.0354)$	$(0.9724, 1.0439)$	1094.246791
100000	150	1.0264	19.8405	$(0.9987, 1.0540)$	$(0.9901, 1.0626)$	2237.045344
100000	200	0.9981	18.5951	$(0.9714, 1.0248)$	$(0.9630, 1.0332)$	3919.410313

Note: the option parameters used are $S_1(0) = 60$, $S_2(0) = 80$, $T = 0.5$, $t = 0$, $\sigma_1 = 0.4$, $\sigma_2 = 0.2$, $\rho = 0.5$ and $r = 0.05$.

From the plot below, we can conclude the Milstein naive sampler is more computationally efficient than the E-M naive sampler, when trying to achieve comparable accuracy.



4.3.2 Antithetic Variate Estimator

There are numerous variance reduction techniques available for Monte Carlo estimators. When the sampling density is symmetric, that is $f(x) = f(-x)$, then the number of samples can be doubled for essentially free. This can be achieved by just incorporating the negative of all the sampled points into the Monte Carlo estimator. Doing this will reduce the sample variance without effecting the sample mean. Suppose $f(x)$ is a symmetrical density and $V(x)$ is a Borel function, the antithetic variate method works because:

$$\begin{aligned}\mathbb{E}[V(X)] &= \int_{-\infty}^{\infty} V(x)f(x)dx = \frac{1}{2}\Big(\int_{-\infty}^{\infty} V(x)f(x)dx + \int_{-\infty}^{\infty} V(-y)f(y)dy\Big)\\ &= \frac{1}{2}\Big(\mathbb{E}\big[V(X)\big] + \mathbb{E}\big[V(-X)\big]\Big) = \mathbb{E}\big[\frac{1}{2}V(X) + \frac{1}{2}V(-X)\big].\end{aligned}$$

On the other hand, the variance is:

$$\begin{aligned}Var\big[\frac{1}{2}V(X) + \frac{1}{2}V(-X)\big] &= \frac{1}{4}\Big(Var\big[V(X)\big] + Var\big[V(-X)\big] + 2Cov\big[V(X), V(-X)\big]\Big)\\ &= \frac{1}{2}Var\big[V(X)\big] + \frac{1}{2}Cov\big[V(X), V(-X)\big].\end{aligned}$$

It is clear that $Var\big[\frac{1}{2}V(X) + \frac{1}{2}V(-X)\big] < Var[V(X)]$ whenever $Cov[V(X), V(-X)]$ is negative.

Returning to the finite liquidity market model of (4.1.1), take $\{S_1^{(i)}, S_2^{(i)}\}_{i=1}^N$ to be a random set of stochastic paths sampled from E-M or Milstein. There exists a set of antithetic paths $\{aS_1^{(i)}, aS_2^{(i)}\}_{i=1}^N$, this is discussed in detail by Giles (2014) [20]. We can setup a matrix recursive version by using (4.2.3) and define the antithetic asset paths:

$$\begin{aligned}\mathbf{aS}(m+1) &= \mathbf{aB}(m)\mathbf{aS}(m), \quad \text{E-M}\\ \mathbf{aS}(m+1) &= \mathbf{aB}(m)\mathbf{aS}(m) + \frac{1}{2}\mathbf{b}(m), \quad \text{Milstein}\end{aligned}$$

where the new term $\mathbf{aB}(m)$ is

$$\mathbf{aB}(m) = \begin{bmatrix} 1 + r\Delta t - \tilde{\sigma}_{11}\big(\mathbf{S}(m)\big)\Delta W_1(m+1) & -\tilde{\sigma}_{12}\big(\mathbf{S}(m)\big)\Delta W_2(m+1) \\ -\tilde{\sigma}_{21}\big(\mathbf{S}(m)\big)\Delta W_1(m+1) & 1 + r\Delta t - \tilde{\sigma}_{21}\big(\mathbf{S}(m)\big)\Delta W_2(m+1) \end{bmatrix}.$$

Pseudo-code for generating antithetic paths from E-M and Milstein are provided in 4.3.2 and 4.3.3.

Algorithm 4.3.2 Antithetic Paths from Euler-Maruyama

Initialize Values $\mathbf{aS}(t) = \mathbf{s}$
Define $\Delta t =: \frac{T-t}{M}$
for $m = 0$ **to** $M-1$ **do**
  $\mathbf{\Delta W}(m) = \big(\Delta w_1(m), \Delta w_2(m)\big) \sim \mathcal{N}_2(0, \Delta t I_2)$
  Set $\mathbf{aB}(m)$
  $\mathbf{aS}(m+1) = \mathbf{aB}(m)\mathbf{aS}(m)$
end for
return $\mathbf{aS}(M)$

Algorithm 4.3.3 Antithetic Paths from Milstein

Initialize Values $\mathbf{aS}(t) = \mathbf{s}$
Define $\Delta t =: \frac{T-t}{M}$
for $m = 0$ **to** $M-1$ **do**
  $\mathbf{\Delta W}(m) = \big(\Delta w_1(m), \Delta w_2(m)\big) \sim \mathcal{N}_2(0, \Delta t I_2)$
  Set $\mathbf{aB}(m)$, $\mathbf{b}(m)$, J_i, Σ and $\mathcal{A}$
  $\mathbf{aS}(m+1) = \mathbf{aB}(m)\mathbf{aS}(m) + \frac{1}{2}\mathbf{b}(m)$
end for
return $\mathbf{aS}(M)$

Apply the generated antithetic paths, the antithetic variate estimator of an option can be defined to be:

$$\theta_{av}^{(ex)} = \frac{e^{-r\tau}}{2N}\sum_{i=1}^{N}\Big(\big(S_1^{(i)}(T) - S_2^{(i)}(T)\big)^+ + \big(aS_1^{(i)}(T) - aS_2^{(i)}(T)\big)^+\Big). \tag{4.3.1}$$

For the exchange option under the finite liquidity framework of (4.1.1), the estimator of (4.3.1) can be set up by the following algorithm:

Algorithm 4.3.4 Antithetic Variate Monte Carlo

Sample N numbers of $\vec{S}_1$,$a\vec{S}_2$, $a\vec{S}_1$ and $a\vec{S}_2$ from E-M or Milstein
$\mathbf{S}_1, \mathbf{S}_2 = [\vec{S}_1^{(1)}, \vec{S}_1^{(2)}, ..., \vec{S}_1^{(N)}; \vec{S}_2^{(1)}, \vec{S}_2^{(2)}, ..., \vec{S}_2^{(N)}]$
$\mathbf{aS}_1, \mathbf{aS}_2 = [a\vec{S}_1^{(1)}, a\vec{S}_1^{(2)}, ..., a\vec{S}_1^{(N)}; \vec{S}_2^{(1)}, a\vec{S}_2^{(2)}, ..., a\vec{S}_2^{(N)}]$
Estimator of Mean:
$\bar{\mu}_{av}^{(ex)} = \frac{1}{N}\sum_{i=1}^{N}\big(aS_1^{(i)}(T) - aS_2^{(i)}(T)\big)^+$
Value of Option:
$\theta_{av}^{(ex)} = e^{-r\tau}(\bar{\mu}_n^{(ex)} + \bar{\mu}_{av}^{(ex)})$
Variance of θ_{av}
$\bar{\sigma}^2 = \frac{e^{-2r\tau}}{4N}\sum_{i=1}^{N}\Big\{\Big(\big(S_1^{(i)}(T) - S_2^{(i)}(T)\big)^+ - \bar{\mu}_n^{(ex)}\Big)^2 + \Big(e^{-r\tau}\big(aS_1^{(i)}(T) - aS_2^{(i)}(T)\big)^+ - \bar{\mu}_{av}^{(ex)}\Big)^2 +$
$2\Big(\big(S_1^{(i)}(T) - S_2^{(i)}(T)\big)^+ - \bar{\mu}_n^{(ex)}\Big)\Big(\big(aS_1^{(i)}(T) - aS_2^{(i)}(T)\big)^+ - \bar{\mu}_{av}^{(ex)}\Big)\Big\}$
$1-\alpha$ Confidence Interval of the Value
$CI^{(ex)} = \big[\theta_{av}^{(ex)} - z_\alpha\frac{\bar{\sigma}}{\sqrt{n}}, \theta_{av}^{(ex)} + z_\alpha\frac{\bar{\sigma}}{\sqrt{n}}\big]$

Using the same option parameters and hardware specifications of section 4.3.1, the experimental results of the antithetic variate estimator is:

Table 4.3.3: Antithetic variate MC of Exchange option by E-M sampling

N	**M**	θ_{av}	**Variance of** θ_{av}	95% **CI of** θ_{av}	99% **CI of** θ_{av}	**Compute Time**
1000	1000	0.8663	6.9150	$(0.7033, 1.0293)$	$(0.6521, 1.0805)$	7.2621531
1000	10000	1.1069	10.9207	$(0.9020, 1.3117)$	$(0.8377, 1.3761)$	69.378477
1000	100000	1.0270	8.5708	$(0.8455, 1.2084)$	$(0.7885, 1.2654)$	725.903939
10000	1000	1.0482	9.8333	$(0.9867, 1.1096)$	$(0.9674, 1.1289)$	75.420984
10000	10000	0.9635	8.6733	$(0.9058, 1.0213)$	$(0.8877, 1.0394)$	682.378098
10000	100000	0.9997	8.6710	$(0.9420, 1.0574)$	$(0.9238, 1.0755)$	7338.541184
100000	1000	0.9955	8.8140	$(0.9771, 1.0139)$	$(0.9713, 1.0197)$	798.884626
100000	10000	1.0017	8.9771	$(0.9831, 1.0203)$	$(0.9773, 1.0261)$	7671.303108
100000	100000	1.0009	9.1100	$(0.9822, 1.0196)$	$(0.9763, 1.0255)$	80655.893226

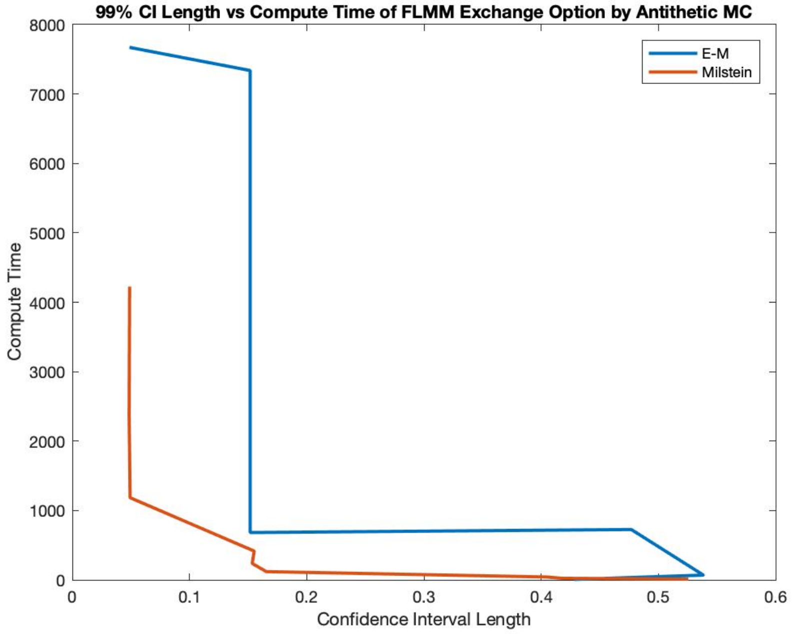
Note: the option parameters used are $S_1(0) = 60$, $S_2(0) = 80$, $T = 0.5$, $t = 0$, $\sigma_1 = 0.4$, $\sigma_2 = 0.2$, $\rho = 0.5$ and $r = 0.05$.

Table 4.3.4: Antithetic variate MC of Exchange option by Milstein sampling

N	**M**	θ_{av}	**Variance of** θ_{av}	95% **CI of** θ_{av}	99% **CI of** θ_{av}	**Compute Time**
1000	100	0.9752	10.4032	$(0.7753, 1.1752)$	$(0.7125, 1.2380)$	11.781618
1000	150	0.8503	6.5731	$(0.6914, 1.0092)$	$(0.6414, 1.0591)$	23.614247
1000	200	1.1215	10.6341	$(0.9193, 1.3236)$	$(0.9193, 1.3236)$	42.144800
10000	100	1.0700	10.3100	$(1.0070, 1.1329)$	$(0.9872, 1.1527)$	119.277441
10000	150	0.9968	8.8748	$(0.9384, 1.0551)$	$(0.9200, 1.0735)$	236.847002
10000	200	1.0032	9.0669	$(0.9442, 1.0622)$	$(0.9256, 1.0808)$	414.999724
100000	100	1.0088	9.1435	$(0.9901, 1.0276)$	$(0.9842, 1.0335)$	1184.310578
100000	150	0.9919	8.8645	$(0.9735, 1.0104)$	$(0.9677, 1.0162)$	2377.319245
100000	200	0.9983	9.0031	$(0.9797, 1.0169)$	$(0.9739, 1.0228)$	4224.681379

Note: the option parameters used are $S_1(0) = 60$, $S_2(0) = 80$, $T = 0.5$, $t = 0$, $\sigma_1 = 0.4$, $\sigma_2 = 0.2$, $\rho = 0.5$ and $r = 0.05$.

The plot below reflects comparable conclusions to the naive samplers.



4.3.3 Control Variate Estimator

For any random variable Y and its random sample $\{Y_i\}_{i=1,2,..}^{N}$, an unbiased estimator of zero can be artificially created as:

$$\theta_0 = \frac{1}{N}\sum_{j=1}^{N}\big[Y_i - \mathbb{E}[Y]\big].$$

The control variate estimator can be created by adding the estimator of zero to the naive Monte Carlo estimator, that is:

$$\theta_{cv} = \theta_n + c\theta_0 = \frac{1}{N}\sum_{j=1}^{N}X_i + \frac{c}{N}\sum_{j=1}^{N}\big(Y_i - \mathbb{E}[Y]\big) = \frac{1}{N}\sum_{j=1}^{N}\big(X_i + cY_i\big) - c\mathbb{E}[Y].$$

Control variate estimator have the same mean as the naive estimator, while having a variance of:

$$Var[\theta_c] = \frac{1}{N^2}\sum_{j=1}^{N}Var\big[X + cY\big] = \frac{1}{N}\sum_{j=1}^{N}\big(Var[X] + c^2Var[Y] + 2cCov[X,Y]\big).$$

The variance is minimized when $\hat{c} = -\frac{Cov[X,Y]}{Var[Y]}$, and the expression will become:

$$Var[\theta_c] = \frac{1}{N}Var(X)\big(1 - Cor^2[X,Y]\big).$$

An important observation to make is that the choice of Y will have a direct impact on the amount of variance reduction. In fact, choice of Y with higher correlation in absolute value with X will cause greater variance reduction effect.

To price exchange option under the finite liquidity framework of (4.1.1), the Margrabe's option would make an excellent candidate as the control variate. This is because the regular GBM paths can be generated simultaneously from the same normal random variables used to numerically solve the SDEs of (4.1.1). One would assume the simulated GBM will be highly correlated with the numerical solution of the SDEs. This thesis will provide algorithms to generate GBMs through E-M or Milstein.

Algorithm 4.3.5 Geometric Brownian Motion from Euler-Maruyama

Set initial values
$CS_1(t) = s_1,\ CS_2(t) = s_2$
Import the normal random variables used for $S_1(t), S_2(t)$
$\mathbf{\Delta W} = [\Delta w_1(1), \Delta w_1(2), ..., \Delta w_1(M), \Delta w_2(1), \Delta w_2(2), ..., \Delta w_2(M),]$
Generate Markov Chain
for $i = 0$ to $M - 1$ **do**
  $CS_1(i+1) = CS_1(i) + rCS_1(i)\Delta t + \sigma_1 CS_1(i)\Delta w_1(i)$
  $CS_2(i+1) = CS_2(i) + rCS_2(i)\Delta t + \rho\sigma_2 CS_2(i)\Delta w_1(i) + \sqrt{1-\rho^2}\sigma_2 CS_2(i)\Delta w_2(i)$
end for
return $\vec{CS}_1,\ \vec{CS}_2$

Algorithm 4.3.6 Geometric Brownian Motion from Milstein

Set initial values
$CS_1(t) = s_1,\ CS_2(t) = s_2$
Import the normal random variables used for $S_1(t), S_2(t)$
$\mathbf{\Delta W} = [\Delta w_1(1), \Delta w_1(2), ..., \Delta w_1(M), \Delta w_2(1), \Delta w_2(2), ..., \Delta w_2(M),]$
Generate Markov Chain
for $i = 0$ to $M - 1$ **do**
  $CS_1(i+1) = CS_1(i) + rCS_1(i)\Delta t + \sigma_1 CS_1(i)\Delta w_1(i) + \frac{1}{2}\sigma_1^2 CS_1(i)\big(\Delta w_1^2 - \Delta t\big)$
  $CS_2(i+1) = CS_2(i) + rCS_2(i)\Delta t + \rho\sigma_2 CS_2(i)\Delta w_1(i) + \sqrt{1-\rho^2}\sigma_2 CS_2(i)\Delta w_2(i) + \frac{1}{2}\rho^2\sigma_2^2 CS_2(i)\Delta w_1^2 + \rho\sqrt{1-\rho^2}\sigma_2^2 CS_2(i)\Delta w_1\Delta w_2 + \frac{1}{2}(1-\rho^2)\sigma_2^2 CS_2(i)\Delta w_2^2 - \frac{1}{2}\sigma_2^2 CS_2(i)\Delta t$
end for
return $\vec{CS}_1,\ \vec{CS}_2$

Use the generated GBMs, the control variate estimator of an option can be defined as:

$$\theta_{cv}^{(ex)} = \frac{e^{-r\tau}}{N}\sum_{j=1}^{N}\Big(\big(S_1^{(i)}(T) - S_2^{(i)(T)}\big)^+ + c\big(CS_1^{(i)}(T) - CS_2^{(i)(T)}\big)^+\Big)$$
$$- c\mathbb{E}\Big[\big(S_1(T) - S_2(T)\big)^+|S_1(t) = s_1, S_2(t) = s_2\Big]. \quad (4.3.1)$$

Here $S_1^{(i)}(T)$ and $S_2^{(i)}(T)$ are from the FLMM of (4.1.3). While $CS_1^{(i)}(T)$ and $CS_2^{(i)}(T)$ are the risk-neural GBMs from the BSM model of (1.3.1). The Exchange option's control variate estimator of (4.3.1) can be set up through the following algorithm:

Algorithm 4.3.7 Control Variate Monte Carlo

Sample N numbers of $\vec{S}_1$,$\vec{S}_2$, $\vec{CS}_1$ and $\vec{CS}_2$ from E-M or Milstein
$\mathbf{S}_1, \mathbf{S}_2 = [\vec{S}_1^{(1)}, \vec{S}_1^{(2)}, ..., \vec{S}_1^{(N)}; \vec{S}_2^{(1)}, \vec{S}_2^{(2)}, ..., \vec{S}_2^{(N)}]$
$\mathbf{CS}_1, \mathbf{CS}_2 = [\vec{CS}_1^{(1)}, \vec{CS}_1^{(2)}, ..., \vec{CS}_1^{(N)}; \vec{CS}_2^{(1)}, \vec{CS}_2^{(2)}, ..., \vec{CS}_2^{(N)}]$
Estimator of mean:
$\bar{\mu}_{cv}^{(ex)} = \frac{1}{N}\sum_{i=1}^{N}\big(CS_1^{(i)}(T) - CS_2^{(i)}(T)\big)^+$
Value of exchange option:
$\theta_{cv}^{(ex)} = e^{-r\tau}(\bar{\mu}_n^{(ex)} - c\bar{\mu}_{cv}^{(ex)}) - c\big(S_1(t)N(d_+) - S_2(t)N(d_-)\big)$
Variance of $\theta_{cv}^{(ex)}$
$\bar{\sigma}^2 = \frac{e^{-2r\tau}}{N-1}\sum_{i=1}^{N}\Big\{\Big(\big(S_1^{(i)}(T) - S_2^{(i)}(T)\big)^+ - \bar{\mu}_n^{(ex)}\Big)^2 + c^2\Big(\big(CS_1^{(i)}(T) - CS_2^{(i)}(T)\big)^+ - \bar{\mu}_{cv}^{(ex)}\Big)^2 +$
$2c\Big(\big(S_1^{(i)}(T) - S_2^{(i)}(T)\big)^+ - \bar{\mu}_n^{(ex)}\Big)\Big(\big(CS_1^{(i)}(T) - CS_2^{(i)}(T)\big)^+ - \bar{\mu}_{cv}^{(ex)}\Big)\Big\}$
$1 - \alpha$ confidence interval of option value
$CI^{(ex)} = \big[\theta_{cv}^{(ex)} - z_\alpha\frac{\bar{\sigma}}{\sqrt{n}}, \theta_{cv}^{(ex)} + z_\alpha\frac{\bar{\sigma}}{\sqrt{n}}\big]$

Using the same option parameters and hardware specifications of section 4.3.1, the experimental results of the control variate estimator for are:

Table 4.3.5: Control variate MC of Exchange option by E-M

N	**M**	θ_n	**Variance of** θ_n	95% **CI of** θ_n	99% **CI of** θ_n	**Compute Time**
1000	1000	1.0011	3.7996e-05	(1.0007, 1.0015)	(1.0006, 1.0016)	4.826849
1000	10000	1.0015	5.8401e-05	(1.0010, 1.0019)	(1.0008, 1.0021)	48.550229
1000	100000	1.0012	3.9162e-05	(1.0008, 1.0016)	(1.0007, 1.0017)	497.311908
10000	1000	1.0011	4.5441e-05	(1.0010, 1.0013)	(1.0010, 1.0013)	49.032302
10000	10000	1.0011	4.3201e-05	(1.0010, 1.0013)	(1.0010, 1.0013)	498.811247
10000	100000	1.0011	4.5981e-05	(1.0010, 1.0012)	(1.0009, 1.0013)	4963.06468
100000	1000	1.0011	4.4595e-05	(1.0011, 1.0012)	(1.0011, 1.0012)	523.006558
100000	10000	1.0012	4.6887e-05	(1.0011, 1.0012)	(1.0011, 1.0012)	4937.107023
100000	100000	1.0012	4.5917e-05	(1.0011, 1.0012)	(1.0011, 1.0012)	49194.245715

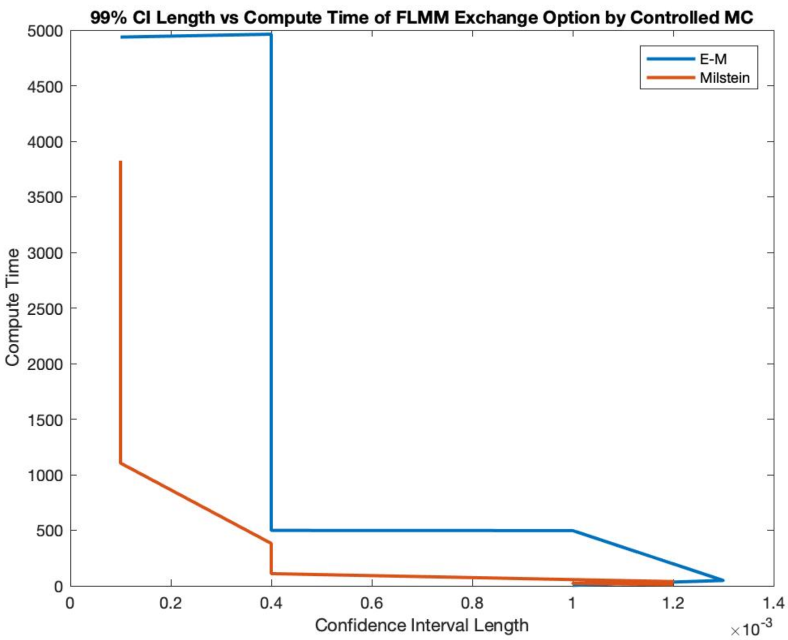
Note: the option parameters used are $S_1(0) = 60$, $S_2(0) = 80$, $T = 0.5$, $t = 0$, $\sigma_1 = 0.4$, $\sigma_2 = 0.2$, $\rho = 0.5$ and $r = 0.05$.

Table 4.3.6: Control variate MC of Exchange option by Milstein

N	**M**	θ_n	**Variance of** θ_n	95% **CI of** θ_n	99% **CI of** θ_n	**Compute Time**
1000	100	1.0019	4.6735e-05	$(1.0015, 1.0023)$	$(1.0013, 1.0025)$	11.142887
1000	150	1.0019	4.3898e-05	$(1.0015, 1.0023)$	$(1.0014, 1.0024)$	22.157359
1000	200	1.0020	5.1035e-05	$(1.0016, 1.0024)$	$(1.0014, 1.0026)$	38.554110
10000	100	1.0019	5.4073e-05	$(1.0018, 1.0021)$	$(1.0017, 1.0021)$	109.983456
10000	150	1.0017	4.6736e-05	$(1.0016, 1.0018)$	$(1.0015, 1.0019)$	222.957975
10000	200	1.0017	5.3887e-05	$(1.0015, 1.0018)$	$(1.0015, 1.0019)$	381.742344
100000	100	1.0018	4.7936e-05	$(1.0017, 1.0018)$	$(1.0017, 1.0018)$	1104.416934
100000	150	1.0018	5.0517e-05	$(1.0017, 1.0018)$	$(1.0017, 1.0018)$	2214.915358
100000	200	1.0018	5.4128e-05	$(1.0017, 1.0018)$	$(1.0017, 1.0018)$	3825.958450

Note: the option parameters used are $S_1(0) = 60$, $S_2(0) = 80$, $T = 0.5$, $t = 0$, $\sigma_1 = 0.4$, $\sigma_2 = 0.2$, $\rho = 0.5$ and $r = 0.05$.

The plot below reflects comparable conclusions to the naive and Antithetic samplers.



Out of all the Monte Carlo samplers we have studied, the controlled variate samplers are the most accurate and computationally efficient. This result is not at all surprising due to the high correlation between FLMM asset and regular GBM asset processes.

4.4 Monte Carlo Methods for Exchange Option Sensitivities

Managing the Greeks is an essential part of trading. To determine the Deltas of FLMM Exchange option, we will adopt the *adjoint method* of Giles and Glasserman (2006) [19]. This method first requires the Greeks to be generated pathwise, then a Monte Carlo can be applied to estimate the actual value. The adjoint method is advantageous because these pathwise Greeks can be generated simultaneously with the assets.

4.4.1 Option Delta

Suppose interchangeability exists between the differential operator and expectation, then the j-th Delta of FLMM Exchange option is:

$$\Delta_j(t) = \frac{\partial}{\partial S_j(t)} \widetilde{\mathbb{E}}^{t,s_1,s_2}\Big[e^{-r(T-t)} V\big(\mathbf{S}(T)\big)\Big] = e^{-r(T-t)} \widetilde{\mathbb{E}}^{t,s_1,s_2}\Big[\frac{\partial}{\partial S_j(t)} V\big(\mathbf{S}(T)\big)\Big].$$

By relaxing certain regularity conditions outlined in Glasserman (2004) [22], we may rewrite it as:

$$\frac{\partial}{\partial S_j(t)} V\big(\mathbf{S}(T)\big) = \sum_{i=1}^{2} \frac{\partial V}{\partial S_i(T)} \frac{\partial S_i(T)}{\partial S_j(t)}.$$

During implementation, $\frac{\partial V}{\partial S_i(T)}$ can be approximated through algorithmic differentiation. While the $\frac{\partial S_i(T)}{\partial S_j(t)}$ term is obtained from taking the path-wise derivative of E-M (4.2.3) or Milstein scheme (4.2.2). Set $\Delta_{ij}(t) = \frac{\partial S_i(T)}{\partial S_j(t)}$, we obtain an approximating scheme for $\Delta_{ij}(m)$ as follows:

$$\Delta_{ij}(m+1) = \Delta_{ij}(m) + r\Delta_{ij}(m)\Delta t + \sum_{k,l=1}^{2} \frac{\partial \widetilde{\sigma}_{ik}}{\partial s_l} \Delta_{lj}(m) \Delta W_k(m+1) \quad \text{E-M}$$

$$\Delta_{ij}(m+1) = \Delta_{ij}(m) + r\Delta_{ij}(m)\Delta t + \sum_{k,l=1}^{2} \frac{\partial \widetilde{\sigma}_{ik}}{\partial s_l} \Delta_{lj}(m) \Delta W_k(m+1) \quad \text{Milstein}$$
$$+ \frac{1}{2} \sum_{k,l,p,q=1}^{2} \Delta_{qj}(m) \Big(\frac{\partial^2 \widetilde{\sigma}_{ik}}{\partial s_p \partial s_q} \widetilde{\sigma}_{pj}\big(\mathbf{S}(m)\big) + \frac{\partial \widetilde{\sigma}_{ik}}{\partial s_p} \frac{\partial \widetilde{\sigma}_{pl}}{\partial s_q} \Big),$$

where $m = 0, 1, ...M - 1$. If we define a matrix $\mathbf{D}(m)$ as:

$$D_{ij}(m) = \delta_{ij}(m) + r\Delta t + \sum_{k=1}^{2} \frac{\partial \widetilde{\sigma}_{ik}}{\partial s_j} \Delta W_k(m+1) \quad \text{E-M}$$

$$\begin{aligned} D_{ij}(m) &= \delta_{ij}(m) + r\Delta t + \sum_{k=1}^{2} \frac{\partial \widetilde{\sigma}_{ik}}{\partial s_j} \Delta W_k(m+1) \quad \text{Milstein} \\ &+ \frac{1}{2} \sum_{k,l,p=1}^{2} \Big(\frac{\partial^2 \widetilde{\sigma}_{ik}}{\partial s_p \partial s_j} \widetilde{\sigma}_{pj}\big(\mathbf{S}(m)\big) + \frac{\partial \widetilde{\sigma}_{ik}}{\partial s_p} \frac{\partial \widetilde{\sigma}_{pl}}{\partial s_j} \Big), \end{aligned}$$

then the evolution of $\mathbf{\Delta}$ can be redefined using matrix recursion as follows:

$$\mathbf{\Delta}(m+1) = D(m)\mathbf{\Delta}(m),$$

where $\mathbf{\Delta}(t) = I$. Similar to estimating the option price, we a can use the Delta from the Magrabe option as a multivariate control variate. We adopt the method presented by Rubinstein and Marcus (1985) [41] and set up the estimator for Delta:

$$\overline{\mathbf{\Delta}} = \frac{e^{-r(T-t)}}{N} \sum_{i=1}^{N} \Big(\mathbf{\Delta}^{(i)}(M) + C_1 \mathbf{\Delta}_{cv}^{(i)}(M) \Big) - C_1 \mathbf{\Delta}_{Margrabe}. \tag{4.4.1}$$

The variance of $\overline{\mathbf{\Delta}}$ is minimized when $\hat{C}_1 = \Sigma_{\mathbf{\Delta}\mathbf{\Delta}_{cv}} \Sigma^{-1}_{\mathbf{\Delta}_{cv}\mathbf{\Delta}_{cv}}$.

Chapter 5

Numerical Methods II - Spread Option

5.1 Existence of a Delta Hedging Price Impact Model

In a market where majority of the participate adopts delta hedging, a reasonable trading strategy function for Market Model (3.1.1) would be the Spread Option's Delta on asset one. That is,

$$f(t, s_1, s_2) = \Delta_1(t) = (2\pi)^{-2} e^{-r\tau} \frac{K}{s_1} \int\int_{\mathbb{R}^2 + i\epsilon} u_1 e^{i\mathbf{u}'\mathbf{X}(t)} \Phi(\mathbf{u}, \tau) \hat{P}(\mathbf{u}) d\mathbf{u}. \tag{5.1.1}$$

To see the SDE $dS_1(t)$ in (4.1.1) have an unique solution in the case of strategy function (5.1.1). Recall back to Equation (2.2.6), it is established that all the Greeks involving partials of s_1 and s_2 can be expressed as linear combination of:

$$\overline{\mathbf{Greek}}(t, s_1, s_2) = \int\int_{\mathbb{R}^2 + i\epsilon} f_{\otimes}(\mathbf{u}) e^{i\mathbf{u}'\mathbf{X}(t)} \Phi(\mathbf{u}, \tau) \hat{P}(\mathbf{u}) d\mathbf{u}.$$

This contour integral is equivalent to:

$$\overline{\mathbf{Greek}}(t, s_1, s_2) = e^{-\epsilon'\mathbf{X}(t)} \int\int_{\mathbb{R}^2} f_{\otimes}(\mathbf{u} + i\epsilon) e^{i\Re(\mathbf{u})'\mathbf{X}(t)} \Phi(\mathbf{u} + i\epsilon, \tau) \hat{P}(\mathbf{u} + i\epsilon) d\mathbf{u}. \tag{5.1.2}$$

Within Expression (5.1.2), recall $\mathbf{X}(t) = log\big(\mathbf{S}(t)\big)$. Since $e^{i\Re(\mathbf{u})'\mathbf{X}(t)}$ is on the complex unit circle, then (5.1.2) is unbounded in $\mathbf{S}(t)$ on $\big([0, T] \times (\mathbb{R}^+)^2\big)$ solely due to the $e^{-\epsilon'\mathbf{X}(t)}$ term. However, the dampening constants has restrictions of $\epsilon_1 < -1$ and $\epsilon_2 > 0$. Because of this reason, we should incorporate a global price cap $\overline{S}$ and a global price floor $\underline{S}$ into a modified price impact function. The purpose of the cap and floor is to ensure the existence of our market model. Consider the Spread Option price impact function:

$$\bar{\lambda}\big(t, s_1, s_2\big) = \begin{cases} \epsilon\big(1 - e^{-\beta(T-t)^{\frac{3}{2}}}\big) & \text{if} \quad \underline{S} < s_1, s_2 < \overline{S}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.1.3}$$

Here ϵ and β still are the price impact per share and decaying constant respectively. For the rest of this chapter, ϵ will be set to 0.04, and β will be set to 100. It is important to emphasize $\bar{\lambda}(t, s_1, s_2)$ will be employed for numerical approximation. The theoretical $\lambda(t, s_1, s_2)$ should be a function with bounded derivative, that is obtained through a standard mollifying of $\bar{\lambda}(t, s_1, s_2)$.

For the remainder of this chapter, the trading strategy and price impact functions will be assumed to be $\Delta_1(t)$ and $\lambda(t, s_1, s_2)$ respectively.

Consider the FLMM with Spread Option Greeks. The dynamics of this model is:

$$\begin{aligned}
\frac{dS_1(t)}{S_1(t)} &= \bar{\mu}_1\big(t, S_1(t), S_2(t)\big)dt + \bar{\sigma}_{11}\big(t, S_1(t), S_2(t)\big)dW_1(t) + \bar{\sigma}_{12}\big(t, S_1(t), S_2(t)\big)dW_2(t),\\
\frac{dS_2(t)}{S_2(t)} &= \mu_2 dt + \bar{\sigma}_{21}dW_1(t) + \bar{\sigma}_{21}dW_2(t),\\
\frac{dD(t)}{D(t)} &= -rdt,
\end{aligned} \tag{5.1.4}$$

where

$$\begin{aligned}
&\bar{\mu}_1(t, s_1, s_2) = \frac{1}{1-\lambda(t, s_1, s_2)\Gamma_{11}(t)}\bigg(\mu_1(t) + \frac{1}{s_1}\lambda(t, s_1, s_2)\Theta(t) + \mu_2(t)\frac{s_2}{s_1}\lambda(t, s_1)\Gamma_{12}(t)\\
&+ \frac{1}{2}Speed_{111}(t)\frac{1}{\big(1-\lambda(t, s_1, s_2)\Gamma_{11}(t)\big)^2}\bigg(\sigma_1^2 s_1 + \sigma_2^2\frac{s_2^2}{s_1}\lambda^2(t, s_1)\big(\Gamma_{12}(t)\big)^2\\
&+ 2\rho\sigma_1\sigma_2 s_2\lambda(t, s_1, s_2)\Gamma_{12}(t)\bigg) + Speed_{112}(t)\frac{1}{1-\lambda(t, s_1, s_2)\Gamma_{11}(t)}\big(\rho\sigma_1\sigma_2 s_2\\
&+ \sigma_2^2\frac{s_2^2}{s_1}\lambda(t, s_1, s_2)\Gamma_{12}(t)\big) + \frac{1}{2}\sigma_2^2\frac{s_2^2}{s_1}Speed_{122}(t)\bigg),\\
&\bar{\sigma}_{11}(t, s_1, s_2) = \frac{\sigma_1}{1-\lambda(t, s_1, s_2)\Gamma_{11}(t)}, \qquad \bar{\sigma}_{12}(t, s_1, s_2) = \frac{\sigma_2 s_2\lambda(t, s_1, s_2)\Gamma_{12}(t)}{s_1\Big(1-\lambda(t, s_1, s_2)\Gamma_{11}(t)\Big)},\\
&\bar{\sigma}_{21} = \sigma_2\rho, \qquad \bar{\sigma}_{22} = \sigma_2\sqrt{1-\rho^2}.
\end{aligned}$$

To avoid the complex $\bar{\mu}_1\big(t, S_1(t), S_2(t)\big)$ term, we choose to work under the risk neutral measure with the model dynamics:

$$\begin{aligned}
\frac{dS_1(t)}{S_1(t)} &= rdt + \bar{\sigma}_{11}\big(t, S_1(t), S_2(t)\big)d\widetilde{W}_1(t) + \bar{\sigma}_{12}\big(t, S_1(t), S_2(t)\big)d\widetilde{W}_2(t),\\
\frac{dS_2(t)}{S_2(t)} &= rdt + \bar{\sigma}_{21}d\widetilde{W}_1(t) + \bar{\sigma}_{21}d\widetilde{W}_2(t),\\
\frac{dD(t)}{D(t)} &= -rdt.
\end{aligned} \tag{5.1.5}$$

Theorem 5.1 (**Existence and Uniqueness of Finite Liquidity Market Model SDE III**).

The SDE system (5.1.5) has a strong solutions.

The proofs of this theorem is available at Appendix B.3.

5.2 Fast Fourier Transform

A fast Fourier transform (FFT) is a discrete Fourier transform algorithm with a complexity of $\mathcal{O}(2N\log(N))$ instead of $\mathcal{O}(2N^2)$ when computing N points. FFT was first adopted by Cooley and Tukey (1965) [14].

In this chapter, the goal is to simulate a solution for the SDEs of the finite liquidity market model. Since $\Gamma_{11}(t)$ and $\Gamma_{12}(t)$ are the only Greeks within $dS_1(t)$, therefore only these Greeks needs to be simulated. To construct the FFT scheme for spread option Gammas, one must first discretize the Fourier frequency domain with the lattice:

$$\mathcal{L} = \left\{\mathbf{u}(\mathbf{k}) = \big(u(k_1), u(k_2)\big)\big|\mathbf{k} = (k_1, k_2) \in (0, 1, ..., N-1)^2\right\},$$

where η is the frequency domain lattice spacing. Then for $\bar{u} = \frac{N\eta}{2}$, the frequency domain bounds are $[-\bar{u}, \bar{u}]$, and the frequency lattice points are

$$u(k) = -\bar{u} + k\eta = \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{bmatrix} + k \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}.$$

The reciprocal domain lattice can be defined as:

$$\mathcal{L}^* = \left\{\mathbf{x}(\mathbf{l}) = \big(u(l_1), u(l_2)\big)\big|\mathbf{l} = (l_1, l_2) \in (0, 1, ..., N-1)^2\right\},$$

where the lattice points are

$$x(l) = -\bar{x} + l\eta^* = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} + l \begin{bmatrix} \eta_1^* \\ \eta_2^* \end{bmatrix},$$

with the corresponding bounds

$$\bar{\mathbf{x}} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} = \begin{bmatrix} \frac{N\eta_1^*}{2} \\ \frac{N\eta_2^*}{2} \end{bmatrix}.$$

The frequency domain bounds $\bar{u}_i$ can be initialized in a manner to ensure the initial asset value $\mathbf{X}(t) = \big(X_1(t), X_2(t)\big)$ lay exactly on the grid. The following algorithm outlines the method

Algorithm 5.2.1 FFT Spread option grid spacing

```
for i = 0 to N − 1 do
  ū_i^(new) = π(i − N/2) / X_i(t)
  if ū_i^(new) > ū then
    return ū_i^(new)
  end if
end for
return ū_i
```

Then it is clear that the frequency and reciprocal domain spacing are:

$$\eta_i = \frac{2\bar{u}_i^{(new)}}{N},$$

and

$$\eta_i^* = \frac{2\pi}{N\eta_i}.$$

Using the frequency domain lattice, (2.2.3) can be discretized in the following manner:

$$\boldsymbol{\Gamma}(t) \sim -\frac{Ke^{-r\tau}\eta_1\eta_2}{(2\pi)^2}\Big\{\mathcal{T}^{(3)}\sum_{k_1,k_2=0}^{N-1} i\big(\mathbf{u}(\mathbf{k})+i\epsilon\big)e^{i\left(\mathbf{u}(\mathbf{k})+i\epsilon\right)'\mathbf{X}(t)}\Phi\big(\mathbf{u}(\mathbf{k})+i\epsilon,\tau\big)\hat{P}\big(\mathbf{u}(\mathbf{k})+i\epsilon\big)d\mathbf{u}$$
$$+\mathcal{H}\Big(\sum_{k_1,k_2=0}^{N-1}\Big(\big(\mathbf{u}(\mathbf{k})+i\epsilon\big)\bar{\otimes}\big(\mathbf{u}(\mathbf{k})+i\epsilon\big)\Big)e^{i\left(\mathbf{u}(\mathbf{k})+i\epsilon\right)'\mathbf{X}(t)}\Phi\big(\mathbf{u}(\mathbf{k})+i\epsilon,\tau\big)\hat{P}\big(\mathbf{u}(\mathbf{k})+i\epsilon\big)d\mathbf{u}\Big)\mathcal{H}\Big\}.$$

Then, by applying the reciprocal domain lattice, FFT can be applied to determine a solution grid that contains $\mathbf{X}(t)$.

$$\begin{aligned}\boldsymbol{\Gamma}(t) &\sim -(-1)^{l_1+l_2}\frac{\eta_1\eta_2KN^2e^{-r\tau}}{(2\pi)^2}e^{-\epsilon'\mathbf{X}(\mathbf{l})}\Big\{\frac{1}{N^2}\sum_{k_1,k_2=0}^{N-1}e^{\frac{2\pi i\mathbf{k}\mathbf{l}}{N}}M(\mathbf{k})\Big\}\\ &= -(-1)^{l_1+l_2}\frac{\eta_1\eta_2KN^2e^{-r\tau}}{(2\pi)^2}e^{-\epsilon'\mathbf{X}(\mathbf{l})}\Big\{IFFT\big(M(\mathbf{K})\big)\Big\},\end{aligned}$$

where

$$M(\mathbf{k}) = \begin{bmatrix}\frac{1}{S_1^2(t)}\big(H_{\Delta_1}(\mathbf{k})-H_{\Gamma_{11}}(\mathbf{k})\big) & \frac{1}{S_1(t)S_2(t)}H_{\Gamma_{12}}(\mathbf{k})\\ \frac{1}{S_1(t)S_2(t)}H_{\Gamma_{12}}(\mathbf{k}) & \frac{1}{S_2^2(t)}\big(H_{\Delta_2}(\mathbf{k})-H_{\Gamma_{22}}(\mathbf{k})\big)\end{bmatrix},$$

and

$$\begin{aligned}H_{\Delta_1}(\mathbf{k}) &= (-1)^{k_1+k_2}\big(iu_1(k_1)-\epsilon_1\big)\Phi\big(\mathbf{u}(\mathbf{k}+i\epsilon),\tau\big)\hat{P}\big(\mathbf{u}(\mathbf{k}+i\epsilon)+i\epsilon\big),\\ H_{\Delta_2}(\mathbf{k}) &= (-1)^{k_1+k_2}\big(iu_2(k_2)-\epsilon_2\big)\Phi\big(\mathbf{u}(\mathbf{k}+i\epsilon),\tau\big)\hat{P}\big(\mathbf{u}(\mathbf{k}+i\epsilon)+i\epsilon\big),\\ H_{\Gamma_{11}}(\mathbf{k}) &= (-1)^{k_1+k_2}\big(iu_1(k_1)-\epsilon_1\big)^2\Phi\big(\mathbf{u}(\mathbf{k}+i\epsilon),\tau\big)\hat{P}\big(\mathbf{u}(\mathbf{k}+i\epsilon)+i\epsilon\big),\\ H_{\Gamma_{12}}(\mathbf{k}) &= (-1)^{k_1+k_2}\big(iu_1(k_1)-\epsilon_1\big)\big(iu_2(k_2)-\epsilon_2\big)\Phi\big(\mathbf{u}(\mathbf{k}+i\epsilon),\tau\big)\hat{P}\big(\mathbf{u}(\mathbf{k}+i\epsilon)+i\epsilon\big),\\ H_{\Gamma_{22}}(\mathbf{k}) &= (-1)^{k_1+k_2}\big(iu_2(k_2)-\epsilon_2\big)^2\Phi\big(\mathbf{u}(\mathbf{k}+i\epsilon),\tau\big)\hat{P}\big(\mathbf{u}(\mathbf{k}+i\epsilon)+i\epsilon\big).\end{aligned}$$

The spread option Gammas can be generated through the following FFT algorithm:

Algorithm 5.2.2 FFT of Spread Option Gammas

Set up the $M(\mathbf{k})$ matrix
Run inverse fast Fourier by $IFFT\big(M(\mathbf{k})\big)$
Set up a $C(\mathbf{x})$ matrix as $C_{ij} = (-1)^{i+j} e^{-r\tau} \frac{\eta_1 \eta_2 N^2}{(2\pi)^2} e^{-\epsilon \mathbf{x}(t)}$
Compute the Gammas value grid by $Grid_\Gamma(i,j) = Ke^{-r\tau} \Re\Big(C(\mathbf{x}) IFFT\big(M(\mathbf{k})\big)\Big)$
Find the position on $\mathcal{L}^*$ corresponding to $\mathbf{X}(t)$ as (p_1, p_2)
return $Grid_\Gamma(p_1, p_2)$

The full code for Spread Option price and Greeks are available at:
https://github.com/ShuAiii/Spread-option-by-fast-Fourier-transform.
The Gammas generated from Algorithm 5.2.2 can be used along with E-M Algorithm 4.2.1 or Milstein Algorithm 4.2.3 to generate the asset processes of FLMM.

5.3 Monte Carlo for Spread Option

For the reminder of this chapter, we will use a naive and a controlled variate estimator to price Spread Option under FLMM. Our goal is to optimize for variance and compute time. The objective function is:

$$V(t, s_1, s_2) = \widetilde{\mathbb{E}}[e^{-r(T-t)}\big(S_1(T) - S_2(T) - K\big)^+ | S_1(T) = s_1, S_2(T) = s_2],$$

This is the result of the discounted Feynman-Kac formula of Section 3.5, with the payoff function $h(s_1, s_2) = (s_1 - s_2 - K)^+$.

5.3.1 Naive Estimator

The previous section covered a method of generating spread option Gammas through FFT. With the Gammas in hand, the FLMM assets can be simulated through E-M or Milstein algorithm in a similar fashion to Algorithm 4.2.1 and 4.2.3. Ultimately, a naive Monte Carlo estimator for spread option's market value can be defined as:

$$\theta_n^{(spr)} = \frac{e^{-r\tau}}{N} \sum_{i=1}^{N} \big(S_1^{(i)}(T) - S_2^{(i)}(T) - K\big)^+. \tag{5.3.1}$$

The following algorithm computes the spread option's market value, variance and confidence interval:

Algorithm 5.3.1 Naive Monte Carlo for spread option

Sample N paths of $\vec{S}_1$ and $\vec{S}_2$ from E-M or Milstein with the FFT Scheme Spread option Gammas

$\mathbf{S}_1, \mathbf{S}_2 = [\vec{S}_1^{(1)}, \vec{S}_1^{(2)}, ..., \vec{S}_1^{(N)}; \vec{S}_2^{(1)}, \vec{S}_2^{(2)}, ..., \vec{S}_2^{(N)}]$

Estimator of Mean:

$\bar{\mu}_n^{(spr)} = \frac{1}{N}\sum_{i=1}^{N}\left(\left(S_1^{(i)}(T) - S_2^{(i)}(T) - K\right)^+\right)$

Option Value:

$\theta_n^{(spr)} = e^{-r\tau}\bar{\mu}_n^{(spr)}$

Variance of $\theta_n^{(spr)}$

$\bar{\sigma}^2 = \frac{e^{-2r\tau}}{N}\sum_{i=1}^{N}\left(\left(S_1^{(i)}(T) - S_2^{(i)}(T) - K\right)^+ - \bar{\mu}_n^{(spr)}\right)^2$

$1-\alpha$ Confidence Interval of the Value

$CI^{(spr)} = \left[\theta_n^{(spr)} - z_\alpha\frac{\bar{\sigma}}{\sqrt{n}}, \theta_n^{(spr)} + z_\alpha\frac{\bar{\sigma}}{\sqrt{n}}\right]$

The next objective of this thesis is to compare E-M and Milstein in terms of variance and compute time.

Table 5.3.1: Naive MC of spread option by E-M sampling

N	**M**	θ_n	**Variance of** θ_n	95% **CI of** θ_n	99% **CI of** θ_n	**Compute Time**
1000	1000	0.5912	11.0583	(0.3851, 0.7974)	(0.3204, 0.8621)	895.112738
1000	2000	0.7469	12.1064	(0.5313, 0.9626)	(0.4635, 1.0303)	1810.373330
1000	5000	0.7359	11.2246	(0.5282, 0.9435)	(0.4629, 1.0088)	4351.69952
2000	1000	0.5133	7.1721	(0.3978, 0.6325)	(0.3609, 0.6694)	1779.017111
2000	2000	0.6276	11.1551	(0.4812, 0.7740)	(0.4352, 0.8200)	3351.880970
2000	5000	0.6752	12.3751	(0.5191, 0.8275)	(0.4706, 0.8759)	8660.636052
5000	1000	0.7050	15.2943	(0.5966, 0.8134)	(0.5626, 0.8475)	4243.803608
5000	2000	0.6844	11.4267	(0.5907, 0.7781)	(0.5612, 0.8075)	8540.922872
5000	5000	0.6838	15.0798	(0.5762, 0.7915)	(0.5423, 0.8253)	21584.663345

Note: the option parameters used are $S_1(0) = 60$, $S_2(0) = 80$, $K = 4$, $T = 0.5$, $t = 0$, $\sigma_1 = 0.4$, $\sigma_2 = 0.2$, $\rho = 0.5$ and $r = 0.05$.

The full code for FLMM Spread Option pricing is available at:
https://github.com/ShuAiii/FLMMSpread.

5.3.2 Control Variate Estimator

Once again, the Margrabe's price formula for Exchange Option can be used as the control variate to price spread option under the FLMM. The control variate estimator of spread option can be defined as:

$$\theta_{cv}^{(spr)} = \frac{e^{-r\tau}}{N}\sum_{j=1}^{N}\Big(\big(S_1^{(i)}(T) - S_2^{(i)}(T) - K\big)^+ + c\big(CS_1^{(i)}(T) - CS_2^{(i)}(T)\big)^+\Big)$$
$$- c\mathbb{E}\Big[\big(S_1(T) - S_2(T)\big)^+ | S_1(t) = s_1, S_2(t) = s_2\Big]. \tag{5.3.1}$$

Similar to the previous chapter, $S_1^{(i)}(T)$ and $S_2^{(i)}(T)$ are from the FLMM of . While $cS_1^{(i)}(T)$ and $cS_2^{(i)}(T)$ are the risk-neural GBMs from the BSM model of (1.3.1). The GBMs of both market models are generated through Algorithm 4.2.1 or 4.2.3.

The control variate estimator for the FLMM spread option value can be computed through the following algorithm:

Algorithm 5.3.2 Control variate Monte Carlo for spread option

Sample N paths of $\vec{S}_1$ and $\vec{S}_2$ from E-M or Milstein with the FFT Scheme Spread option Gammas

$\mathbf{S}_1, \mathbf{S}_2 = [\vec{S}_1^{(1)}, \vec{S}_1^{(2)}, ..., \vec{S}_1^{(N)}; \vec{S}_2^{(1)}, \vec{S}_2^{(2)}, ..., \vec{S}_2^{(N)}]$

$\mathbf{CS}_1, \mathbf{CS}_2 = [\vec{CS}_1^{(1)}, \vec{CS}_1^{(2)}, ..., \vec{CS}_1^{(N)}; \vec{CS}_2^{(1)}, \vec{CS}_2^{(2)}, ..., \vec{CS}_2^{(N)}]$

Estimator of mean:

$\bar{\mu}_{cv} = \frac{1}{N}\sum_{i=1}^{N}\big(CS_1^{(i)}(T) - CS_2^{(i)}(T)\big)^+$

Value of exchange option

$\theta_{cv}^{(spr)} = e^{-r\tau}(\bar{\mu}_n^{(spr)} - c\bar{\mu}_{cv}) - c\big(S_1(t)N(d_+) - S_2(t)N(d_-)\big)$

Variance of $\theta_{cv}^{(spr)}$:

$\bar{\sigma}^2 = \frac{e^{-2r\tau}}{N-1}\sum_{i=1}^{N}\Big\{\Big(\big(S_1^{(i)}(T) - S_2^{(i)}(T) - K\big)^+ - \bar{\mu}_n^{(spr)}\Big)^2 + c^2\Big(\big(CS_1^{(i)}(T) - CS_2^{(i)}(T)\big)^+ - \bar{\mu}_{cv}\Big)^2 + 2c\Big(\big(S_1^{(i)}(T) - S_2^{(i)}(T) - K\big)^+ - \bar{\mu}_n^{(spr)}\Big)\Big(\big(CS_1^{(i)}(T) - CS_2^{(i)}(T)\big)^+ - \bar{\mu}_{cv}\Big)\Big\}$

$1 - \alpha$ confidence interval of option value:

$CI^{(spr)} = \big[\theta_{cv}^{(spr)} - z_\alpha\frac{\bar{\sigma}}{\sqrt{n}}, \theta_{cv}^{(spr)} + z_\alpha\frac{\bar{\sigma}}{\sqrt{n}}\big]$

Using the same hardware specifications of Section 4.3.1, the experimental results of the control variate estimator for are:

Table 5.3.2: Control variate MC of spread option by E-M

N	**M**	θ_n	**Variance of** θ_n	95% **CI of** θ_n	99% **CI of** θ_n	**Compute Time**
1000	1000	0.6654	0.4585	$(0.6234, 0.7074)$	$(0.6102, 0.7206)$	928.132601
1000	2000	0.7080	0.3891	$(0.6694, 0.7467)$	$(0.6572, 0.7589)$	1660.219954
1000	5000	0.6872	0.4403	$(0.6461, 0.7284)$	$(0.6332, 0.7413)$	4409.007364
2000	1000	0.6949	0.4349	$(0.6660, 0.7238)$	$(0.6569, 0.7329)$	1630.974633
2000	2000	0.6908	0.3729	$(0.6640, 0.7176)$	$(0.6556, 0.7260)$	3248.058178
2000	5000	0.6654	0.4075	$(0.6374, 0.6934)$	$(0.6286, 0.7022)$	8307.109659
5000	1000	0.6781	0.4997	$(0.6585, 0.6977)$	$(0.6585, 0.6977)$	4165.095556
5000	2000	0.6792	0.4045	$(0.6615, 0.6968)$	$(0.6560, 0.7023)$	8326.471157
5000	5000	0.6790	0.4150	$(0.6611, 0.6968)$	$(0.6555, 0.7024)$	21638.734037

Note: the option parameters used are $S_1(0) = 60$, $S_2(0) = 80$, $K = 4$, $T = 0.5$, $t = 0$, $\sigma_1 = 0.4$, $\sigma_2 = 0.2$, $\rho = 0.5$ and $r = 0.05$.

Of course, the controlled variate estimator yields a more attractive result. However, the controlled variate is not as effective compared to the effect on FLMM Exchange Option. This result is not surprising because the Margrabe Option is more correlated with FLMM Exchange Option.

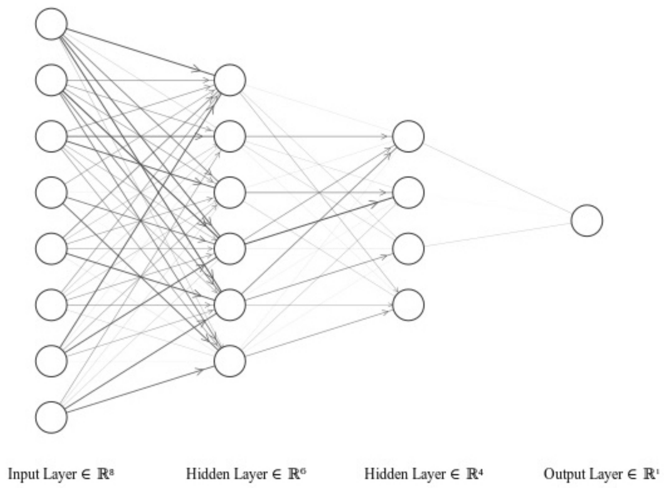
Chapter 6

Deep Learning Methods

6.1 Feed Forward Network

Artificial neural networks have powerful predictive capabilities, one of the first version is the *feed forward network*(FFN). This network is structured as a sequence of layers, with various number of neurons embedded in each layer.

Figure 6.1.1: A Feed-Forward Network



We shall use N to denote the number of layers, and n_i to denote the number of neurons in the i-th layer. In a fully connected FFN, each neuron in the current layer has a connection with each neuron in the subsequent layer. The strength of these connection are known as *weights*, we denote the weights connected to the j-th neuron in the i-th layer as $\mathbf{w}_j^{[i]}$. Each neuron also carries a unique bias term $b_j^{[i]}$, this term has a similar effect as the regression intercept. The final component of a neuron is the activation function $\sigma(z)$, similar to linking functions of non-linear regression, its purpose is to add non-linearity. The table below contains some of the widely used activation functions:

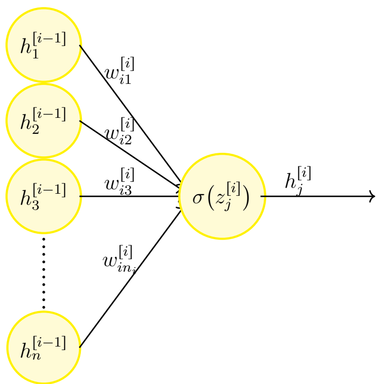
Table 6.1.1: Neuron Activation Functions and Derivatives

Type	Activation Function	Derivative Function
Linear	$\sigma(x) = x$	$\sigma'(x) = 1$
Sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}}$	$\sigma'(x) = \sigma(x)\big(1 - \sigma(x)\big)$
ReLU	$\sigma(x) = \max(x, 0)$	$\sigma'(x) = \mathbb{1}_{(x>0)}$
Gaussian	$\sigma(x) = e^{-x^2}$	$\sigma'(x) = 2xe^{-x^2}$

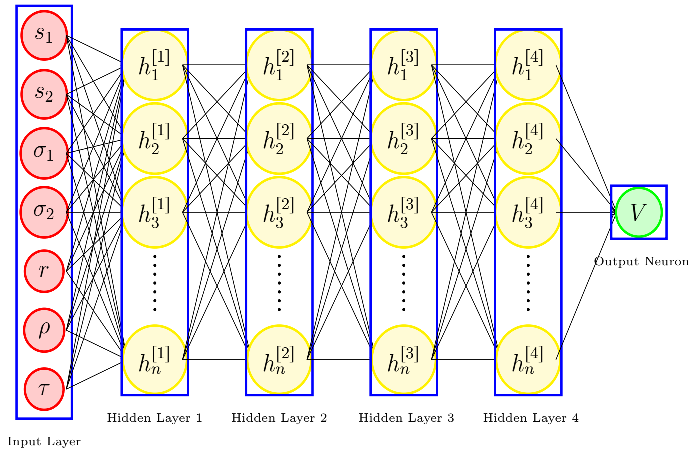
The operation of a neuron can be expressed as:

$$z_j^{[i]} = \mathbf{w}_j^{[i]}\mathbf{h}^{[i-1]} + b_j^{[i]},$$
$$h_j^{[i-1]} = f(z_j^{[i]}).$$

We also provide a computation graph on the j-th neuron in the i-th layer:



This process is repeated for every single neuron, which allows us transverse through the network and arrive at the output layer $h^{[N]} = \hat{y}$(For the purpose of option pricing, we have a single output $h^{[N]}$, but in general $h^{[N]}$ is a vector). This entire process is often refereed to as *forward propagation*. The figure below describes the FFN architecture deployed to price Exchange option under FLMM:



The *loss function* measures the goodness of fit. We will use it to evaluate the result of the forward propagation. This evaluation is preformed for every B inputs; B is known as the *batch size*. Some commonly used loss functions are:

Table 6.1.2: Neural Network Loss Functions

Type	Loss Function
Mean Squared Error	$\mathcal{L}(\vec{y}, \vec{\hat{y}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
Mean Absolute Error	$\mathcal{L}(\vec{y}, \vec{\hat{y}}) = \frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$
Cross Entropy	$\mathcal{L}(\vec{y}, \vec{\hat{y}}) = -\frac{1}{n}\sum_{i=1}^{n}\big(y_n \log(\hat{y}_n) - (1-y_n)\log(1-\hat{y}_n)\big)$

In this thesis, we use the mean squared error(MSE) loss function. The minimizing of the loss function follows the steepest descent idea, so one has to compute gradient fields with respect to the weights and biases. This is often accomplished through algorithmic differentiation and a process called *back propagation*. Then, the weights and biases are updated in the direction of gradient field, hoping of the discovery of a “good enough” local minimum. The common methodology is the *batch gradient descent*:

$$\mathbf{w}_j^{[i],(new)} = \mathbf{w}_j^{[i],(old)} - \alpha\frac{\partial \mathcal{L}}{\partial \mathbf{w}_j^{[i],(old)}},$$

$$\mathbf{b}_j^{[i],(new)} = \mathbf{b}_j^{[i],(old)} - \alpha\frac{\partial \mathcal{L}}{\partial \mathbf{b}_j^{[i],(old)}},$$

$$\text{for} \quad j = 1, 2, ...n_i, \text{ and } i = 1, 2, ...N.$$

In the above expression, α is the *learning rate*.

One batch of forward propagation combined with one instance of back propagation is considered as one iteration of batch training. An *epoch* encompasses a series of batch training that exhausts the entire data set. Normally, the training is either repeated for a fixed number of epochs, or stopped early when the loss function cease to to decrease further.

The most famous theorem in neural networks is the *universal approximation theorem*, it highlights the approximation power of feed forward networks. Hornik (1989) [26] establish the fact deep forward network are universal approximators, in other words, any function can be accurately approximated by some deep feed forward network. Since option prices are smooth solutions of PDEs, then it should be feasible to predict these solutions with feed forward networks.

6.2 Predicting Exchange Option Price with Feed Forward Network

Options pricing are very computational expensive tasks. This becomes especially true under practical settings due to complexities of the underlying yield curve, volatility surface and the payoff function itself. Some methods typically used for option pricing are Monte Carlo and finite difference, these methods slows down drastically as sample size and grid space gets finer. Feed forward networks can be used to predict option prices much faster and up to a certain degree of accuracy. Ferguson and Green (2018) [17] have employed this method to price basket options. However, as they mentioned, there are initial costs from generating the option price data set and training the network. Nevertheless, this strategy could be reasonably implemented by large financial institutions and in theory, should integrate well with their current operations. This is largely due to the following reasons:

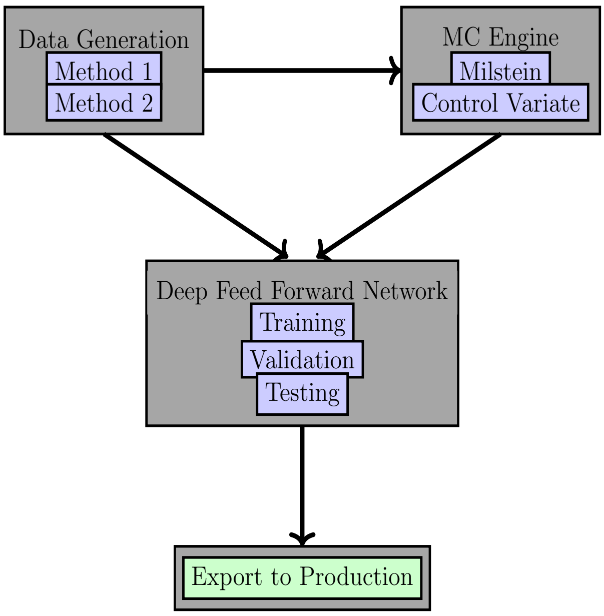
- The data set generation and training can be done offline while the markets are closed.
- Trading operations in the front office requires the option pricing models to be accurate. However, the performance speed is just as important.

To demonstrate option pricing with FFN, we take the MC approach of pricing FLMM Exchange Option from subsection 4.3.1. In this case, the input space has 7 dimensions, that is $\mathbf{x} = (s_1, s_2, r, \rho, \sigma_1, \sigma_2, \tau)$. By selecting an appropriate range for each dimension, we can generate the input data from which the output data can be calculated from. In this case, the output data are the option price computed from Algorithm 4.3.1. It is important to emphasize the distribution used to generate each parameter in the input space is completely problem dependent. Some factors could be considered when choosing the distributions are:

- The physical meaning of each undying parameter.
- The payoff function itself should be considered because it is pointless to generate a bunch of out of money MC path.
- The Margrabe Greeks should also be considered given its high correlation to our model. The Greeks will provide valuable insights on option price sensitivity due to each underlying parameter. Presence of certain large Greeks in some option parameter sub-spaces indicates a higher sensitivity of the pricing function to the parameters in that sub-space. Without a doubt, a higher concentration of sample points should be chosen from that sub-space.

Doing this will not only help the error to converge faster, but also help the FFN to approximate a more meaningful solution. The implementation of Deeply Learning Derivative method can be summarized by the following programming architectural graph:

Figure 6.2.1: Architecture for Exchange option price



To demonstrate this method, we will first consider the Exchange Option under BSM and FLMM. We used two different schemes to generate the training/testing data set. Scheme 1 is a simple sampling method, while scheme 2 is a sophisticated method that attempts to capture more realistic scenarios.

Table 6.2.1: Data Generation Scheme 1

Parameter	Distribution
s_1	$\mathcal{U}(0, 100)$
s_2	$\mathcal{U}(0, 100)$
σ_1	$\mathcal{U}(0, 0.5)$
σ_2	$\mathcal{U}(0, 0.5)$
r	$\mathcal{U}(0, 0.1)$
ρ	$\mathcal{U}(1, -1)$
τ	$\mathcal{U}(0, 2)$

Table 6.2.2: Data Generation Scheme 2

Parameter	Distribution
s_1	$50\exp(X_1),\ X_1 \sim \mathcal{N}(0.5, 0.25)$
s_2	$50\exp(X_1 - X_2),\ X_2 \sim \mathcal{N}(0.5, 0.25)$
σ_1	$\mathcal{U}(0, 0.5)$
σ_2	$\mathcal{U}(0, 0.5)$
r	$\mathcal{U}(0, 0.1)$
ρ	$2(X_3 - 0.5),\ X_3 \sim \beta(5, 2)$
τ	$\mathcal{U}(0, 2)$

We set up our network in Tensorflow [2]. The FFN has 4 fully connected deep layers with 300 ReLu neurons per layer. The output layer contains a single SoftPlus Neuron to make sure the prediction would be positive definite. We generated 1 million inputs and uses a relatively inaccurate MC engine (N=100, M=100) to construct the training set. The reason behind this choice is, in practice, a well-trained deep FFN has the ability to remove the inaccuracy of weak MC estimators. We trained the FNN with mini-batch size of 1024, and updated the gradient with ADAM optimizer (2015) [17]. We performed validation with samples created from a highly accurate MC engine (N=100k, M=100), at a 100/1 ratio. Initially, the FFN was set to train for 1000 epochs. After 850 epochs of training, the loss function ceases to decreases further significantly. To prevent overfitting, it is justifiable to apply early stopping and make validation conclusions based on MAE and MSE.

Figure 6.2.2: Exchange option MAE

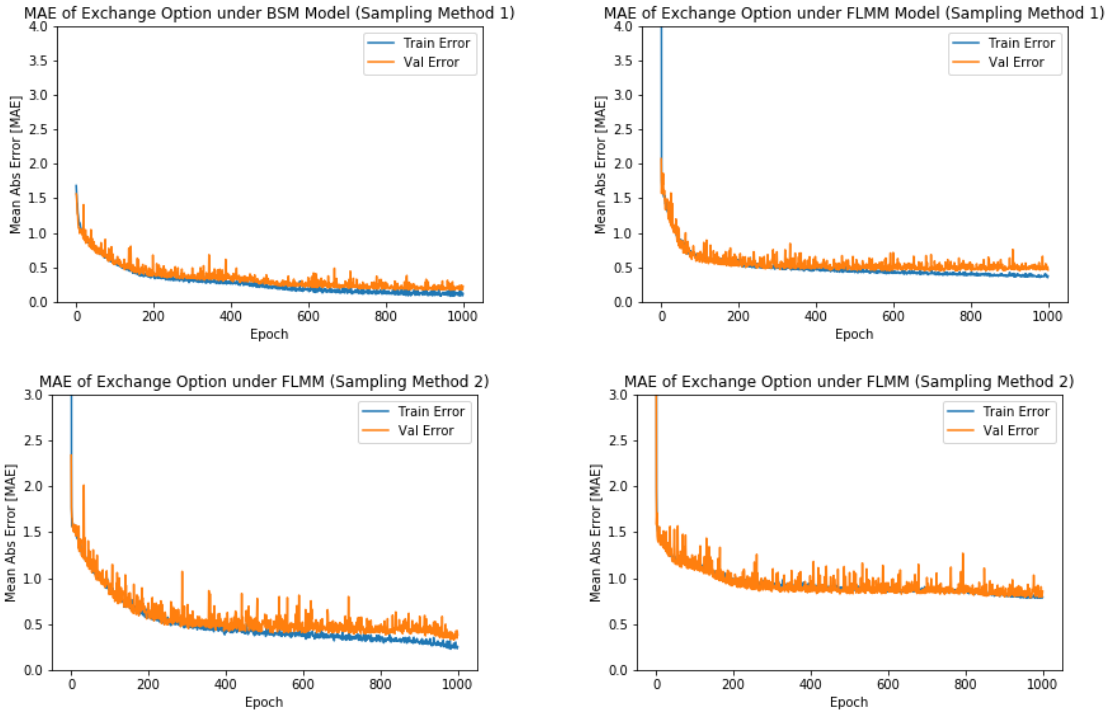
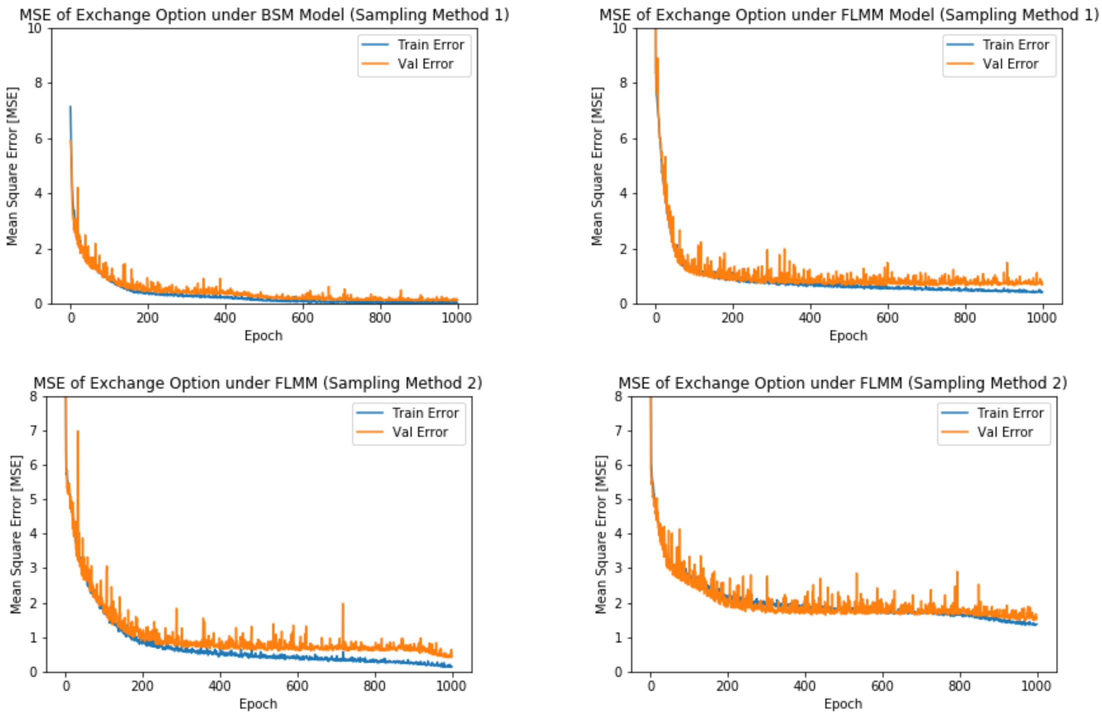


Figure 6.2.3: Exchange option MSE



We observe both of the MAE and MSE of the testing data oscillate around the MAE and MSE of the training data for all 4 cases. Furthermore, the amplitude of the oscillation decreases as the Epoch index increases. This shows all predictive models built from the training data are good predictive model for each respective testing data. Another important result to notice is that the models were more attractive in terms of MSE. This would imply the bigger errors matched up more consistently between training and testing datasets.

In the testing phase, we generated 1000 highly accurate samples with MC engine specifications (N=100k, M=100). We test our trained network and came to the following testing results:

Figure 6.2.4: Exchange option predicted value vs true value

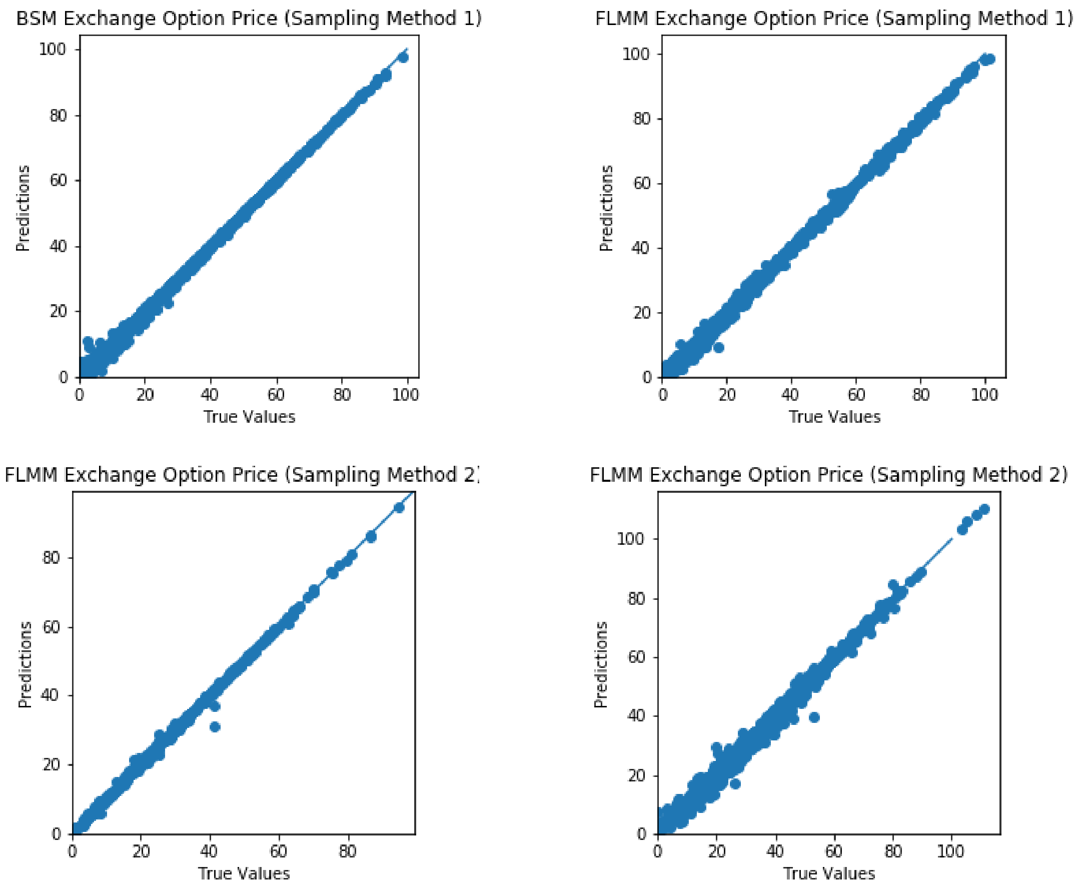
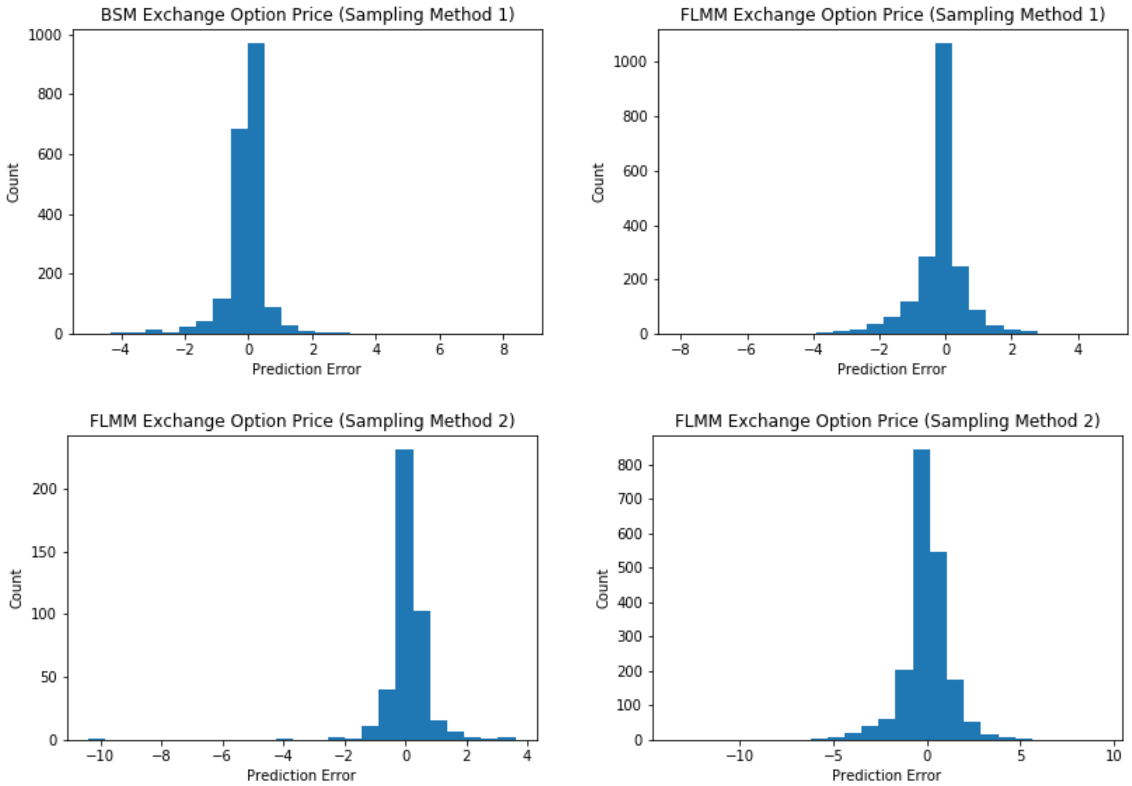


Figure 6.2.5: Exchange option prediction error frequency

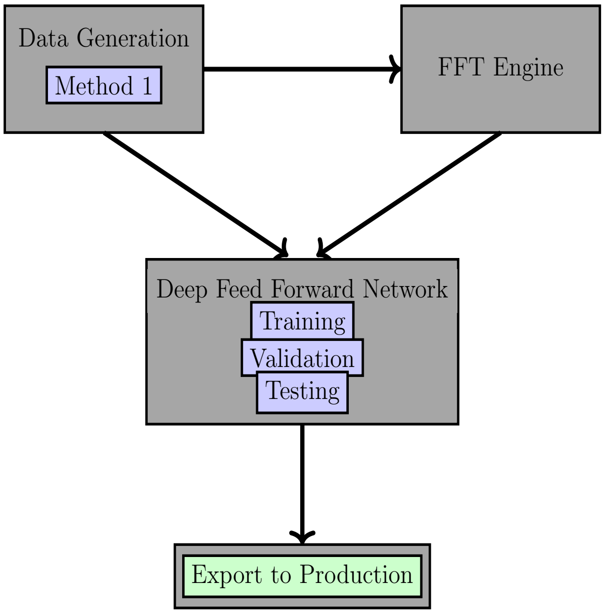


We performed our final analysis on the testing data. It can be clearly observed there is a strong linear relationship between each set of true values and predicted value. Furthermore, by examining frequencies of the prediction errors, we observe normality in each case.

6.3 Predicting Spread Option Greeks

Similarly, feed-forward nets can be used to determine Spread Option Greeks. This can be achieved by replacing the Monte Carlo step in 6.1.1 with the respective FFT Scheme from the theoretical Greeks of 2.2.2.

Figure 6.3.1: Architecture for spread option sensitivities



We generated a sample batch of 100,000 points by uniform sampling. The samples were divided into a 90/10/10 manner for training/validation/testing. The nets were set up to have 4 fully connected hidden layers with 300 ReLu neurons in each layer. The output layer has a single linear neuron. We used the ADAM optimizer with a learning rate $\alpha = 0.0001$. After 2000 epochs of training, we validated our model by comparing training MAE/MSE against validation.

We observe both of the MAE and MSE of the validation data is higher than that of the training data for all Greeks. Furthermore, there is some evidence of over-fitting as the training error deviates from validation error. The deviation is most clear in the case of Theta. To avoid over-fitting, one could incorporate early-stopping or regularization. The error seems to converge well for Deltas, while only converging to a satisfactory level for Theta and Gammas. This shows the predictive models for Delta will be accurate, while the rest might produce unsatisfactory predictions.

Figure 6.3.2: Spread Greeks MAE

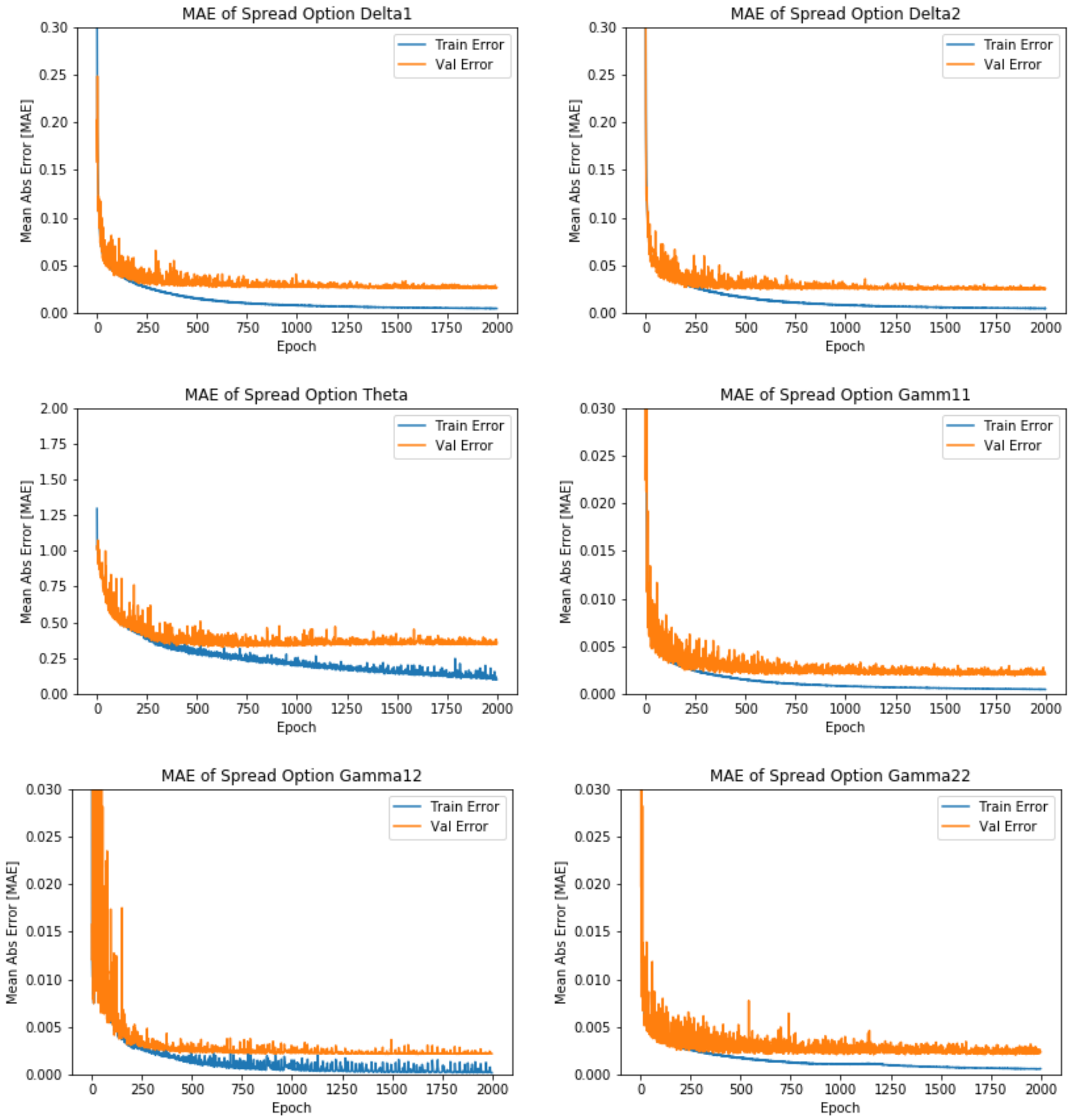
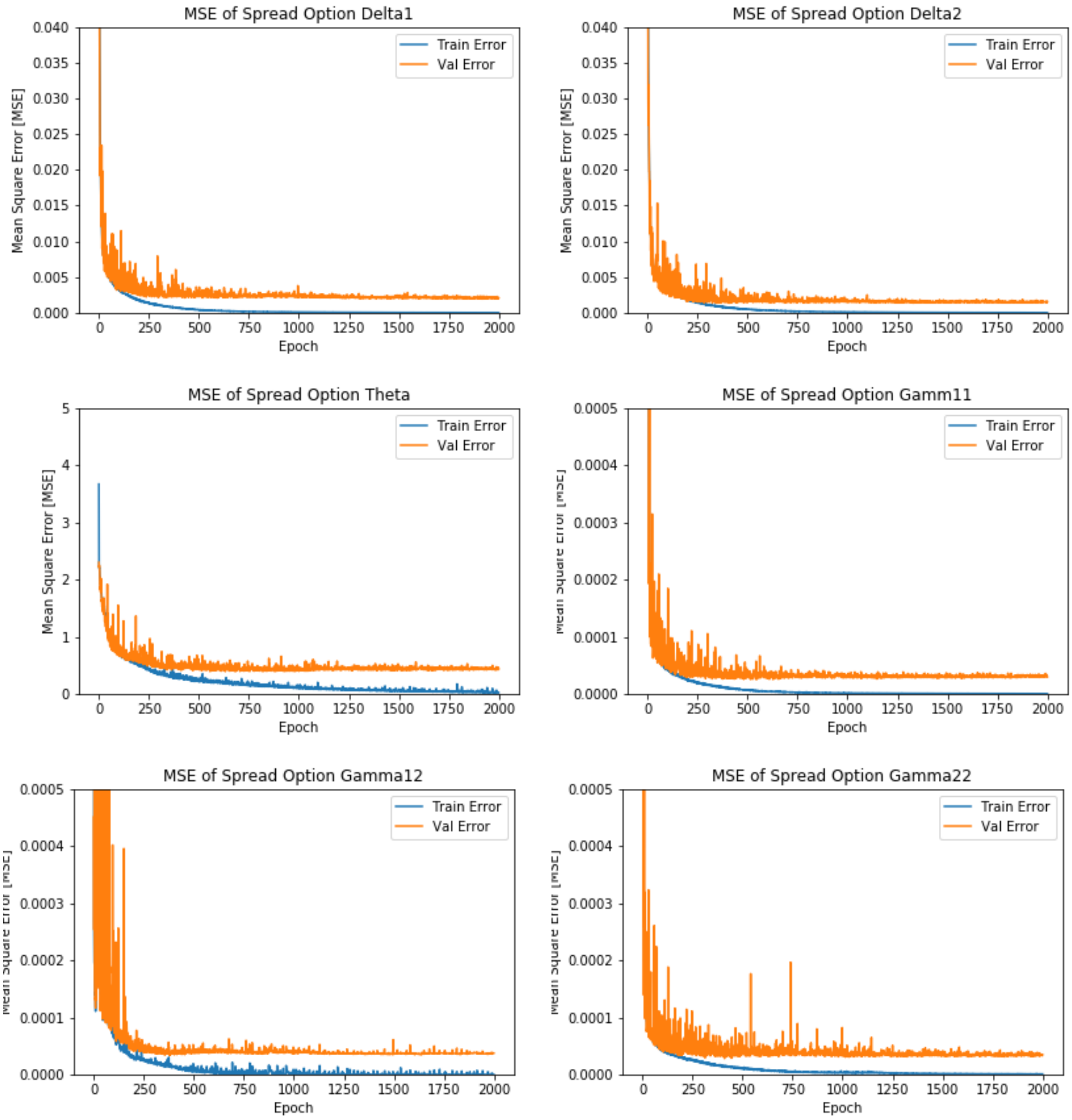
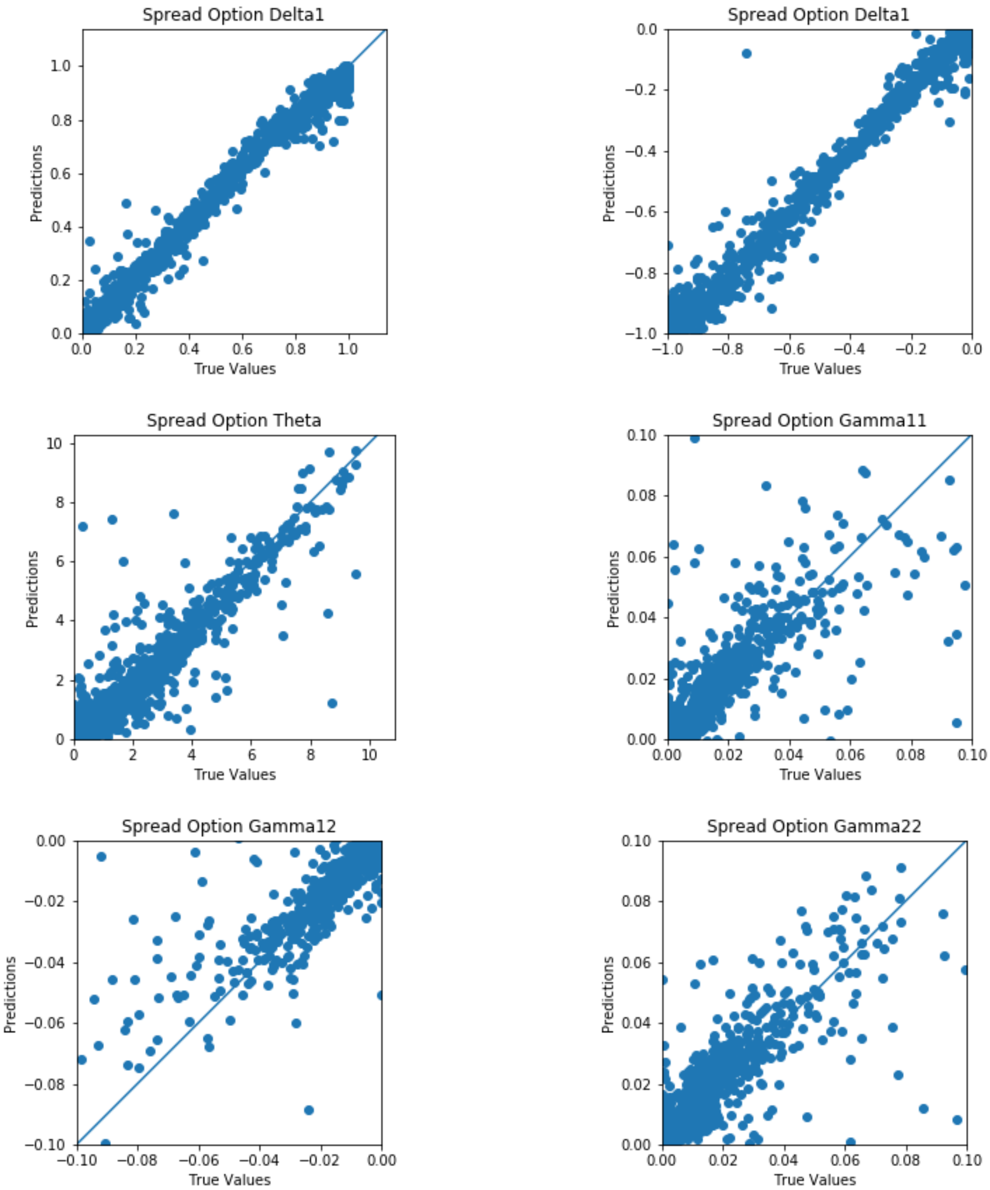


Figure 6.3.3: Spread Greeks MSE



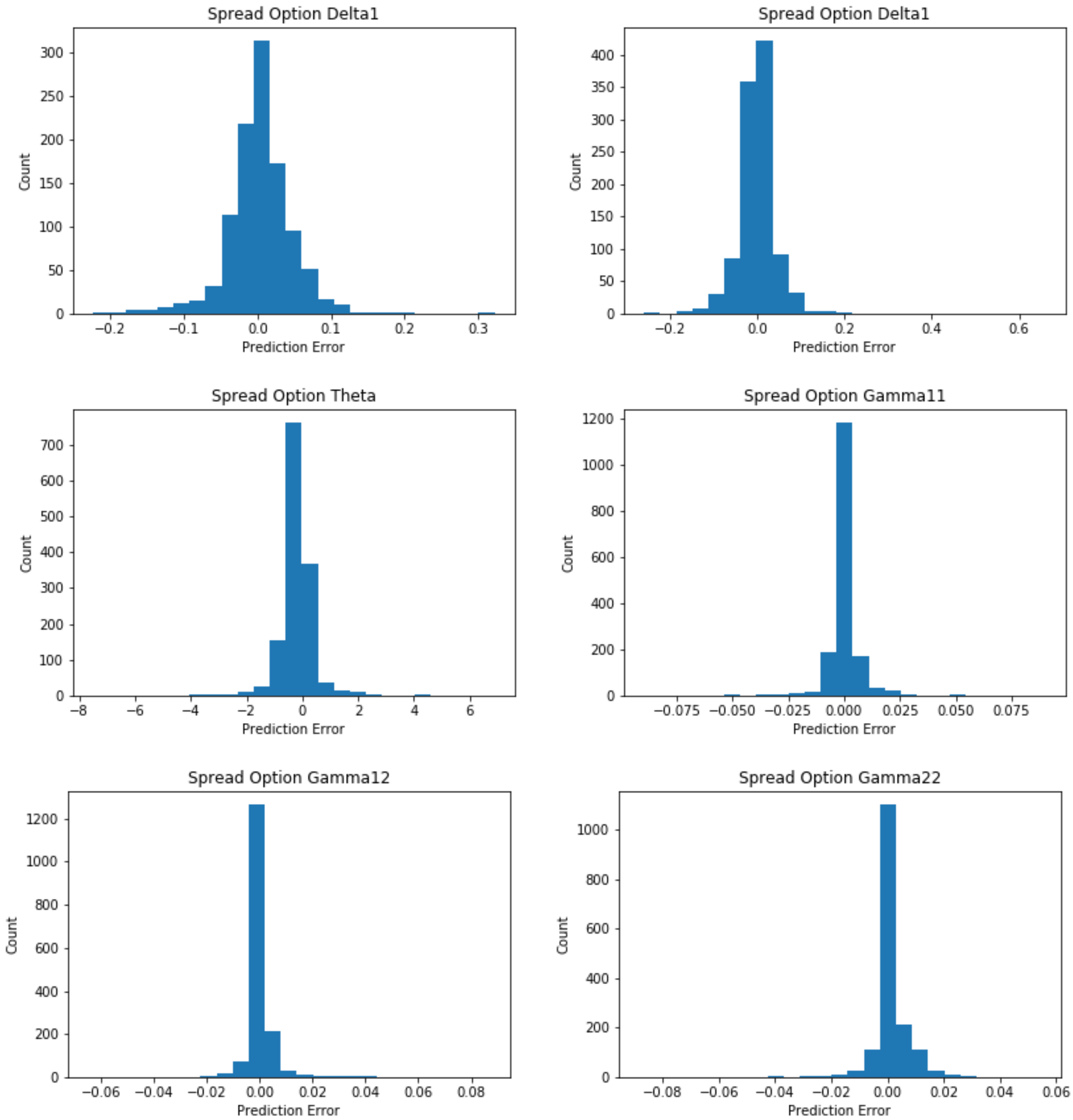
By comparing the true Greek values against the predicted values, we can conclude the networks for Deltas will produce accurate predictions. The Theta network will produce somewhat satisfactory predictions, and the Gamma networks will be inaccurate.

Figure 6.3.4: Spread Greeks true value vs predicted value



The Delta error frequency plots resemble normal distributions very well. The Theta error frequency plot somewhat resembles a normal distribution. While the Gamma error frequency plots resemble Student's t distribution. A t-distributed error frequency plot could be caused by a lack of sample size. In other words, we might be able to fix this problem by generating more option data.

Figure 6.3.5: Spread Greeks prediction error frequency

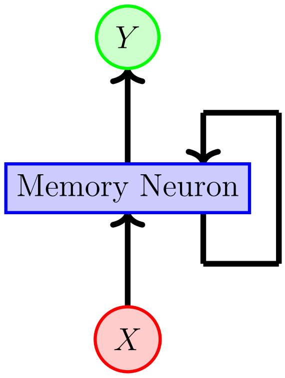


This procure of combining FFN and FFT provides an interesting new way to Spread Option obtain Spread Option sensitivities. From our experiments, we conclude the first order Greeks can be accurately trained with $100k$ sample points. The second-order Greeks cannot be sufficiently trained with $100k$ points.

6.4 Recurrent Neural Network

Recurrent neural networks (RNN) are artificial neural networks with embedded looping structures. This allows RNN to store information in its memory. A *rolled* up architecture is referred to the RNN architectural form with the loops present, as shown in Figure 6.4.1.

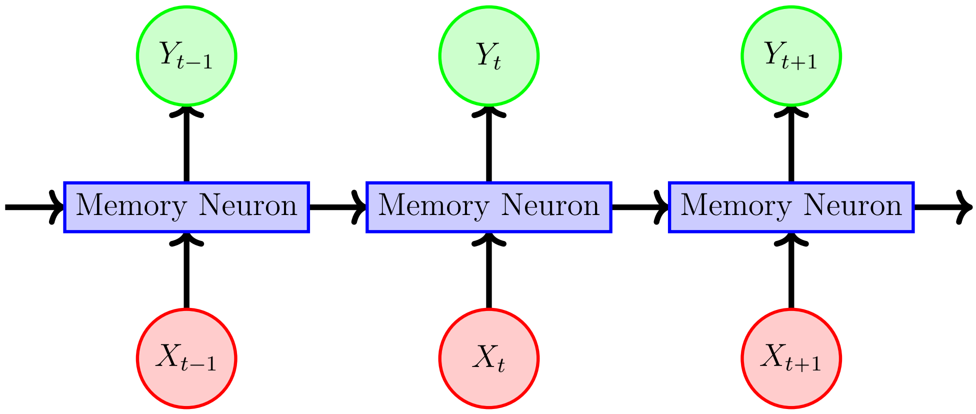
Figure 6.4.1: Rolled up Recurrent Neural Network



Observed in the above diagram, the looping structures allow the information present in the current state of the memory neuron be relevant to the next state.

As compact as the rolled-up architectural form of RNN is, sometimes it could be confusing to the reader. The *unrolled* form of RNN is essentially an expansion of the looping structure, as shown in Figure 6.4.2.

Figure 6.4.2: Unrolled Recurrent Neural Network



The unrolled structure of RNN has a close resemblance to linked lists. This is the reason behind the promising results of certain sub-family of RNN in the field of translation and speech recognition [42, 46].

6.5 Deep Galerkin Method

The curse of dimensionality is a common issue when attempting to solve high dimensional PDEs (include some literature). In high dimensions, methods such as finite-difference not only become costly but are often unstable. Muti-asset option pricing PDEs are affected by the curse of dimensionality. The *Deep Galerkin Method* (DGM), developed by Sirignano and Spiliopoulos (2018) [45], have the potential to address these issues. DGM takes advantage of the approximation power of the neural network and solves PDEs numerically. Consider a two-asset option pricing PDE:

$$\begin{cases} \mathcal{L}V(t, s_1, s_2) = 0, & (t, s_1, s_2) \in [0, T] \times (\mathbb{R}^+)^2, \quad (interior) \\ V(T, s_1, s_2) = h(s_1, s_2), & (s_1, s_2) \in (\mathbb{R}^+)^2, \quad (terminal) \\ V(t, s_1, s_2) = g(t, s_1, s_2), & (t, s_1, s_2) \in [0, T] \times \mathcal{B} \quad (boundary). \end{cases} \tag{6.5.1}$$

For the Sobolev space $\mathcal{H}_0^1 = \mathcal{H}_0^1\big([0, T] \times (\mathbb{R}^+)^2\big)$, the equivalent weak formulation of (6.5.1) is:

$$\begin{cases} \langle \mathcal{L}V, u \rangle = 0 & \forall u \in \mathcal{H}_0^1, \\ \langle V - h, v \rangle = 0 & \forall v \in \bar{\mathcal{H}}_0^1, \\ \langle V - g, w \rangle = 0 & \forall w \in \partial\mathcal{H}_0^1. \end{cases} \tag{6.5.2}$$

The regular Galerkin Method would require the careful selection of a set of basis functions $(\phi_1, \phi_2, ..., \phi_N)$, that characterizes a finite approximation space $E_N \subset \mathcal{H}_0^1$. A unique best approximation, $\hat{V}_N$ of V, can be determined by projecting the PDE onto E_N. By increasing the dimension of approximation space, the projection theorem gives a unique best approximation $\hat{V}_i$ in each approximation space E_i. The result is a sequence of approximators $\{\hat{V}_i\}_{i=N,N+1,..}$, that converges to V by the completeness of $\mathcal{H}_0^1$. The rigorous formulation of this method can be found in [28].

DGM deviates from the regular Galerkin Method by assuming a neural network $f(t, s_1, s_2; \theta) : \mathbb{R}^3 \to \mathbb{R}$ has the potential to capture the behavior of V. The network is subsequently initialized and trained with information gathered from the domain of V. This requires the selection of a meaningful objective function. The suggested objective function in [45] closely resembles the weak formulation of (6.5.2) with the $\mathcal{L}^2$ inner product. The choice of $\mathcal{L}^2$ norm can be supported by literature such as [8]. The construction of the objective function proceeds as follows. For unit vectors $u \in \mathcal{H}_0^1, v \in \bar{\mathcal{H}}_0^1$, and $\partial\mathcal{H}_0^1$, we apply the Cauchy-Schwartz Inequality to the weak formulation equations in (6.5.2). Next, we sum the resulting terms, thus producing the objective function defined by

$$\mathcal{J} = \|\mathcal{L}V\|^2_{[0,T]\times(\mathbb{R}^+)^2} + \|V - h\|^2_{(\mathbb{R}^+)^2} + \|V - g\|^2_{[0,T]\times(\mathbb{R}^+)^2}. \tag{6.5.3}$$

During the implementation stage, distributions are selected to generate points in the domain. This gives rise to a $\mathcal{L}^2$ distributional norm $\|f(\mathbf{x})\|^2_{\mathcal{D},\phi} = \int_{\mathcal{D}} |f(\mathbf{x})|^2 \phi(\mathbf{x}) d\mathbf{x}$, where $\phi(\mathbf{x})$ is a probability density function on the domain. Thus, the choice of $\phi(\mathbf{x})$ will significantly impact the performance of this method. A suitable objective function can be reformulated for our option pricing problems as:

$$J(\theta) = J_1(\theta) + J_2(\theta) + J_3(\theta), \tag{6.5.4}$$

where $J_i(\theta)$, for $i = 1, 2, 3$, are defined as follows:

$$J_1(\theta) = \|\mathcal{L}f(t, s_1, s_2; \theta)\|^2_{[0,T]\times(\mathbb{R}^+)^2,\phi_1}. \tag{6.5.5}$$

This objective function measures how well the network satisfies the pricing PDE's differential operator.

$$J_2(\theta) = \|f(T, s_1, s_2; \theta) - h(s_1, s_2)\|^2_{(\mathbb{R}^+)^2,\phi_2}. \tag{6.5.6}$$

This objective function measures how closely the network resembles the payoff function at maturity.

$$J_3(\theta) = \|f(t, s_1, s_2; \theta) - g(t, s_1, s_2)\|^2_{[0,T]\times\mathcal{B},\phi_3}. \tag{6.5.7}$$

The last objective function characterizes boundary conditions, or artificially created boundaries from asymptotics. For European style option pricing PDEs, these boundaries exist when underlying prices reach 0. The asymptotic appears when a price cap is impose on the underlings.

The training data are generated as a tuple $(x^{(i)}, x^{(T)}, x^{(b)})$, where $x^{(i)} = (t, s_1, s_2) \sim \phi_1$, $x^{(T)} = (T, s_1, s_2) \sim \phi_2$ and $x^{(b)} = (t, s_1^{(b)}, s_2^{(b)}) \sim \phi_3$. In particular, $x^{(i)}$, $x^{(T)}$ and $x^{(b)}$ are generated from the interior, terminal and boundary (or artificial boundary) of the PDE respectively. The generated data are used to compute the objective function (6.5.4).

In the next phase, we apply a gradient descent algorithm, in hope of eventually finding a set of parameters θ for $f(t, s_1, s_2; \theta)$ that will produce a minima for the objective function. In fact, Correia et al. (2019) [4] mentioned DGM is strictly an optimization problem. Validation is unnecessary because the objective function directly characterizes the weak formulation of the PDE. This also means a network that produces zero-valued objective function is the analytical solution of the PDE.

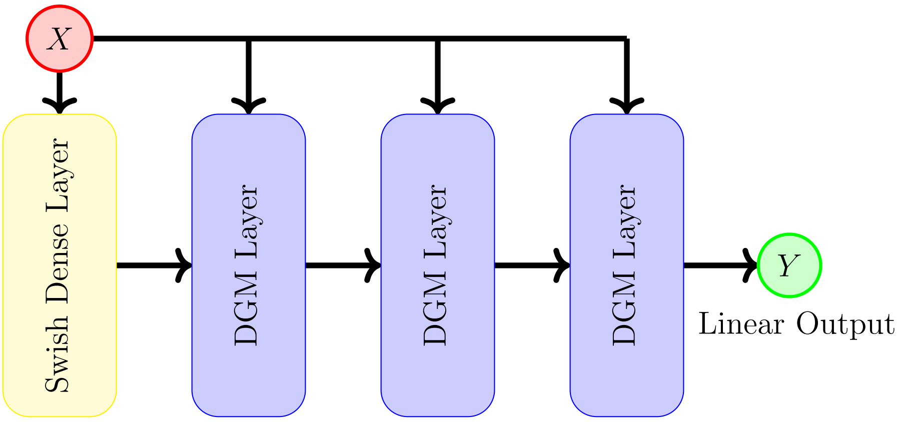
The network architecture adopted in [45] contains 1 dense layer and 3 DGM layers, all embedded with *tanh* activation function. We modify the structure and incorporate the *swish* activation function [40]. Detailed arguments on the effectiveness of using *swish* may be found in [12]. A summary of different types of activation function used in our network architecture is included in Table 6.5.1.

Table 6.5.1: Activation Functions

Sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}}$
Tanh	$\sigma(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$
Swish	$\sigma(x) = \frac{x}{1+e^{-x}}$

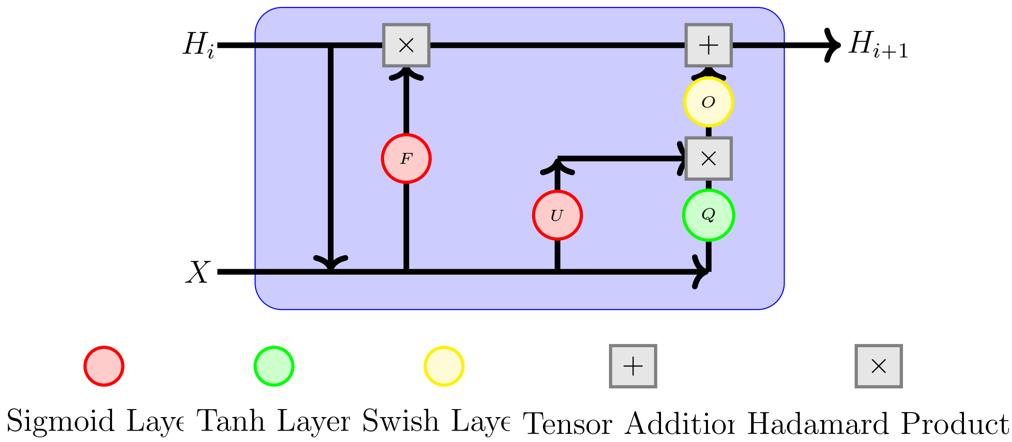
Figure 6.5.1 captures the DGM network structure.

Figure 6.5.1: DGM network architecture



Our modified DGM layer is inspired by *Gated Recurrent Unit* by Chung et al. [13] (2014). Figure 6.5.2 captures the structure of each modified DGM layer.

Figure 6.5.2: Modified DGM layer



The mathematical operation behind our entire DGM network can be represented by the following set of equations:

$$\begin{aligned}
H_1 &= Swish\big(W_0 X + b_0\big),\\
F_l &= Sigmoid\big(W_{fx,l}X + W_{fh,l}H_l + b_{f,l}\big), \quad \text{for } l = 1,2,3,\\
U_l &= Sigmoid\big(W_{ux,l}X + W_{uh,l}H_l + b_{u,l}\big),\\
Q_l &= Tanh\big(W_{qx,l}X + W_{qh,l}H_l + b_{q,l}\big),\\
O_l &= Swish\big(W_{ox,l}(U_l \circ O_l) + b_{o,l}\big).\\
H_{l+1} &= F_l \circ H_l + O_l,\\
Y &= H_4 W_y + b_y,
\end{aligned}$$

where W are the weights, b are the biases and $\circ$ is the Hadamard product.

For iteration size I and batch size B, we provide a general overview of the implementation of DGM in Algorithm 6.5.1.

Algorithm 6.5.1 Deep Galerkin Method for option pricing

Initialize learning rate α and network parameters θ
for $i = 1$ **to** I **do**
  Generate interior sample point $\mathbf{x}_i^{(i)} = [x_{i1}^{(i)}, x_{i1}^{(i)}, ..., x_{iB}^{(i)}]$ from ϕ_1
  Generate terminal sample point $\mathbf{x}_i^{(T)} = [x_{i1}^{(T)}, x_{i1}^{(T)}, ..., x_{iB}^{(T)}]$ from ϕ_2
  Generate boundary sample point $\mathbf{x}_i^{(b)} = [x_{i1}^{(b)}, x_{i1}^{(b)}, ..., x_{iB}^{(b)}]$ from ϕ_3
  Compute the loss function:
    $J(\theta) = \|\mathcal{L}f(\mathbf{x}_i^{(i)};\theta)\|^2 + \|f(\mathbf{x}_i^{(T)};\theta) - h(\mathbf{x}_i^{(T)})\|^2 + \|f(\mathbf{x}_i^{(b)};\theta) - g(\mathbf{x}_i^{(b)})\|^2$
  Take a descent step:
    $\theta^{(new)} = \theta^{(old)} - \alpha \frac{\partial J(\theta)}{\partial \theta^{(old)}}$
  Apply decay to the learning rate α
end for

6.6 Approximating Solution of Finite Liquidity Partial Differential Equation

If an undergraduate student is given the task of learning graduate material. It is unlikely the student will do very well. However, if that same student were to learn the prerequisites knowledge beforehand, and attempt again. That student certainly stands a better chance. In machine learning, this concept is often called *transfer learning*. It is the method of applying prior knowledge to related problems but often difficult to solve directly. Bengio (2012) [9] goes into extensive detail on transfer learning. Weiss et al. (2016) [48] provides a formal definition for this method in terms of a domain $\mathcal{D} = \{\mathcal{X}, \phi_{\mathcal{X}}\}$

and learning task $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$ ($\mathcal{X}$-feature space, $\phi_{\mathcal{X}}$-feature distribution, $\mathcal{Y}$-label space, $f(\cdot)$-predictive function).

Definition 2. *(**Transfer Learning by Weiss et al.**)*
For a pair of domain and learning task $\mathcal{D}_s = \{\mathcal{X}_s, \phi_{\mathcal{X}_s}\}$, $\mathcal{T}_s = \{\mathcal{Y}_s, f_s(\cdot)\}$. Consider a target domain and learning task $\mathcal{D}_t = \{\mathcal{X}_t, \phi_{\mathcal{X}_t}\}$, $\mathcal{T}_t = \{\mathcal{Y}_t, f_t(\cdot)\}$. Transfer learning is the process of using relevant information of $f_s(\cdot)$ to improve the predictive ability of $f_t(\cdot)$.

By adopting transfer learning, we may train DGM nets to learn the FLMM pricing PDEs (3.4.1). The aforementioned PDEs are special cases of the 2-dimensional BS PDE (1.4.3). Therefore, we should train an initial DGM net to learn this relatively simpler. Subsequently, we may modify the objective function (6.5.4) in accordance to the more complicated PDE with price impacts (3.4.1), and further train the network.

For some asset price cap C, we restrict the domain to the finite cube $[0, T] \times [0, C]^2$. This will allow us to impose asymptotics as boundary conditions (see (6.6) for more details), and in turn get a faster convergence. During implementation, we use mean squared error (MSE) as an estimator for the $\mathcal{L}^2$ norms in (6.5.4). In the calculation of MSE, N is the mini-batch size for training, it should be large to ensure the accuracy of the estimator.

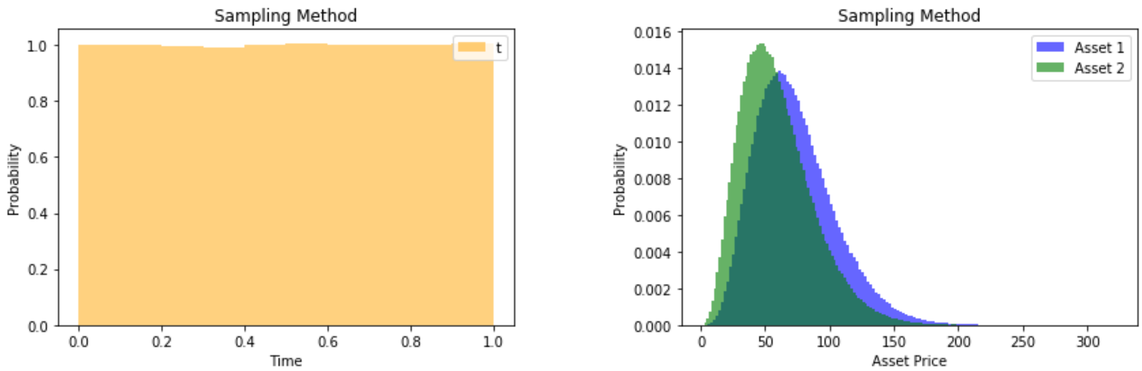
The sampling method is completely problem depended, user should focus sampling from a sub-region of likely option input parameters. In the case of Spread Option, we noticed the DGM net convergence faster when we choose sampling distribution that produce more non-zero option value. We present details on our sampling distributions for each scenario in Table 6.6.1.

Table 6.6.1: Sampling method

Domain	Variables	Distribution
$[0, T] \times [0, C]^2$	(t, s_1, s_2)	$t \sim \mathcal{U}(0, T)$, $s_1 \sim C\beta(3, 10)$, $s_2 \sim C\beta(2, 10)$
$[0, C]^2$	(s_1, s_2)	$s_1 \sim C\beta(3, 10)$, $s_2 \sim C\beta(2, 10)$
$[0, T] \times [0, C]$	(t, s_1)	$t \sim \mathcal{U}(0, T)$, $s_1 \sim C\beta(3, 10)$
$[0, T] \times [0, C]$	(t, s_1)	$t \sim \mathcal{U}(0, T)$, $s_1 \sim C\beta(3, 10)$
$[0, T] \times [0, C]$	(t, s_2)	$t \sim \mathcal{U}(0, T)$, $s_2 \sim C\beta(2, 10)$
$[0, T] \times [0, C]$	(t, s_2)	$t \sim \mathcal{U}(0, T)$, $s_2 \sim C\beta(2, 10)$

We present histograms for our sampling method in Figure 6.6.1. One may notice we sample t uniformly, this because we desire to learn the option price function evenly across a span of time to maturities. For the assets, we adopted two beta distributions, one slightly more centred than the other. The reason is that we desire the option value to be non-zero, therefore the first asset should be greater than the second asset with a high probability.

Figure 6.6.1: Sampling method



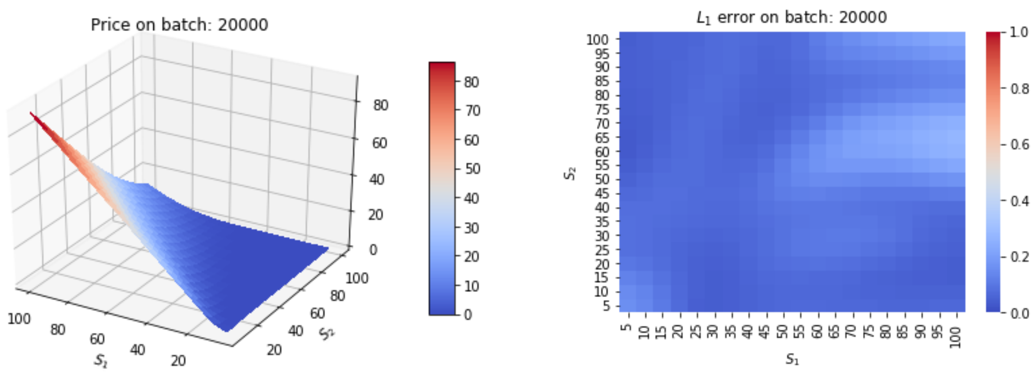
For Spread Option under BSM and FLMM, the shared option parameters we used are: $k = 4$, $r = 0.05$, $\rho = 0.5$, $\sigma_1 = 0.4$, $\sigma_2 = 0.2$ and $T = 1$. We are interested in having a model that works well in the input region $(s_1, s_2) \in [0, 100]^2$, then the asset price cap is set to $C = 600$. We set up the loss function in the following manner:

Spread Option under 2-dimensional BS model

$$
\begin{aligned}
J_1(\theta) &= \|\mathcal{L} f(t, s_1, s_2; \theta)\|^2_{[0,T]\times[0,C]^2}, \\
\mathcal{L} &= \frac{\partial}{\partial t} + \frac{1}{2}\sum_{i,j=1,2} \rho_{ij}\sigma_i\sigma_j s_i s_j \frac{\partial^2}{\partial s_i \partial s_j} + r\sum_{i=1,2} s_i \frac{\partial}{\partial s_i} - r, \\
J_2(\theta) &= \|(s_1 - s_2 - k)^+ - f(T, s_1, s_2; \theta)\|^2_{[0,C]^2}, \\
J_3(\theta) &= \|C - s_2 - ke^{-r(T-t)} - f(t, C, s_2; \theta)\|^2_{[0,T]\times[0,C]} \\
&\quad + \|s_1(d_+) - ke^{-r(T-t)}(d_-) - f(t, s_1, 0; \theta)\|^2_{[0,T]\times[0,C]} \\
&\quad + \|f(t, 0, s_2; \theta)\|^2_{[0,T]\times[0,C]} + \|f(t, 0, C; \theta)\|^2_{[0,T]\times[0,C]}.
\end{aligned}
$$

After training the first DGM network with the loss function above, we illustrate the results in Figure 6.6.2.

Figure 6.6.2: Spread option (BS model)



*Benchmarked against FFT [29] with grid size N=512.

As we see, the results from the trained net match extremely well with the FFT results. Furthermore, we observe our modified DGM net outperforms the canonical DGM net for this particular pricing PDE. An important cautions note is although the trained net performs well on $[0, 100]^2$, we should not expect the same level of performance will extend to $(\mathbb{R}^+)^2$.

Next, we take the previously trained model and apply transfer learning by switching the loss function to $\mathcal{L}^{(p)}(\theta)$.

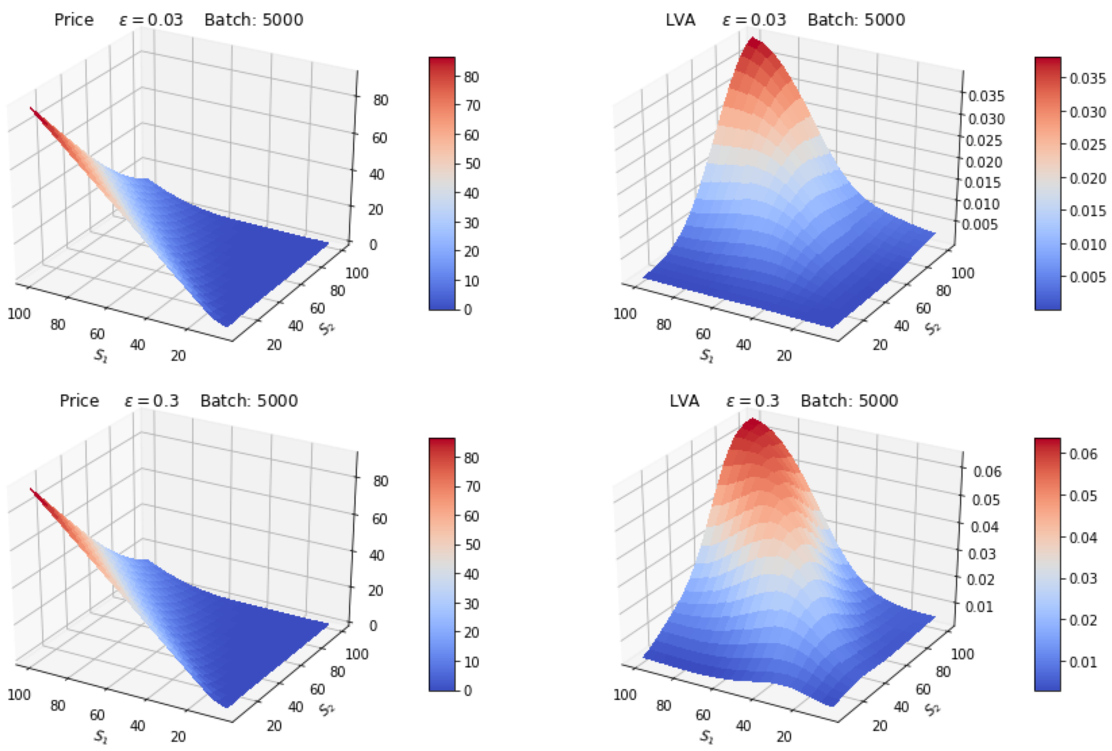
Spread Option under Partial Impact FLMM

$$
\begin{aligned}
J_1(\theta) &= \|\mathcal{L}^{(p)} f(t, s_1, s_2; \theta)\|^2_{[0,T]\times[0,C]^2}, \\
\mathcal{L}^{(p)} &= \frac{\partial}{\partial t} + \frac{\sigma_1^2 s_1^2 + \sigma_2^2 s_2^2 \lambda^2 (V^{(BS)}_{s_1 s_2})^2 + 2\rho\sigma_1\sigma_2 s_1 s_2 \lambda V^{(BS)}_{s_1 s_2}}{2(1-\lambda V^{(BS)}_{s_1 s_1})^2} \frac{\partial^2}{\partial s_1^2} \\
&\quad + \frac{\rho\sigma_1\sigma_2 s_1 s_2 + \sigma_2^2 s_2^2 \lambda V^{(BS)}_{s_1 s_2}}{1-\lambda V^{(BS)}_{s_1 s_1}} \frac{\partial^2}{\partial s_1 \partial s_2} + \frac{1}{2}\sigma_2^2 s_2^2 \frac{\partial^2}{\partial s_2^2} + r s_1 \frac{\partial}{\partial s_1} + r s_2 \frac{\partial}{\partial s_2} - r,
\end{aligned}
$$

It is clear from (3.4.1) that $J_2(\theta)$ and $J_3(\theta)$ will be the same as the ones from 6.6.

Results of the trained network is illustrated in Figure 6.6.3.

Figure 6.6.3: Spread option (Partial Impact FLMM)



*Benchmarked against FFT [29] with grid size N=512.

With the aid of transfer learning, we were able to get convergence in less than 5000 batches. A nearly impossible task for a network directly due to the PDE's complexity. From the result, we observe the liquidity valuation adjustment more prominent for at the money option with higher underlying.

Chapter 7

Summary and Future Work

In this thesis, we developed a mathematical framework for FLMM. By implementing various numerical methods, we were able to not only accurately calculate the options values, but also achieve satisfactory computation time.

The first option we examined under FLMM is the Exchange Option, for which we constructed 3 separate Monte Carlo estimators: Naive, Antithetic and Controlled. For each of the aforementioned MC estimators, we simulated the paths of $S_1(t)$ and $S_2(t)$ under Euler-Maruyama and Milstein schemes. In conclusion, the Controlled estimator performed the best when Margrabe Option was chosen as the controlled variate. We also concluded sampling with Milstein will achieve similar variance but at taking approximately 5% the compute time as E-M.

Then, we looked at the Spread Option under FLMM. For this option, We only constructed the Naive and Controlled MC estimator. During the implementation of the estimators, we simulated the paths of $S_1(t)$ and $S_2(t)$ only under E-M. The Spread Option Greeks required in the simulation were solved numerically from Fast Fourier Transform, the interpolated. In conclusion, the Controlled estimator was superior. The Margrabe Option was chosen as the controlled variate once again. We also concluded Margrabe Option as the control variate has a diminishing effect compared to the Exchange Option.

In the final chapter, we showed the Exchange Option value from FLMM can be accurately predicted by a Feed-Forward network. Then, we realized Feed-Forward networks tend to struggle more in its predictive capability as the order of the Greeks increase. We also implemented DGM net and able to numerically approximate the pricing PDE for Spread Option.

Moving forward, I would like to spend more time on the development of asset pricing models, on a wider variety of exotic options. I would also like to adopt more diversity machine learning methods into my work (such as Gradient Boasted Machines, Reinforcement Learning), then take a deeper dive into pricing and hedging methodologies. At last, I would like to validate and back test some of these models on real world data.

Appendix A

Greek Computations

A.1 Exchange Option Greeks

A.1.1 Delta

Proof:

$$
\begin{aligned}
\Delta_1(t) =& \frac{\partial V\big(t, s_1, s_2\big)}{\partial s_1} = N(d_+) + s_1 N'(d_+)\frac{\partial d_+}{\partial s_1} - s_2 N'(d_-)\frac{\partial d_-}{\partial s_1} \\
=& N(d_+) + N'(d_+)\frac{1}{\sigma\sqrt{T-t}} - \frac{s_2}{s_1}N'(d_+ - \sigma\sqrt{T-t})\frac{1}{\sigma\sqrt{T-t}} \\
=& N(d_+) + N'(d_+)\frac{1}{\sigma\sqrt{T-t}} \\
& - \frac{s_2}{s_1}\frac{1}{\sqrt{2\pi}}\exp\Big\{-\frac{1}{2}d_+^2 + d_+\sigma\sqrt{T-t} - \frac{1}{2}\sigma^2(T-t)\Big\}\frac{1}{\sigma\sqrt{T-t}} \\
=& N(d_+) + N'(d_+)\frac{1}{\sigma\sqrt{T-t}} - \frac{s_2}{s_1}\frac{1}{\sqrt{2\pi}}\exp\Big\{-\frac{1}{2}d_+^2 + \log\big(\frac{s_1}{s_2}\big)\Big\}\frac{1}{\sigma\sqrt{T-t}} \\
=& N(d_+) + N'(d_+)\frac{1}{\sigma\sqrt{T-t}} - N'(d_+)\frac{1}{\sigma\sqrt{T-t}} = N(d_+).
\end{aligned}
$$

$$
\begin{aligned}
\Delta_2(t) =& \frac{\partial V\big(t, s_1, s_2\big)}{\partial s_2} = s_1 N'(d_+)\frac{\partial d_+}{\partial s_2} - s_2 N'(d_-)\frac{\partial d_-}{\partial s_2} - N(d_-) \\
=& -\frac{s_1}{s_2}N'(d_- + \sigma\sqrt{T-t})\frac{1}{\sigma\sqrt{T-t}} + N'(d_-)\frac{1}{\sigma\sqrt{T-t}} - N(d_-) \\
=& -\frac{s_1}{s_2}\frac{1}{\sqrt{2\pi}}\exp\Big\{-\frac{1}{2}d_-^2 - d_-\sigma\sqrt{T-t} - \frac{1}{2}\sigma^2(T-t)\Big\}\frac{1}{\sigma\sqrt{T-t}} \\
& + N'(d_-)\frac{1}{\sigma\sqrt{T-t}} - N(d_-) \\
=& -\frac{s_1}{s_2}\frac{1}{\sqrt{2\pi}}\exp\Big\{-\frac{1}{2}d_-^2 - \log\big(\frac{s_1}{s_2}\big)\Big\}\frac{1}{\sigma\sqrt{T-t}} + N'(d_-)\frac{1}{\sigma\sqrt{T-t}} - N(d_-) \\
=& -N'(d_-)\frac{1}{\sigma\sqrt{T-t}} + N'(d_-)\frac{1}{\sigma\sqrt{T-t}} - N(d_-) = -N(d_-).
\end{aligned}
$$

□

A.1.2 Theta

Proof:

$$
\begin{aligned}
\Theta(t) =& \frac{\partial V\big(t, s_1, s_2\big)}{\partial \tau} = s_1 N'(d_+)\frac{\partial d_+}{\partial \tau} - s_2 N'(d_-)\frac{\partial d_-}{\partial \tau} \\
=& s_1 N'(d_+)\frac{\partial d_+}{\partial \tau} - s_2 N'(d_+ - \sigma\sqrt{T-t})\frac{\partial d_-}{\partial \tau} \\
=& s_1 N'(d_+)\frac{\partial d_+}{\partial \tau} - s_2 \frac{1}{\sqrt{2\pi}} \exp\Big\{ -\frac{1}{2}d_+^2 + d_+\sigma\sqrt{T-t} - \frac{1}{2}\sigma^2(T-t)\Big\}\frac{\partial d_-}{\partial \tau} \\
=& s_1 N'(d_+)\frac{\partial d_+}{\partial \tau} - s_2 \frac{1}{\sqrt{2\pi}} \exp\Big\{ -\frac{1}{2}d_+^2 + \log\big(\frac{s_1}{s_2}\big)\Big\}\frac{\partial d_-}{\partial \tau} \\
=& s_1 N'(d_+)\big(\frac{\partial d_+}{\partial \tau} - \frac{\partial d_-}{\partial \tau}\big) \\
=& \frac{\sigma}{2\sqrt{T-t}} s_1 N'(d_+) \quad \text{or} \quad -\frac{\sigma}{2\sqrt{T-t}} s_2 N'(d_-).
\end{aligned}
$$

□

A.1.3 Gamma

Proof:

$$
\begin{aligned}
\Gamma_{11}(t) =& \frac{\partial \Delta_1(t)}{\partial s_1} = N'(d_+)\frac{\partial d_+}{\partial s_1} = \frac{1}{\sigma\sqrt{T-t}}\frac{N'(d_+)}{s_1}. \\
\Gamma_{22}(t) =& \frac{\partial \Delta_2(t)}{\partial s_2} = -N'(d_-)\frac{\partial d_-}{\partial s_2} = \frac{1}{\sigma\sqrt{T-t}}\frac{N'(d_-)}{s_2}. \\
\Gamma_{12}(t) =& \frac{\partial \Delta_1(t)}{\partial s_2} = N'(d_+)\frac{\partial d_+}{\partial s_2} = -\frac{1}{\sigma\sqrt{T-t}}\frac{N'(d_+)}{s_2} = -\frac{1}{\sigma\sqrt{T-t}}\frac{N'(d_- + \sigma\sqrt{T-t})}{s_2} \\
=& -\frac{1}{\sigma\sqrt{T-t}}\frac{1}{s_2}\frac{1}{\sqrt{2\pi}} \exp\Big\{ -\frac{1}{2}d_-^2 - d_-\sigma\sqrt{T-t} - \frac{1}{2}\sigma^2(T-t)\Big\} \\
=& -\frac{1}{\sigma\sqrt{T-t}}\frac{1}{s_2}\frac{1}{\sqrt{2\pi}} \exp\Big\{ -\frac{1}{2}d_-^2 - \log\big(\frac{s_1}{s_2}\big)\Big\} \\
=& -\frac{1}{\sigma\sqrt{T-t}}\frac{N'(d_-)}{s_1} = \Gamma_{21}(t).
\end{aligned}
$$

□

A.1.4 Charm

Proof:

$$Charm_1(t) = \frac{\partial \Delta_1(t)}{\partial \tau} = N'(d_+)\frac{\partial d_+}{\partial \tau} = N'(d_+)\Big(- \frac{\log\left(\frac{s_1}{s_2}\right)}{2\sigma(T-t)^{\frac{3}{2}}} + \frac{\sigma}{4\sqrt{T-t}}\Big).$$
$$Charm_2(t) = \frac{\partial \Delta_2(t)}{\partial \tau} = -N'(d_-)\frac{\partial d_-}{\partial \tau} = N'(d_-)\Big(\frac{\log\left(\frac{s_1}{s_2}\right)}{2\sigma(T-t)^{\frac{3}{2}}} + \frac{\sigma}{4\sqrt{T-t}}\Big).$$

□

A.1.5 Speed

Proof:

$$Speed_{111}(t) = \frac{\partial \Gamma_{11}(t)}{\partial s_1} = \frac{1}{\sigma\sqrt{T-t}}\frac{N''(d_+)\frac{\partial d_+}{\partial s_1} - N'(d_+)}{s_1^2} = \frac{1}{\sigma\sqrt{T-t}}\frac{-\frac{2d_+N'(d_+)}{\sigma s_1\sqrt{T-t}} - N'(d_+)}{s_1^2}$$
$$= - \frac{N'(d_+)}{\sigma s_1^2\sqrt{T-t}}\big(\frac{2d_+}{\sigma s_1\sqrt{T-t}} + 1\big).$$
$$Speed_{222}(t) = \frac{\partial \Gamma_{22}(t)}{\partial s_2} = \frac{1}{\sigma\sqrt{T-t}}\frac{N''(d_-)\frac{\partial d_-}{\partial s_2} - N'(d_-)}{s_2^2} = \frac{1}{\sigma\sqrt{T-t}}\frac{-\frac{2d_-N'(d_-)}{\sigma s_2\sqrt{T-t}} - N'(d_-)}{s_2^2}$$
$$= - \frac{N'(d_-)}{\sigma s_2^2\sqrt{T-t}}\big(\frac{2d_-}{\sigma s_2\sqrt{T-t}} + 1\big).$$
$$Speed_{112}(t) = \frac{\partial \Gamma_{11}(t)}{\partial s_2} = \frac{1}{\sigma\sqrt{T-t}}\frac{N''(d_+)\frac{\partial d_+}{\partial s_2}}{s_1} = -\frac{2d_+N'(d_+)}{\sigma^2(T-t)s_1s_2}.$$
$$Speed_{221}(t) = \frac{\partial \Gamma_{22}(t)}{\partial s_1} = \frac{1}{\sigma\sqrt{T-t}}\frac{N''(d_-)\frac{\partial d_-}{\partial s_1}}{s_2} = -\frac{2d_-N'(d_-)}{\sigma^2(T-t)s_1s_2}.$$

□

A.1.6 Colour

Proof:

$$Colour_{11}(t) = \frac{\partial \Gamma_{11}(t)}{\partial \tau} = \frac{1}{\sigma s_1}\Big(- \frac{1}{2(T-t)^{\frac{3}{2}}}N'(d_+) - \frac{1}{(T-t)^{\frac{1}{2}}}N'(d_+)d_+\frac{\partial d_+}{\partial \tau}\Big)$$
$$= \frac{N'(d_+)}{2\sigma(T-t)^{\frac{3}{2}}s_1}\Big\{ - 1 + d_+\Big(\log(\frac{s_1}{s_2})\frac{1}{\sigma\sqrt{T-t}} - \frac{1}{2}\sigma\sqrt{T-t}\Big)\Big\}$$
$$= - \frac{N'(d_+)}{2^3\sigma^3(T-t)^{\frac{5}{2}}s_1}\Big(\sigma^4(T-t)^2 + 4\sigma^2(T-t) - 4\log^2(\frac{s_1}{s_2})\Big),$$

$$
\begin{aligned}
Colour_{22}(t) =& \frac{\partial \Gamma_{22}(t)}{\partial \tau} = \frac{1}{\sigma s_2}\Big(- \frac{1}{2(T-t)^{\frac{3}{2}}} N'(d_-) - \frac{1}{(T-t)^{\frac{1}{2}}} N'(d_-) d_+ \frac{\partial d_-}{\partial \tau}\Big) \\
=& - \frac{N'(d_-)}{2^3\sigma^3 (T-t)^{\frac{5}{2}} s_2}\Big(\sigma^4 (T-t)^2 + 4\sigma^2 (T-t) - 4\log^2(\frac{s_1}{s_2})\Big),
\end{aligned}
$$

$$
\begin{aligned}
Colour_{12}(t) =& \frac{\partial \Gamma_{12}(t)}{\partial \tau} = \frac{1}{\sigma s_2}\Big(\frac{1}{2(T-t)^{\frac{3}{2}}} N'(d_+) + \frac{1}{(T-t)^{\frac{1}{2}}} N'(d_+) d_+ \frac{\partial d_+}{\partial \tau}\Big) \\
=& \frac{N'(d_+)}{2^3\sigma^3 (T-t)^{\frac{5}{2}} s_2}\Big(\sigma^4 (T-t)^2 + 4\sigma^2 (T-t) - 4\log^2(\frac{s_1}{s_2})\Big) \\
=& \frac{N'(d_-)}{2^3\sigma^3 (T-t)^{\frac{5}{2}} s_1}\Big(\sigma^4 (T-t)^2 + 4\sigma^2 (T-t) - 4\log^2(\frac{s_1}{s_2})\Big) \\
=& Colour_{21}(t).
\end{aligned}
$$

□

A.1.7 Acceleration

Proof:

$$
\begin{aligned}
& Acceleration_{1111}(t) \\
=& \frac{\partial Speed_{111}(t)}{\partial s_1} = -\Big(\frac{\partial \frac{\Gamma_{11}}{s_1}(t)}{\partial s_1}\big(\frac{2d_+}{\sigma s_1\sqrt{T-t}} + 1\big) + \frac{\Gamma_{11}(t)}{s_1}\frac{2}{\sigma\sqrt{T-t}}\frac{\partial \frac{d_+}{s_1}(t)}{\partial s_1}\Big) \\
=& -\Big(\frac{Speed_{111}(t)s_1 - \Gamma_{11}(t)}{s_1^2}\big(\frac{2d_+}{\sigma s_1\sqrt{T-t}} + 1\big) + \frac{\Gamma_{11}(t)}{s_1}\frac{2}{\sigma\sqrt{T-t}}\big(\frac{\frac{1}{\sigma\sqrt{T-t}} - d_+}{s_1^2}\big)\Big) \\
=& -\frac{2\Gamma_{11}(t)}{\sigma\sqrt{T-t}s_1^3}\Big(d_+\big(\frac{2d_+}{\sigma\sqrt{T-t}s_1} + 1\big) + \big(\frac{1}{\sigma\sqrt{T-t}} - d_+\big)\Big) \\
=& -\frac{2\Gamma_{11}(t)}{\sigma^2(T-t)s_1^3}\big(\frac{2d_+^2}{s_1} + 1\big),
\end{aligned}
$$

$$
\begin{aligned}
& Acceleration_{1112}(t) \\
=& \frac{\partial Speed_{111}(t)}{\partial s_2} = -\Big(\frac{1}{s_1}\frac{\partial \Gamma_{11}}{\partial s_2}\big(\frac{2d_+}{\sigma s_1\sqrt{T-t}} + 1\big) + \frac{\Gamma_{11}(t)}{s_1}\frac{2}{\sigma\sqrt{T-t}s_1}\frac{\partial d_+}{\partial s_2}\Big) \\
=& -\Big(\frac{Speed_{112}(t)}{s_1}\big(\frac{2d_+}{\sigma\sqrt{T-t}s_1} + 1\big) - \frac{\Gamma_{11}(t)}{s_1}\frac{2}{\sigma\sqrt{T-t}s_1}\frac{1}{\sigma\sqrt{T-t}s_2}\Big) \\
=& \frac{2\Gamma_{11}(t)}{\sigma\sqrt{T-t}s_1 s_2}\big(\frac{2d_+^2}{\sigma\sqrt{T-t}s_1} + d_+ + \frac{1}{\sigma\sqrt{T-t}s_1}\big),
\end{aligned}
$$

$$
\begin{aligned}
&Acceleration_{1122}(t)\\
&= \frac{\partial Speed_{112}(t)}{\partial s_2} = \frac{2}{\sigma\sqrt{T-t}s_1}\Big(\frac{\partial d_+}{\partial s_2}(t)\Gamma_{12}(t) + d_+\frac{\partial \Gamma_{12}}{\partial s_2}(t)\Big)\\
&= \frac{2\Gamma_{12}}{\sigma^2(T-t)s_1 s_2}\big(d_+ d_- - 1\big),
\end{aligned}
$$

$$
\begin{aligned}
&Acceleration_{1222}(t)\\
&= \frac{\partial Speed_{222}(t)}{\partial s_1} = -\Big(\frac{1}{s_2}\frac{\partial \Gamma_{22}}{\partial s_1}\big(\frac{2d_-}{\sigma s_2\sqrt{T-t}} + 1\big) + \frac{\Gamma_{22}(t)}{s_2}\frac{2}{\sigma\sqrt{T-t}s_2}\frac{\partial d_-}{\partial s_1}\Big)\\
&= -\Big(\frac{Speed_{122}(t)}{s_2}\big(\frac{2d_-}{\sigma\sqrt{T-t}s_2} + 1\big) + \frac{\Gamma_{22}(t)}{s_2}\frac{2}{\sigma\sqrt{T-t}s_2}\frac{1}{\sigma\sqrt{T-t}s_1}\Big)\\
&= \frac{2\Gamma_{22}(t)}{\sigma\sqrt{T-t}s_1 s_2}\big(\frac{2d_-^2}{\sigma\sqrt{T-t}s_2} + d_- - \frac{1}{\sigma\sqrt{T-t}s_2}\big),
\end{aligned}
$$

$$
\begin{aligned}
&Acceleration_{2222}(t)\\
&= \frac{\partial Speed_{222}(t)}{\partial s_2} = -\Big(\frac{\partial \frac{\Gamma_{22}}{S_2}(t)}{\partial s_2}\big(\frac{2d_-}{\sigma s_2\sqrt{T-t}} + 1\big) + \frac{\Gamma_{22}(t)}{s_2}\frac{2}{\sigma\sqrt{T-t}}\frac{\partial \frac{d_-}{s_2}(t)}{\partial s_2}\Big)\\
&= -\frac{2\Gamma_{22}(t)}{\sigma^2(T-t)s_2^3}\big(\frac{2d_-^2}{s_2} + 1\big).
\end{aligned}
$$

□

A.2 Spread Option Greeks

A.2.1 Delta

Proof: Since $\mathbf{x} = log(\mathbf{s})$ and

$$
\mathbf{\Delta}(t) = \frac{\partial V_{spr}(t, \mathbf{s})}{\partial \mathbf{x}} = \frac{\partial \mathbf{x}}{\partial \mathbf{s}}\frac{\partial V_{spr}(t, \mathbf{s})}{\partial \mathbf{x}},
$$

if we let

$$
\mathcal{H} = \frac{\partial \mathbf{x}}{\partial \mathbf{s}} = \begin{bmatrix} \frac{1}{s_1} & 0 \\ 0 & \frac{1}{s_2} \end{bmatrix},
$$

then by taking the matrix derivative of (2.2.2), we have

$$
\mathbf{\Delta}(t) = (2\pi)^{-2}e^{-r\tau}K\mathcal{H}\frac{\partial}{\partial \mathbf{x}}\Big(\int\int_{\mathbb{R}^2+i\epsilon} e^{i\mathbf{u}'\mathbf{X}(t)}\Phi(\mathbf{u},\tau)\hat{P}(\mathbf{u})d\mathbf{u}\Big).
$$

By Dominated Convergence theorem, the order of differentiation and integration can be switched, and it follows

$$\begin{aligned}\mathbf{\Delta}(t) &= (2\pi)^{-2}e^{-r\tau}K\mathcal{H}\int\int_{\mathbb{R}^2+i\epsilon}\frac{\partial}{\partial \mathbf{x}}\Big(e^{i\mathbf{u}'\mathbf{X}(t)}\Big)\Phi(\mathbf{u},\tau)\hat{P}(\mathbf{u})d\mathbf{u}\\ &= (2\pi)^{-2}e^{-r\tau}K\mathcal{H}\int\int_{\mathbb{R}^2+i\epsilon} i\mathbf{u}e^{i\mathbf{u}'\mathbf{X}(t)}\Phi(\mathbf{u},\tau)\hat{P}(\mathbf{u})d\mathbf{u},\end{aligned}$$

if we let

$$\bar{\mathbf{\Delta}}(t) = \int\int_{\mathbb{R}^2+i\epsilon} i\mathbf{u}e^{i\mathbf{u}'\mathbf{X}(t)}\Phi(\mathbf{u},\tau)\hat{P}(\mathbf{u})d\mathbf{u},$$

then we have

$$\mathbf{\Delta}(t) = (2\pi)^{-2}e^{-r\tau}K\mathcal{H}\bar{\mathbf{\Delta}}(t) = (2\pi)^{-2}e^{-r\tau}K\begin{bmatrix}\frac{1}{s_1}\bar{\mathbf{\Delta}}_1\\ \frac{1}{s_2}\bar{\mathbf{\Delta}}_2\end{bmatrix}(t).$$

□

A.2.2 Theta

Proof:

$$\begin{aligned}\Theta(t) = \frac{\partial V_{spr}(t,\mathbf{s})}{\partial \tau} &= (2\pi)^{-2}K\big\{e^{-r\tau}\int\int_{\mathbb{R}^2+i\epsilon} e^{i\mathbf{u}'\mathbf{X}(t)}\frac{\partial}{\partial\tau}\Phi(\mathbf{u},\tau)\hat{P}(\mathbf{u})d\mathbf{u}\\ &- re^{-r\tau}\int\int_{\mathbb{R}^2+i\epsilon} e^{i\mathbf{u}'\mathbf{X}(t)}\frac{\partial}{\partial\tau}\Phi(\mathbf{u},\tau)\hat{P}(\mathbf{u})d\mathbf{u}\big\}\\ &= (2\pi)^{-2}Ke^{-r\tau}\int\int_{\mathbb{R}^2+i\epsilon}\Big(i\mathbf{u}'\big(r\mathbf{1}-\frac{1}{2}(\sigma_1^2+\sigma_2^2)\big)-\frac{1}{2}\mathbf{u}'\Sigma\mathbf{u}-r\Big)e^{i\mathbf{u}'\mathbf{X}(t)}\Phi(\mathbf{u},T)\hat{P}(\mathbf{u})d\mathbf{u},\end{aligned}$$

let

$$\bar{\mathbf{\Theta}}(t) = \int\int_{\mathbb{R}^2+i\epsilon}\Big(i\mathbf{u}'\big(r\mathbf{1}-\frac{1}{2}(\sigma_1^2+\sigma_2^2)\big)-\frac{1}{2}\mathbf{u}'\Sigma\mathbf{u}-r\Big)e^{i\mathbf{u}'\mathbf{X}(t)}\Phi(\mathbf{u},\tau)\hat{P}(\mathbf{u})d\mathbf{u},$$

then we have

$$\mathbf{\Theta}(t) = (2\pi)^{-2}Ke^{-r\tau}\bar{\mathbf{\Theta}}(t).$$

□

A.2.3 Gamma

Proof:

$$\mathbf{\Gamma}(t) = \frac{\partial\mathbf{\Delta}(t)}{\partial\mathbf{s}} = (2\pi)^{-2}e^{-r(T-t)}K\Big(\frac{\partial\mathcal{H}}{\partial\mathbf{s}}\bar{\mathbf{\Delta}}(t)+\mathcal{H}\frac{\partial\bar{\mathbf{\Delta}}(t)}{\partial\mathbf{x}}\frac{\partial\mathbf{x}}{\partial\mathbf{s}}\Big),$$

let

$$\bar{\mathbf{\Gamma}}(t) = \int\int_{\mathbb{R}^2+i\epsilon} (\mathbf{u}\bar{\otimes}\mathbf{u}) e^{i\mathbf{u}'\mathbf{X}(t)} \Phi(\mathbf{u},\tau)\hat{P}(\mathbf{u})d\mathbf{u}, \quad \text{and}$$

$$\mathcal{T}^{(3)} = \begin{bmatrix} \frac{1}{s_1^2} & 0 \\ 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{s_2^2} \end{bmatrix},$$

then we have

$$\begin{aligned}
\mathbf{\Gamma}(t) &= (2\pi)^{-2} e^{-r(T-t)} K \Big\{ -\mathcal{T}^{(3)} \int\int_{\mathbb{R}^2+i\epsilon} i\mathbf{u} e^{i\mathbf{u}'\mathbf{X}(t)} \Phi(\mathbf{u},\tau)\hat{P}(\mathbf{u})d\mathbf{u} \\
&- \mathcal{H}\Big(\int\int_{\mathbb{R}^2+i\epsilon} (\mathbf{u}\bar{\otimes}\mathbf{u}) e^{i\mathbf{u}'\mathbf{X}(t)} \Phi(\mathbf{u},\tau)\hat{P}(\mathbf{u})d\mathbf{u}\Big)\mathcal{H}\Big\} \\
&= -(2\pi)^{-2} e^{-r(T-t)} K \Big(\mathcal{T}^{(3)} \bar{\mathbf{\Delta}}(t) + \mathcal{H}\bar{\mathbf{\Gamma}}(t)\mathcal{H} \Big) \\
&= -(2\pi)^{-2} e^{-r(T-t)} K \begin{bmatrix} \frac{1}{s_1^2}(\bar{\Delta}_1 + \bar{\Gamma}_{11}) & \frac{1}{s_1 s_2}\bar{\Gamma}_{12} \\ \frac{1}{s_1 s_2}\bar{\Gamma}_{21} & \frac{1}{s_2^2}(\bar{\Delta}_2 + \bar{\Gamma}_{22}) \end{bmatrix}(t)
\end{aligned}$$

□

A.2.4 Charm

Proof:

$$\begin{aligned}
\mathbf{Chm}(t) = \frac{\partial \mathbf{\Delta}(t)}{\partial \tau} &= (2\pi)^{-2} K \mathcal{H} \frac{\partial}{\partial \tau} \Big(e^{-r\tau} \int\int_{\mathbb{R}^2+i\epsilon} e^{i\mathbf{u}'\mathbf{X}(t)} \Phi(\mathbf{u},\tau)\hat{P}(\mathbf{u})d\mathbf{u} \Big) \\
&= (2\pi)^{-2} e^{-r(T-t)} K \mathcal{H} \int\int_{\mathbb{R}^2+i\epsilon} \Big(i\mathbf{u}'\big(r\mathbf{1} - \frac{1}{2}(\sigma_1^2 + \sigma_2^2)\big) - \frac{1}{2}\mathbf{u}'\Sigma\mathbf{u} - r \Big) \\
&\times i\mathbf{u} e^{i\mathbf{u}'\mathbf{X}(t)} \Phi(\mathbf{u},\tau)\hat{P}(\mathbf{u})d\mathbf{u},
\end{aligned}$$

let

$$\overline{\mathbf{Chm}}(t) = \int\int_{\mathbb{R}^2+i\epsilon} \Big(i\mathbf{u}'\big(r\mathbf{1} - \frac{1}{2}(\sigma_1^2 + \sigma_2^2)\big) - \frac{1}{2}\mathbf{u}'\Sigma\mathbf{u} - r \Big) i\mathbf{u} e^{i\mathbf{u}'\mathbf{X}(t)} \Phi(\mathbf{u},\tau)\hat{P}(\mathbf{u})d\mathbf{u},$$

then we have

$$\mathbf{Chm}(t) = (2\pi)^{-2} e^{-r(T-t)} K \mathcal{H} \overline{\mathbf{Chm}}(t) = (2\pi)^{-2} e^{-r(T-t)} K \begin{bmatrix} \frac{1}{s_1}\overline{Chm}_1 \\ \frac{1}{s_2}\overline{Chm}_2 \end{bmatrix}(t).$$

□

A.2.5 Speed

Proof:

$$\begin{aligned}
\mathbf{Spd}(t) = \frac{\partial \mathbf{\Gamma}(t)}{\partial \mathbf{s}} &= -(2\pi)^{-2} e^{-r(T-t)} K \Big\{ \frac{\partial \mathcal{T}^{(3)}}{\partial \mathbf{s}} \bar{\mathbf{\Delta}}(t) + \mathcal{T}^{(3)} \frac{\partial \bar{\mathbf{\Delta}}(t)}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{s}} \\
&+ \frac{\partial \mathcal{H}}{\partial \mathbf{s}} \bar{\mathbf{\Gamma}}(t)\mathcal{H} + \mathcal{H}\Big(\frac{\partial \bar{\mathbf{\Gamma}}(t)}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{s}} \mathcal{H} + \bar{\mathbf{\Gamma}}(t) \frac{\partial \mathcal{H}}{\partial \mathbf{s}} \Big) \Big\},
\end{aligned}$$

let

$$\overline{\mathbf{Spd}}(t) = \int\int_{\mathbb{R}^2+i\epsilon} i(\mathbf{u}\bar{\otimes}\mathbf{u}\bar{\otimes}\mathbf{u})e^{i\mathbf{u'X}(t)}\Phi(\mathbf{u},\tau)\hat{P}(\mathbf{u})d\mathbf{u}, \quad \text{and}$$

$$\mathcal{T}^{(4)} = \Big(\begin{bmatrix}\frac{1}{s_1^3} & 0\\ 0 & 0\end{bmatrix}\otimes\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\Big)\otimes\Big(\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\otimes\begin{bmatrix}0 & 0\\ 0 & \frac{1}{s_2^3}\end{bmatrix}\Big),$$

then we have,

$$\begin{aligned}\mathbf{Spd}(t) &= -(2\pi)^{-2}e^{-r(T-t)}K\Big\{-2\mathcal{T}^{(4)}\bar{\mathbf{\Delta}}(t) + \mathcal{T}^{(3)}\bar{\mathbf{\Gamma}}(t)\mathcal{H} - \mathcal{T}^{(3)}\bar{\mathbf{\Gamma}}(t)\mathcal{H}\\ &\quad + \mathcal{H}\Big(\Big(\int\int_{\mathbb{R}^2+i\epsilon} i(\mathbf{u}\bar{\otimes}\mathbf{u}\bar{\otimes}\mathbf{u})e^{i\mathbf{u'X}(t)}\Phi(\mathbf{u},\tau)\hat{P}(\mathbf{u})d\mathbf{u}\Big)\mathcal{H}^2 - \bar{\mathbf{\Gamma}}(t)\mathcal{T}^{(3)}\Big)\Big\}\\ &= (2\pi)^{-2}e^{-r(T-t)}K\Big(2\mathcal{T}^{(4)}\bar{\mathbf{\Delta}}(t) + \mathcal{H}\bar{\mathbf{\Gamma}}(t)\mathcal{T}^{(3)} - \mathcal{H}\overline{\mathbf{Spd}}(t)\mathcal{H}^2\Big)\\ &=(2\pi)^{-2}e^{-r(T-t)}K\Big\{\begin{bmatrix}\frac{1}{s_1^3}(2\bar{\Delta}_1 + \bar{\Gamma}_{11} - \overline{Spd}_{111}) & -\frac{1}{s_1^2 s_2}\overline{Spd}_{121}\\ \frac{1}{s_1^2 s_2}(\bar{\Gamma}_{21} - \overline{Spd}_{211}) & -\frac{1}{s_1 s_2^2}\overline{Spd}_{221}\end{bmatrix}\\ &\quad\otimes\begin{bmatrix}-\frac{1}{s_1^2 s_2}\overline{Spd}_{112} & \frac{1}{s_1^2 s_2}(\bar{\Gamma}_{12} - \overline{Spd}_{122})\\ -\frac{1}{s_1 s_2^2}\overline{Spd}_{212} & \frac{1}{s_2^3}(2\bar{\Delta}_2 + \bar{\Gamma}_{22} - \overline{Spd}_{222})\end{bmatrix}\Big\}(t)\end{aligned}$$

□

A.2.6 Acceleration

Proof:

$$\begin{aligned}\mathbf{Acc}(t) = \frac{\partial\mathbf{Spd}(t)}{\partial\mathbf{s}} &= (2\pi)^{-2}e^{-r(T-t)}K\Big\{2\Big(\frac{\partial\mathcal{T}^{(4)}}{\partial\mathbf{s}}\bar{\mathbf{\Delta}}(t) + \mathcal{T}^{(4)}\frac{\partial\mathbf{x}}{\partial\mathbf{s}}\frac{\partial\bar{\mathbf{\Delta}}(t)}{\partial\mathbf{x}}\Big)\\ &\quad + \frac{\partial\mathcal{T}^{(3)}}{\partial\mathbf{s}}\mathcal{H}\bar{\mathbf{\Gamma}}(t) + \mathcal{T}^{(3)}\Big(\frac{\partial\mathcal{H}}{\partial\mathbf{s}}\bar{\mathbf{\Gamma}}(t) + \mathcal{H}\frac{\partial\mathbf{x}}{\partial\mathbf{s}}\frac{\partial\bar{\mathbf{\Gamma}}(t)}{\partial\mathbf{x}}\Big)\\ &\quad - \frac{\partial\mathcal{H}^3}{\partial\mathbf{s}}\overline{\mathbf{Spd}}(t) - \mathcal{H}^3\frac{\partial\mathbf{x}}{\partial\mathbf{s}}\frac{\partial\overline{\mathbf{Spd}}(t)}{\partial\mathbf{x}}\Big\},\end{aligned}$$

let

$$\overline{\mathbf{Acc}}(t) = \int\int_{\mathbb{R}^2+i\epsilon}(\mathbf{u}\bar{\otimes}\mathbf{u}\bar{\otimes}\mathbf{u}\bar{\otimes}\mathbf{u})e^{i\mathbf{u'X}(t)}\Phi(\mathbf{u},\tau)\hat{P}(\mathbf{u})d\mathbf{u}, \quad \text{and}$$

$$\begin{aligned}\mathcal{T}^{(5)} = \Big\{&\Big(\begin{bmatrix}\frac{1}{s_1^4} & 0\\ 0 & 0\end{bmatrix}\otimes\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\Big)\otimes\Big(\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\otimes\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\Big),\\ &\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\otimes\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\Big)\otimes\Big(\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\otimes\begin{bmatrix}0 & 0\\ 0 & \frac{1}{s_2^4}\end{bmatrix}\Big)\Big\},\end{aligned}$$

then we have,

$$
\begin{aligned}
&\mathbf{Acc}(t)\\
&= (2\pi)^{-2}e^{-r(T-t)}K\Big\{2\Big(-3\mathcal{T}^{(5)}\bar{\boldsymbol{\Delta}}(t)+\mathcal{T}^{(4)}\mathcal{H}\bar{\boldsymbol{\Gamma}}(t)\Big)-2\mathcal{T}^{(4)}\mathcal{H}\bar{\boldsymbol{\Gamma}}(t)\\
&+\mathcal{T}^{(3)}\Big(-\mathcal{T}^{(3)}\bar{\boldsymbol{\Gamma}}(t)-\mathcal{H}^2\overline{\mathbf{Spd}}(t)\Big)+3\mathcal{T}^{(3)}\mathcal{H}^2\overline{\mathbf{Spd}}(t)-\mathcal{H}^4\overline{\mathbf{Acc}}(t)\Big\}\\
&= (2\pi)^{-2}e^{-r(T-t)}K\Big\{-6\mathcal{T}^{(5)}\bar{\boldsymbol{\Delta}}(t)-\mathcal{T}^{(4)}\mathcal{H}\bar{\boldsymbol{\Gamma}}(t)+2\mathcal{T}^{(3)}\mathcal{H}^2\overline{\mathbf{Spd}}(t)+\mathcal{H}^4\overline{\mathbf{Acc}}(t)\Big\}\\
&= (2\pi)^{-2}e^{-r(T-t)}K\\
&\Big\{\Big(\begin{bmatrix}\frac{1}{s_1^4}(-6\bar{\Delta}_1-\bar{\Gamma}_{11}+2\overline{Spd_{111}}+\overline{Acc_{1111}}) & \frac{1}{s_1^4}(-\bar{\Gamma}_{12}+2\overline{Spd_{121}}+\overline{Acc_{1211}})\\ \frac{1}{s_2^4}\overline{Acc_{2111}} & \frac{1}{s_2^4}\overline{Acc_{2211}}\end{bmatrix}\\
&\otimes\begin{bmatrix}\frac{1}{s_1^4}\overline{Acc_{1121}} & \frac{1}{s_1^4}\overline{Acc_{1221}}\\ \frac{1}{s_2^4}(2\overline{Spd_{211}}+\overline{Acc_{2121}}) & \frac{1}{s_2^4}(2\overline{Spd_{221}}+\overline{Acc_{2221}})\end{bmatrix}\Big)\\
&\otimes\Big(\begin{bmatrix}\frac{1}{s_1^4}(2\overline{Spd_{112}}+\overline{Acc_{1112}}) & \frac{1}{s_1^4}(2\overline{Spd_{122}}+\overline{Acc_{1212}})\\ \frac{1}{s_2^4}\overline{Acc_{2112}} & \frac{1}{s_2^4}\overline{Acc_{2212}}\end{bmatrix}\otimes\\
&\begin{bmatrix}\frac{1}{s_1^4}\overline{Acc_{1122}} & \frac{1}{s_1^4}\overline{Acc_{1222}}\\ \frac{1}{s_2^4}(-\bar{\Gamma}_{21}+2\overline{Spd_{212}}+\overline{Acc_{2122}}) & \frac{1}{s_2^4}(-6\bar{\Delta}_2-\bar{\Gamma}_{22}+2\overline{Spd_{222}}+\overline{Acc_{2222}})\end{bmatrix}\Big)\Big\}(t)
\end{aligned}
$$

□

Appendix B

Existence and Uniqueness Proofs

B.1 Finite Liquidity Existence and Uniqueness Theorem I

In this section, $\|\cdot\|$ and $|||\cdot|||$ represents the supremum norms:

$$\|f\| = \sup_{(s_1,s_2)\in\mathcal{D}_1} |f(t,s_1,s_2)|, \quad \text{where } \mathcal{D}_1 = \big(\mathbb{R}^+\big)^2,$$
$$|||f||| = \sup_{(t,s_1,s_2)\in\mathcal{D}_2} |f(t,s_1,s_2)|, \quad \text{where } \mathcal{D}_2 = [0,T]\times\big(\mathbb{R}^+\big)^2,$$

The following combination of conditions $(1)-(6)$ will guarantee existence and uniqueness of a strong solution for FLMM.

(1) $\|\lambda(s_1 f_{s_1s_1} + s_1 f_{s_1s_2} + f_{s_2} + s_2 f_{s_2} + s_2 f_{s_1s_1} + s_2 f_{s_1s_2} + s_2 f_{s_2s_2})\| < \infty.$ (B.1.1)

(2) $\|\big(\lambda_{s_1} + \lambda_{s_2}\big)\big(s_1 f_{s_1} + s_2 f_{s_1} + s_2 f_{s_2}\big)\| < \infty.$ (B.1.2)

(3) $|||1-\lambda f_{s_1}||| > \delta_0,$ for some $\delta_0 > 0.$ (B.1.3)

(4) $\|(\lambda + \lambda_{s_1} + \lambda_{s_2})(f_t + f_{s_1} + f_{s_2} + f_{s_1s_1} + f_{s_1s_2} + f_{s_2s_2} + f_{s_1s_1s_2} + f_{s_1s_2s_2})\| < \infty.$

(5) $\|\lambda(f_{ts_1} + f_{ts_2} + f_{s_1s_1s_1} + f_{s_2s_2s_2}) + \lambda_{s_1} f_{s_1s_1s_1} + \lambda_{s_2} f_{s_2s_2s_2}\| < \infty.$ (B.1.4)

(6) $\|s_1 f_{s_1s_1} + s_1 f_{s_1s_2} + s_2 f_{s_1s_2} + s_2 f_{s_2s_2} + s_1^2 f_{s_1s_1s_2} + s_1^2 f_{s_1s_2s_2} + s_1 s_2 f_{s_1s_1s_2}$ (B.1.5)
$+ s_1 s_2 f_{s_1s_2s_2} + s_2^2 f_{s_1s_2s_2} + s_2^2 f_{s_2s_2s_2}\| < \infty.$

Proof: Recall the SDE system has the form

$$\begin{aligned}
dS_1(t) &= \bar{\mu}_1\big(t,\mathbf{S}(t)\big)S_1(t)dt + \bar{\sigma}_{11}\big(t,\mathbf{S}(t)\big)S_1(t)dW_1(t) + \bar{\sigma}_{12}\big(t,\mathbf{S}(t)\big)S_1(t)dW_2(t),\\
dS_2(t) &= \mu_2 S_2(t)dt + \sigma_2\rho S_2(t)dW_1(t) + \sigma_2\sqrt{1-\rho^2}S_2(t)dW_2(t),\\
dD(t) &= -rD(t)dt.
\end{aligned}$$

and the SDE system under risk neutral measure is:

$$\begin{aligned}
dS_1(t) &= rS_1(t)dt + \bar{\sigma}_{11}\big(t, \mathbf{S}(t)\big)S_1(t)d\widetilde{W}_1(t) + \bar{\sigma}_{12}\big(t, \mathbf{S}(t)\big)S_1(t)d\widetilde{W}_2(t),\\
dS_2(t) &= rS_2(t)dt + \sigma_2\rho S_2(t)d\widetilde{W}_1(t) + \sigma_2\sqrt{1-\rho^2}S_2(t)d\widetilde{W}_2(t),\\
dD(t) &= -rD(t)dt.
\end{aligned}$$

where

$$\begin{aligned}
\bar{\sigma}_{11}(t, s_1, s_2) &= \frac{\sigma_1}{1-\lambda(t, s_1, s_2)\frac{\partial f}{\partial s_1}},\\
\bar{\sigma}_{12}(t, s_1, s_2) &= \frac{\sigma_2\frac{s_2}{s_1}\lambda(t, s_1, s_2)\frac{\partial f}{\partial s_2}}{1-\lambda(t, s_1, s_2)\frac{\partial f}{\partial s_1}}.\\
\bar{\mu}_1(t, s_1, s_2) &= \frac{1}{1-\lambda(t, s_1, s_2)\frac{\partial f}{\partial s_1}}\Big(\mu_1 + \frac{1}{s_1}\lambda(t, s_1, s_2)\frac{\partial f}{\partial t} + \mu_2\frac{s_2}{s_1}\lambda(t, s_1, s_2)\frac{\partial f}{\partial s_2}\\
&+ \frac{1}{2}\frac{\partial^2 f}{\partial s_1^2}\frac{1}{\big(1-\lambda(t, s_1, s_2)\frac{\partial f}{\partial s_1}\big)^2}\big(\sigma_1^2 s_1 + \sigma_2^2\frac{s_2}{s_1}\lambda^2(t, s_1, s_2)\big(\frac{\partial f}{\partial s_2}\big)^2 + 2\rho\sigma_1\sigma_2 s_2\lambda(t, s_1, s_2)\frac{\partial f}{\partial s_2}\big)\\
&+ \frac{\partial^2 f}{\partial s_1\partial s_2}\frac{1}{1-\lambda(t, s_1, s_2)\frac{\partial f}{\partial s_1}}\big(\rho\sigma_1\sigma_2 s_2 + \sigma_2^2\frac{s_2^2}{s_1}\lambda(t, s_1, s_2)\frac{\partial f}{\partial s_2}\big) + \frac{1}{2}\sigma_2^2\frac{s_2^2}{s_1}\frac{\partial^2 f}{\partial s_2^2}\Big).
\end{aligned}$$

To show $s_1\bar{\mu}_1(t, s_1, s_2)$, $s_1\bar{\sigma}_{11}(t, s_1, s_2)$ and $s_1\bar{\sigma}_{12}(t, s_1, s_2)$ are uniformly Lipschitz continuous in (s_1, s_2), it is sufficient to show their respective partial derivatives are bounded. The derivatives of $s_1\bar{\sigma}_{11}(t, s_1, s_2)$ and $s_1\bar{\sigma}_{12}(t, s_1, s_2)$ are:

$$\begin{aligned}
\big[\bar{\sigma}_{11}s_1\big]_{s_1} &= \sigma_1\big(\frac{1}{1-\lambda f_{s_1}} + \frac{s_1(\lambda_{s_1}f_{s_1} + \lambda f_{s_1 s_1})}{(1-\lambda f_{s_1})^2}\big),\\
\big[\bar{\sigma}_{11}s_1\big]_{s_2} &= \sigma_1 s_1\frac{\lambda_{s_2}f_{s_1} + \lambda f_{s_1 s_2}}{(1-\lambda f_{s_1})^2},\\
\big[\bar{\sigma}_{12}s_1\big]_{s_1} &= \sigma_2 s_2\big(\frac{(\lambda_{s_1}f_{s_2} + \lambda f_{s_1 s_2})}{1-\lambda f_{s_1}} + \frac{\lambda f_{s_2}(\lambda f_{s_1 s_1} + \lambda_{s_1}f_{s_1})}{(1-\lambda f_{s_1})^2}\big),\\
\big[\bar{\sigma}_{12}s_1\big]_{s_2} &= \sigma_2\big(\frac{\lambda f_{s_2} + s_2(\lambda_{s_2}f_{s_2 s_2} + \lambda f_{s_2 s_2})}{1-\lambda f_{s_1}} + \lambda\frac{s_2 f_{s_2}(\lambda_{s_2}f_{s_1} + \lambda f_{s_1 s_2})}{(1-\lambda f_{s_1})^2}\big).
\end{aligned}$$

We can clearly see the boundedness requirement for $\big[\bar{\sigma}_{11}\big]_{s_1}$, $\big[\bar{\sigma}_{12}\big]_{s_1}$, $\big[\bar{\sigma}_{11}\big]_{s_2}$ and $\big[\bar{\sigma}_{12}\big]_{s_2}$ can be condensed into:

$$\|\lambda(s_1 f_{s_1 s_1} + s_1 f_{s_1 s_2} + f_{s_2} + s_2 f_{s_2} + s_2 f_{s_1 s_1} + s_2 f_{s_1 s_2} + s_2 f_{s_2 s_2})\| < \infty, \tag{B.1.6}$$

$$\|\big(\lambda_{s_1} + \lambda_{s_2}\big)\big(s_1 f_{s_1} + s_2 f_{s_1} + s_2 f_{s_2}\big)\| < \infty. \tag{B.1.7}$$

Furthermore, we will require the denominator terms in the partial derivatives above to satisfy:

$$|||1-\lambda f_{s_1}||| > \delta_0,\ \text{for some}\ \delta_0 > 0. \tag{B.1.8}$$

The partial derivative of $s_1\bar{\mu}_1(t, s_1, s_2)$ with respect to s_1 is:

$$
\begin{aligned}
\left[\bar{\mu}_1 s_1\right]_{s_1} =& \mu_1\Big(\frac{1}{1-\lambda f_{s_1}} + \frac{s_1(\lambda_{s_1} f_{s_1} + \lambda f_{s_1 s_1})}{(1-\lambda f_{s_1})^2}\Big) + \frac{\lambda f_{t s_1} + \lambda_{s_1} f_t}{1-\lambda f_{s_1}} + \frac{\lambda f_t(\lambda_{s_1} f_{s_1} + \lambda f_{s_1 s_1})}{(1-\lambda f_{s_1})^2} \\
&+\mu_2 s_2\Big(\frac{\lambda f_{s_1 s_2} + \lambda_{s_1} f_{s_2}}{1-\lambda f_{s_1}} + \frac{\lambda f_{s_2}(\lambda_{s_1} f_{s_1} + \lambda f_{s_1 s_1})}{(1-\lambda f_{s_1})^2}\Big) \\
&+\frac{1}{2}\sigma_1^2\Big(\frac{s_1^2 f_{s_1 s_1 s_1} + 2 s_1 f_{s_1 s_1}}{(1-\lambda f_{s_1})^3} + 3\frac{s_1^2 f_{s_1 s_1}(\lambda_{s_1} f_{s_1} + \lambda f_{s_1 s_1})}{(1-\lambda f_{s_1})^4}\Big) \\
&+\frac{1}{2}\sigma_2^2 s_2^2\Big(\frac{\lambda^2 f_{s_2}^2 f_{s_1 s_1 s_1} + 2\lambda f_{s_2} f_{s_1 s_1}(\lambda_{s_1} f_{s_2} + \lambda f_{s_1 s_2})}{(1-\lambda f_{s_1})^3} \\
&+3\frac{\lambda^2 f_{s_2}^2 f_{s_1 s_1}(\lambda_{s_1} f_{s_1} + \lambda f_{s_1 s_1})}{(1-\lambda f_{s_1})^4}\Big) \\
&+\rho\sigma_1\sigma_2 s_2\Big(\frac{s_1 \lambda f_{s_2} f_{s_1 s_1 s_1} + s_1 f_{s_1 s_1}(\lambda_{s_1} f_{s_2} + \lambda f_{s_1 s_2}) + \lambda f_{s_1 s_1} f_{s_2}}{(1-\lambda f_{s_1})^3} \\
&+3\frac{s_1 \lambda f_{s_2} f_{s_1 s_1}(\lambda_{s_1} f_{s_1} + \lambda f_{s_1 s_1})}{(1-\lambda f_{s_1})^4}\Big) \\
&+\rho\sigma_1\sigma_2 s_2\Big(\frac{f_{s_1 s_2} + s_1 f_{s_1 s_1 s_2}}{(1-\lambda f_{s_1})^2} + 2\frac{s_1(\lambda_{s_1} f_{s_1} + \lambda f_{s_1 s_1})}{(1-\lambda f_{s_1})^3}\Big) \\
&+\sigma_2^2 s_2^2\Big(\frac{\lambda f_{s_2} f_{s_1 s_1 s_2} + f_{s_1 s_2}(\lambda_{s_1} f_{s_1} + \lambda f_{s_1 s_1})}{(1-\lambda f_{s_1})^2} + \frac{\lambda f_{s_2} f_{s_1 s_2}(\lambda_{s_1} f_{s_1} + \lambda f_{s_1 s_1})}{(1-\lambda f_{s_1})^3}\Big) \\
&+\frac{1}{2}\sigma_2^2 s_2^2\Big(\frac{f_{s_1 s_2 s_2}}{1-\lambda f_{s_1}} + \frac{f_{s_2 s_2}(\lambda_{s_1} f_{s_1} + \lambda f_{s_1 s_1})}{(1-\lambda f_{s_1})^2}\Big).
\end{aligned}
$$

The partial derivative of $s_1\bar{\mu}_1(t,s_1,s_2)$ with respect to s_2 is:

$$
\begin{aligned}
\big[\bar{\mu}_1 s_1\big]_{s_2} =& \mu_1 s_1 \frac{\lambda_{s_2} f_{s_1} + \lambda f_{s_1 s_2}}{(1-\lambda f_{s_1})^2} + \frac{\lambda_{s_2} f_t + \lambda f_{t s_2}}{1-\lambda f_{s_1}} + \frac{\lambda f_t(\lambda_{s_2} f_{s_1} + \lambda f_{s_1 s_2})}{(1-\lambda f_{s_1})^2} \\
&+\mu_2\Big(\frac{\lambda f_{s_2} + s_2(\lambda_{s_2} f_{s_2} + \lambda f_{s_1 s_2})}{1-\lambda f_{s_1}} + \frac{s_2 \lambda f_{s_2}(\lambda_{s_2} f_{s_1} + \lambda f_{s_1 s_2})}{(1-\lambda f_{s_1})^2}\Big) \\
&+\frac{1}{2}\sigma_1^2 s_1^2\Big(\frac{f_{s_1 s_1 s_2}}{(1-\lambda f_{s_1})^3} + 3\frac{f_{s_1 s_1}(\lambda_{s_2} f_{s_1} + \lambda f_{s_1 s_2})}{(1-\lambda f_{s_1})^4}\Big) \\
&+\frac{1}{2}\sigma_2^2\Big(\frac{\lambda^2 f_{s_1 s_1} f_{s_2}^2 + s_2\big(\lambda^2 + f_{s_1 s_1 s_2} f_{s_2}^2 + f_{s_1 s_1}(2\lambda\lambda_{s_2} f_{s_2}^2 + s\lambda^2 f_{s_2} f_{s_2 s_2})\big)}{(1-\lambda f_{s_1})^3} \\
&+3\frac{s_2 \lambda^2 f_{s_1 s_1} f_{s_2}^2(\lambda_{s_2} f_{s_1} + \lambda f_{s_1 s_2})}{(1-\lambda f_{s_1})^4}\Big) \\
&+\rho\sigma_1\sigma_2 s_1\Big(\frac{\lambda f_{s_2} f_{s_1}^2 + s_2\big(\lambda_{s_2} f_{s_2} f_{s_1}^2 + \lambda(f_{s_2 s_2} f_{s_1}^2 + 2 f_{s_2} f_{s_1} f_{s_1 s_2})\big)}{(1-\lambda f_{s_1})^3} \\
&+3\frac{s_2 \lambda f_{s_2} f_{s_1 s_1}(\lambda_{s_2} f_{s_1} + \lambda f_{s_1 s_2})}{(1-\lambda f_{s_1})^4}\Big) \\
&+\rho\sigma_1\sigma_2 s_1\Big(\frac{f_{s_1 s_2} + s_2 f_{s_1 s_2 s_2}}{(1-\lambda f_{s_1})^2} + \frac{s_2 f_{s_1 s_2}(\lambda_{s_2} f_{s_1} + \lambda f_{s_1 s_2})}{(1-\lambda f_{s_1})^3}\Big) \\
&+\sigma_2^2\Big(\frac{2 s_2^2 \lambda f_{s_2} f_{s_1 s_2} + s_2^2\big(\lambda_{s_2} f_{s_2} f_{s_1 s_2} + \lambda(f_{s_1 s_2} f_{s_2 s_2} + f_{s_2} f_{s_1 s_2 s_2})\big)}{(1-\lambda f_{s_1})^2} \\
&+\frac{s_2^2 \lambda f_{s_2} f_{s_1 s_2}(\lambda_{s_2} f_{s_1} + \lambda f_{s_1 s_2})}{(1-\lambda f_{s_1})^3}\Big) \\
&+\frac{1}{2}\sigma_2^2\Big(\frac{2 s_2 f_{s_2 s_2} + s_2^2 f_{s_2 s_2 s_2}}{1-\lambda f_{s_1}} + \frac{s_2^2 f_{s_2 s_2}(\lambda_{s_2} f_{s_1} + \lambda f_{s_1 s_2})}{(1-\lambda f_{s_1})^2}\Big).
\end{aligned}
$$

We conclude the partial derivatives of $\bar{\mu}_1(t,s_1,s_2)$ will be bonded when $|||1-\lambda f_{s_1}||| > \delta_0$ and:

$$
\begin{aligned}
&\|(\lambda + \lambda_{s_1} + \lambda_{s_2})(f_t + f_{s_1} + f_{s_2} + f_{s_1 s_1} + f_{s_1 s_2} + f_{s_2 s_2} + f_{s_1 s_1 s_2} \\
&\quad + f_{s_1 s_2 s_2})\| < \infty,
\end{aligned} \tag{B.1.9}
$$

$$
\|\lambda(f_{t s_1} + f_{t s_2} + f_{s_1 s_1 s_1} + f_{s_2 s_2 s_2}) + \lambda_{s_1} f_{s_1 s_1 s_1} + \lambda_{s_2} f_{s_2 s_2 s_2}\| < \infty, \tag{B.1.10}
$$

$$
\begin{aligned}
&\|s_1 f_{s_1 s_1} + s_1 f_{s_1 s_2} + s_2 f_{s_1 s_2} + s_2 f_{s_2 s_2} + s_1^2 f_{s_1 s_1 s_2} + s_1^2 f_{s_1 s_2 s_2} + s_1 s_2 f_{s_1 s_1 s_2} \\
&\quad + s_1 s_2 f_{s_1 s_2 s_2} + s_2^2 f_{s_1 s_2 s_2} + s_2^2 f_{s_2 s_2 s_2}\| < \infty,
\end{aligned} \tag{B.1.11}
$$

The combination of requirements (B.1.6), (B.1.7), (B.1.8), (B.1.9), (B.1.10), (B.1.11) will guarantee $s_1\bar{\mu}_1(t,s_1,s_2)$, $s_1\bar{\sigma}_{11}(t,s_1,s_2)$ and $s_1\bar{\sigma}_{12}(t,s_1,s_2)$ are uniformly Lipschitz continuous in $\big(\mathbb{R}^+\big)^2$.

To show the linear growth condition of 3.1, take $\tilde{\mathbf{s}} = \mathbf{0}$, then from the Lipschitz condition we just showed, we have

$$
|\boldsymbol{\mu}(t,\mathbf{s}) - \boldsymbol{\mu}(t,\mathbf{0})| + |\boldsymbol{\sigma}(t,\mathbf{s}) - \boldsymbol{\sigma}(t,\mathbf{0})| \leqslant c|\mathbf{s}|, \quad \forall \mathbf{s} \in (\mathbb{R}^+)^2, t \in [0,T].
$$

By the triangle inequality we have

$$|\boldsymbol{\mu}(t,\mathbf{s})| - |\boldsymbol{\mu}(t,\mathbf{0})| + |\boldsymbol{\sigma}(t,\mathbf{s})| - |\boldsymbol{\sigma}(t,\mathbf{0})| \leqslant |\boldsymbol{\mu}(t,\mathbf{s}) - \boldsymbol{\mu}(t,\mathbf{0})| + |\boldsymbol{\sigma}(t,\mathbf{s}) - \boldsymbol{\sigma}(t,\mathbf{0})| \leqslant c|\mathbf{s}|,$$

but $|\boldsymbol{\mu}(t,\mathbf{0})|$ and $|\boldsymbol{\sigma}(t,\mathbf{0})|$ are both zero, then we have the expression

$$|\boldsymbol{\mu}(t,\mathbf{s})| + |\boldsymbol{\sigma}(t,\mathbf{s})| \leqslant c|\mathbf{s}|,$$

which is a stronger condition than the linear growth condition. We can conclude when the uniform Lipschitz condition hold, linear growth condition will hold as well. By using Theorem 3.1, the SDE system will emit a unique solution in $\mathbb{P}$ sense.

The combination of requirement B.1.6, B.1.7 and B.1.8 will only guarantee the diffusion functions to be uniform Lipschitz and have the linear growth property. By using Theorem 3.1, the SDE system will emit a unique solution in $\widetilde{\mathbb{P}}$ sense.

This completes the proof. $\square$

B.2 Finite Liquidity Existence and Uniqueness Theorem II

To show the SDE have a unique strong solution, it is sufficient to show that the conditions (1) − (6) in Appendix Section B.1 are satisfied for the particular choice of $\lambda(t, s_1)$ and $f(t, s_1, s_2) = \Delta_1(t)$.

- Condition (1):

$$\begin{aligned}
&\|\lambda(s_1 Spd_{111} + s_1 Spd_{112} + \Gamma_{12} + s_2\Gamma_{12} + s_2 Spd_{111} + s_2 Spd_{112} + s_2 Spd_{122})\| \\
&= \|\lambda\Big(\frac{N'(d_+)}{\sigma s_1\sqrt{\tau}}\big(\frac{2d_+}{\sigma s_1\sqrt{\tau}} + 1\big) + s_1\frac{2d_+ N'(d_+)}{\sigma^2\tau s_1 s_2} + \frac{1}{\sigma\sqrt{\tau}}\frac{s_2 N'(d_+)}{s_1^2} + \frac{s_2^2 N'(d_+)}{\sigma\sqrt{\tau}s_1^2} \\
&+ \frac{s_2 N'(d_+)}{\sigma s_1^2\sqrt{\tau}}\big(\frac{2d_+}{\sigma s_1\sqrt{\tau}} + 1\big) + \frac{2d_+ N'(d_+)}{\sigma^2\tau s_1} + \frac{2d_- s_2 N'(d_+)}{\sigma^2\tau s_1^2}\Big)\| < \infty.
\end{aligned}$$

Proof: Notice there is a common term of the form $\frac{N'(d_+)}{s_1^n}$. These terms appears naturally in higher order Greeks. Consider any real number n, we have:

$$\begin{aligned}
\frac{N'(d_+)}{s_1^n} &= \frac{1}{s_1^n\sqrt{2\pi}}\exp\Big\{-\Big(\frac{\log(\frac{s_1}{s_2}) + \frac{1}{2}\sigma^2\tau}{\sigma\sqrt{\tau}}\Big)^2\Big\} \\
&= \frac{1}{s_1^n\sqrt{2\pi}}e^{\Big\{-\frac{\log^2(s_1)+\log(s_1)\left(\frac{1}{2}\sigma^2\tau-\log(s_2)\right)+\left(\frac{1}{2}\sigma^2\tau-\log(s_2)\right)^2}{\sigma^2\tau}\Big\}}e^{-n\log(s_1)} \\
&= \frac{1}{\sqrt{2\pi}}\exp\Big\{-\frac{\log^2(s_1) + o\big(\log(s_1)\big)}{\sigma^2\tau}\Big\},
\end{aligned}$$

which approaches to 0 as s_1 approaches to zero, and approaches to 0 as well as s_1 approaches ∞. Since n was arbitrary, then all of the functions in Condition (1) are bounded in s_1. With a similar method involving the common term $\frac{N'(d_+)}{s_2^n}$, we can also show that all of the terms in Condition (1) are bounded in s_2. We can ultimately conclude that the entire function of Condition (1) is bounded in (s_1, s_2). □

- Condition (2):

$$\|\lambda_{s_1}\big(s_1\Gamma_{11} + s_2\Gamma_{11} + s_2\Gamma_{12}\big)\| < \infty.$$

Proof: Same proof as Condition (1). □

- Condition (3):

$$|||1 - \lambda\Gamma_{11}||| > \delta_0, \text{ for some } \delta_0 > 0.$$

Proof: This condition already holds in the s_1, s_2 dimension. For t we have $\lim_{t\to T}\bar{\lambda}(t,s_1) = 0$ and $\lim_{t\to T}\Gamma_{11}(t) = \infty$ for at the money options. Since $\bar{\lambda}(t,s_1)$ approach to 0 at a greater rate, then $\lim_{t\to T}\bar{\lambda}(t,s_1)\Gamma_{11}(t) = 0$. In fact, this ensures the $\bar{\lambda}(t,s_1)\Gamma_{11}(t)$ term stays small, which ultimately guarantees the existence of δ_0. There is a more detailed explanation in Pirvu et al (2014) [1]. □

- Condition (4):

$$\|(\lambda + \lambda_{s_1})(Chm_1 + \Gamma_{11} + \Gamma_{12} + Spd_{111} + Spd_{112} + Spd_{122} + Acc_{1112} + Acc_{1122})\| < \infty.$$

Proof: Same proof as Condition (1). □

- Condition (5):

$$\|\lambda(Col_1 + Col_2 + Acc_{1111} + Acc_{1222}) + \lambda_{s_1}Acc_{1111} + \lambda_{s_2}Acc_{1222}\| < \infty.$$

Proof: Same proof as Condition (1). □

- Condition (6):

$$\|s_1Spd_{111} + s_1Spd_{112} + s_2Spd_{112} + s_2Spd_{122} + s_1^2Acc_{1112} + s_1^2Acc_{1122} + s_1s_2Acc_{1112} + s_1s_2Acc_{1122} + s_2^2Acc_{1122} + s_2^2Acc_{1222}\| < \infty.$$

Proof: Same proof as Condition (1). □

Since we have shown Condition (1) to (6) in the Appendix Section B.2 holds for our price impact trading strategy $\lambda\big(t, S_1(t)\big)df\big(t, S_1(t), S_2(t)\big)$. We can conclude the system of SDE in (4.1.1) has a unique solution.

B.3 Finite Liquidity Existence and Uniqueness Theorem III

Proof: To show the SDE emit weak solutions, we need to show the conditions (1)-(3) in Theorem 3.3 are satisfied for the particular choice of $\lambda(t, s_1, s_2)$ and $f(t, s_1, s_2)$ of Spread Option. Recall from (5.1.2) that all the Greeks are just linear combinations of the form:

To achieve this, first recall from (2.2.6) that all the Greeks are just linear combinations of the form:

$$\begin{aligned}\overline{\mathbf{Greek}}(t, s_1, s_2) &= \int\int_{\mathbb{R}^2+i\epsilon} f_{\otimes}(\mathbf{u}) e^{i\mathbf{u}'\mathbf{x}} \Phi(\mathbf{u}, \tau)\hat{P}(\mathbf{u})d\mathbf{u} \\ &= e^{-\epsilon'\mathbf{x}} \int\int_{\mathbb{R}^2} f_{\otimes}(\mathbf{u}+i\epsilon) e^{i\Re(\mathbf{u})'\mathbf{x}} \Phi(\mathbf{u}+i\epsilon, \tau)\hat{P}(\mathbf{u}+i\epsilon)d\mathbf{u} \\ &= \frac{1}{s_1^{\epsilon_1} s_2^{\epsilon_2}} \int\int_{\mathbb{R}^2} f_{\otimes}(\mathbf{u}+i\epsilon) e^{i\Re(\mathbf{u})'\mathbf{x}} \Phi(\mathbf{u}+i\epsilon, \tau)\hat{P}(\mathbf{u}+i\epsilon)d\mathbf{u} \\ &= \frac{1}{s_1^{\epsilon_1} s_2^{\epsilon_2}} \overline{\mathbf{Greek}}^{\Re}(t, s_1, s_2).\end{aligned}$$

Here we use $\overline{\mathbf{Greek}}^{\Re}(t, s_1, s_2)$ to distinguish between contour and real integrals forms. The term $e^{i\Re(\mathbf{u})'\mathbf{x}}$ lays on the complex unit circle, this results in $\|\overline{\mathbf{Greek}}^{\Re}(t, s_1, s_2)\| < \infty$ for all Greeks. Then proving the regularity conditions only boils down to the terms $\frac{1}{s_1^{\epsilon_1} s_2^{\epsilon_2}}$.

When we rewrite the counter integral as real integrals and substitute the BS Spread Greeks into Condition (1), we get:

$$\begin{aligned}&\lambda(t, s_1, s_2)\Big(\frac{ke^{-r\tau}}{(2\pi)^2}\Big)\Big(\frac{s_1+s_2}{s_1^{3+\epsilon_1} s_2^{\epsilon_2}}(2\bar{\Delta}_1^{\Re} + \bar{\Gamma}_{11}^{\Re} - \overline{Spd}_{111}^{\Re}) - \frac{s_1+s_2}{s_1^{2+\epsilon_1} s_2^{1+\epsilon_2}}\overline{Spd}_{112}^{\Re} \\ &- \frac{1+s_2}{s_1^{1+\epsilon_1} s_2^{1+\epsilon_2}}\bar{\Gamma}_{12}^{\Re} - \frac{1+s_2}{s_1^{1+\epsilon_1} s_2^{1+\epsilon_2}}\overline{Spd}_{122}^{\Re}\Big).\end{aligned}$$

By dropping the constants and bounded real integral terms, we have

$$\begin{aligned}&\lambda(t, s_1, s_2)\Big(\frac{s_1+s_2}{s_1^{3+\epsilon_1} s_2^{\epsilon_2}} - \frac{s_1+s_2}{s_1^{2+\epsilon_1} s_2^{1+\epsilon_2}} - \frac{1+s_2}{s_1^{1+\epsilon_1} s_2^{1+\epsilon_2}} - \frac{1+s_2}{s_1^{1+\epsilon_1} s_2^{1+\epsilon_2}}\Big) \\ &= \lambda(t, s_1, s_2)\Big(\frac{s_2^2 - 3s_1^2 - 2s_1^2 s_2}{s_1^{3+\epsilon_1} s_2^{1+\epsilon_2}}\Big).\end{aligned} \tag{B.3.1}$$

Since $\lambda(t, s_1, s_2)$ is only non-zero between $\underline{S}$ and $\overline{S}$, we conclude Expression (B.3.1) is bounded.

Substitute the BS Spread Greeks into Condition (2) and adopting real integrals, we get:

$$\big(\lambda_{s_1} + \lambda_{s_2}\big)\Big(\frac{ke^{-r\tau}}{(2\pi)^2}\Big)\Big(\frac{s_1+s_2}{s_1^{2+\epsilon_1} s_2^{\epsilon_2}}(\bar{\Delta}_1^{\Re} + \bar{\Gamma}_{11}^{\Re}) + \frac{s_2}{s_1^{1+\epsilon_1} s_2^{1+\epsilon_2}}\bar{\Gamma}_{12}^{\Re}\Big).$$

By dropping the constants and bounded real integral terms, we have

$$\big(\lambda_{s_1} + \lambda_{s_2}\big)\big(\frac{2s_1 s_2 + s_2^2}{s_1^{2+\epsilon_1} s_2^{1+\epsilon_2}}\big). \tag{B.3.2}$$

Since $\lambda(t, s_1, s_2)$ is only non-zero between $\underline{S}$ and $\overline{S}$, then λ_{s_1} and λ_{s_2} are also only non-zero in the same range. We can conclude Expression (B.3.2) is bounded.

Condition (3) is bounded in s_1 and s_2 by the same logic as Condition (1) and (2). For the t dimension of Condition (3), $\Gamma_{11}(t, s_1, s_2)$ diverges as $t \to T$. By design $\lambda(t, s_1, s_2)$ has a higher order decaying that ensures $\lambda(t, s_1, s_2)\Gamma_{11}(t, s_1, s_2)$ to stay finite. Therefore we can always find a $\delta_0 > 0$ for which Condtion (3) holds.

Since we have showed Condition (1) to (3) in Theorem 3.3 holds for our choice of $\lambda(t, s_1, s_2)$ and $f(t, s_1, s_2)$ in this chapter. We can conclude the system of SDEs (4.1.3) will emit a unique solution. $\square$

Appendix C

Identities and Operations

C.1 Two-Dimensional Parseval's Identity

Let $f(\mathbf{x})$ and $g(\mathbf{x})$ be two-dimensional real functions, then

$$\int\int_{\mathbb{R}^2} f(\mathbf{x})g(\mathbf{x})d\mathbf{x} = \frac{1}{(2\pi)^2}\int\int_{\mathbb{R}^2} \mathcal{F}\big[f(\mathbf{x})\big](\mathbf{u})\overline{\mathcal{F}\big[g(\mathbf{x})\big](\mathbf{u})}d\mathbf{u},$$

where $\mathcal{F}\big[\,\cdot\,\big]$ is the Fourier Transform.

C.2 Tensor Operations

C.2.1 Tensor Product

For vectors $\mathbf{a}$ and $\mathbf{b}$, the tensor product is defined to be:

$$\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \otimes \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix},$$

the result is a matrix.

For matrices $\mathbf{A}$ and $\mathbf{B}$, the tensor product is defined to as:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \otimes \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \left[\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}\right],$$

the result is a 3-dimensional tensor.

For two 3-dimensional tensor $\mathbf{T}_A^{(3)}$ and $\mathbf{T}_B^{(3)}$, the tensor product is defined to as:

$$\begin{aligned}\mathbf{T}_A^{(3)} \otimes \mathbf{T}_B^{(3)} &= \left[\begin{bmatrix} a_{111} & a_{121} \\ a_{211} & a_{221} \end{bmatrix}, \begin{bmatrix} a_{112} & a_{122} \\ a_{212} & a_{222} \end{bmatrix}\right] \otimes \left[\begin{bmatrix} b_{111} & b_{121} \\ b_{211} & b_{221} \end{bmatrix}, \begin{bmatrix} b_{112} & b_{122} \\ b_{212} & b_{222} \end{bmatrix}\right] \\ &= \left\{\left[\begin{bmatrix} a_{111} & a_{121} \\ a_{211} & a_{221} \end{bmatrix}, \begin{bmatrix} a_{112} & a_{122} \\ a_{212} & a_{222} \end{bmatrix}\right], \left[\begin{bmatrix} b_{111} & b_{121} \\ b_{211} & b_{221} \end{bmatrix}, \begin{bmatrix} b_{112} & b_{122} \\ b_{212} & b_{222} \end{bmatrix}\right]\right\},\end{aligned}$$

the result is a 4-dimensional tensor.

C.2.2 Tensor Multiplication

Let $\mathcal{T}_A^{(4)}$ be a 3-dimensional tensor and $\mathbf{b}$ be vector, then define the multiplication operation to be:

$$\mathcal{T}_A^{(3)} \times \mathbf{b} = \left[\left[\begin{bmatrix} a_{1111} & a_{1211} \\ a_{2111} & a_{2211} \end{bmatrix}, \begin{bmatrix} a_{1112} & a_{1212} \\ a_{2112} & a_{2212} \end{bmatrix}\right], \left[\begin{bmatrix} a_{1121} & a_{1221} \\ a_{2121} & a_{2221} \end{bmatrix}, \begin{bmatrix} a_{1122} & a_{1222} \\ a_{2122} & a_{2222} \end{bmatrix}\right]\right] \times \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$
$$= \left[\begin{bmatrix} a_{1111}b_1 + a_{1211}b_2 & a_{1112}b_1 + a_{1212}b_2 \\ a_{2111}b_1 + a_{2211}b_2 & a_{2112}b_1 + a_{2212}b_1 \end{bmatrix}, \begin{bmatrix} a_{1121}b_1 + a_{1221}b_2 & a_{1122}b_1 + a_{1222}b_2 \\ a_{2121}b_1 + a_{2221}b_2 & a_{2122}b_1 + a_{2222}b_2 \end{bmatrix}\right]$$

Let $\mathcal{T}_A^{(3)}$ be a 3-dimensional tensor and $\mathbf{B}$ be matrix, then define the multiplication operation to be:

$$\mathcal{T}_A^{(3)} \times \mathbf{B} = \left[\begin{bmatrix} a_{111} & a_{121} \\ a_{211} & a_{221} \end{bmatrix}, \begin{bmatrix} a_{112} & a_{122} \\ a_{212} & a_{222} \end{bmatrix}\right] \times \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$
$$= \left[\begin{bmatrix} a_{111}b_{11} + a_{121}b_{21} & a_{111}b_{12} + a_{121}b_{22} \\ a_{211}b_{11} + a_{221}b_{21} & a_{211}b_{12} + a_{221}b_{22} \end{bmatrix}, \begin{bmatrix} a_{112}b_{11} + a_{122}b_{21} & a_{112}b_{12} + a_{122}b_{22} \\ a_{212}b_{11} + a_{222}b_{21} & a_{212}b_{12} + a_{222}b_{22} \end{bmatrix}\right]$$

C.2.3 Outer Product

For vectors $\mathbf{a}$ and $\mathbf{b}$, the outer product is defined to be:

$$\mathbf{a}\bar{\otimes}\mathbf{b} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \bar{\otimes} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} a_1b_1 & a_1b_1 \\ a_2b_1 & a_2b_2 \end{bmatrix},$$

the result is a matrix.
For matrices $\mathbf{A}$ and vector $\mathbf{b}$, the outer product can be defined as:

$$\mathbf{A}\bar{\otimes}\mathbf{b} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \bar{\otimes} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \left[\begin{bmatrix} a_{11}b_1 & a_{12}b_1 \\ a_{21}b_1 & a_{22}b_1 \end{bmatrix}, \begin{bmatrix} a_{11}b_2 & a_{12}b_2 \\ a_{21}b_2 & a_{22}b_2 \end{bmatrix}\right],$$

the result is a 3-dimensional tensor.

Bibliography

[1] A.R. Yazdanian A. Shidfar, Kh. Paryab and Traian A. Pirvu. Numerical analysis for spread option pricing model of markets with finite liquidity: first-order feedback model. *International Journal of Computer Mathematics*, 91:2603–2620, 2014.

[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[3] D. Ahmadian, O. Farkhondeh Rouz, K. Ivaz, and A. Safdari-Vaighani. Robust numerical algorithm to the european option with illiquid markets. *Applied Mathematics and Computation*, 366:124693, 2020.

[4] Ali Al-Aradi, Adolfo Correia, Danilo Naiff, Gabriel Jardim, and Yuri Saporito. Solving nonlinear and high-dimensional partial differential equations via deep learning, 2018.

[5] Elisa Alòs and Michael Coulon. On the optimal choice of strike conventions in exchange option pricing. *EconPapers*, 2018.

[6] Elisa Alòs and Thorsten Rheinländer. Pricing and hedging margrabe options with stochastic volatilities. *EconPapers*, 2017.

[7] Abraham J. Arenas, Gilberto Gonzalez-Parra, and Blas Melendez Caraballo. A nonstandard finite difference scheme for a nonlinear black-scholes equation. *Mathematical and Computer Modelling*, 57(7):1663 – 1670, 2013.

[8] Bruce P. Ayati and Todd F. Dupont. Convergence of a step-doubling galerkin method for parabolic problems, 1999.

[9] Yoshua Bengio. *Deep Learning of Representations for Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*. PMLR, 02 Jul 2012.

[10] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

[11] Rena Carmona and Valdo Durrleman. Pricing and hedging spread options. *SIAM Review*, 45(4):627–685, 2003.

[12] Jingrun Chen, Rui Du, and Keke Wu. A comprehensive study of boundary conditions when solving pdes by dnns. *preprint*, 05 2020.

[13] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[14] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.

[15] S. S. G. Dempster, M. A. H.and Hong. *Spread Option Valuation and the Fast Fourier Transform*. Springer Berlin Heidelberg, 2002.

[16] Mikhail Dyshaev and Vladimir Fedorov. The sensitivities (greeks) for some models of option pricing with market illiquidity, 02 2019.

[17] Ryan Ferguson and Andrew David Green. Applying deep learning to derivatives valuation. *SSRN Electronic Journal*, 2018.

[18] Avner Friedman. *Stochastic Differential Equations and Applications*. Academic Press, 1st edition edition, 1975.

[19] Michael Giles and Paul Glasserman. Smoking adjoints: fast evaluation of greeks in monte carlo calculations. *Risk Journals*, 2005.

[20] Michael B. Giles and Lukasz Szpruch. Antithetic multilevel monte carlo estimation for multi-dimensional sdes without lÉvy area simulation. *The Annals of Applied Probability*, 24(4):1585–1620, 2014.

[21] Michael B. Giles and Lukasz Szpruch. Multilevel monte carlo methods for applications in finance. *High-Performance Computing in Finance*, pages 197–247, 2018.

[22] Paul Glasserman. *Monte Carlo methods in financial engineering.* Springer, 2004.

[23] Kristoffer J. Glover, Peter W. Duck, and David P. Newton. On nonlinear models of markets with finite liquidity: Some cautionary notes. *SIAM Journal on Applied Mathematics*, 70(8):3252–3271, 2010.

[24] Donatien Hainaut. Calendar spread exchange options pricing with gaussian random fields. *Risks*, 6:77, 08 2018.

[25] Desmond J. Higham. An introduction to multilevel monte carlo for option valuation. *International Journal of Computer Mathematics*, 92(12):2347–2360, 2015.

[26] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359 – 366, 1989.

[27] John C. Hull. *Options, futures, and other derivatives.* Pearson Prentice Hall, 8th edition edition, 2006.

[28] John K. Hunter. Notes on partial differential equations, 2010.

[29] T. R. Hurd and Zhuowei Zhou. A fourier transform method for spread option pricing. *SIAM J. Financial Mathematics*, 1(1):142–157, 2010.

[30] Ichiro Itô. On the existence and uniqueness of solutions of stochastic integral equations of the volterra type. *Kodai Math*, 2:158–170, 1979.

[31] E. Kirk and J. Aron. Correlation in the energy markets. managing energy price risk. *Risk Books*, pages 71–78, 1995.

[32] Minqiang Li, ShiJie Deng, and Jieyun Zhou. Closed-form approximations for spread option prices and greeks. *The Journal of Derivatives*, 15, 03 2008.

[33] Hong Liu and Jiongmin Yong. Option pricing with an illiquid underlying asset market. *Journal of Economic Dynamics & Control*, 29:2125–2156, 2005.

[34] William Margrabe. The value of an option to exchange one asset for another. *Journal of Finance*, 33:177–186, 1978.

[35] Robert C Merton. Optimum consumption and portfolio rules in a continuous-time model. *Journal of Economic Theory*, 3(4):373 – 413, 1971.

[36] Grigori Noyhovich Mil'shtein. Approximate integration of stochastic differential equations. *Theory of Probability & Its Applications.*, 19:557–000, 1975.

[37] Bernt Oksendal. *Stochastic Differential Equations (3rd Ed.): An Introduction with Applications.* Springer-Verlag, Berlin, Heidelberg, 1992.

[38] Traian Pirvu and Ahmadreza Yazdanian. Numerical analysis for spread option pricing model in illiquid underlying asset market: Full feedback model. *Applied Mathematics and Information Sciences*, 10:1271–1281, 05 2015.

[39] Eckhard Platen. An introduction to numerical methods for stochastic differential equations. *Acta Numerica*, 8:197–246, 1999.

[40] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2017.

[41] Reuven Y. Rubinstein and Ruth Marcus. Efficiency of multivariate control variates in monte carlo simulation. *Operations Research*, 33(3):661–677, 1985.

[42] Hasim Sak, Andrew W. Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling, 2014.

[43] Klaus Scheicher. Complexity and effective dimension of discrete lévy areas. *Journal of Complexity*, 23(2):152–168, 2007.

[44] Steven E. Shreve. *Stochastic calculus for finance II, Continuous-time models.* Springer, New York, NY; Heidelberg, 2004.

[45] Justin Sirignano and Konstantinos Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, Dec 2018.

[46] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.

[47] Siraj ul Islam and Imtiaz Ahmad. A comparative analysis of local meshless formulation for multi-asset option models. *Engineering Analysis with Boundary Elements*, 65:159 – 176, 2016.

[48] Karl Weiss, Taghi Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3, 12 2016.

[49] Paul Wilmott and Philipp J. Schönbucher. The feedback effect of hedging in illiquid markets. *SIAM Journal on Applied Mathematics*, 61(1):232–272, 2000.