

PRICING AND HEDGING ASIAN OPTIONS UNDER LÉVY
PROCESSES AND ROBUST LONG-TERM INVESTING
WITH LEARNING ABOUT STOCK RETURNS

by

Andrew Na

B.Eng. Ryerson University, 2016

A thesis

presented to Ryerson University

in partial fulfillment
of the requirements for the degree of
Master of Science
in the program of
Applied Mathematics

Toronto, Ontario, Canada, 2018

© Andrew Na, 2018

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public

**Pricing and Hedging Asian Options Under Lévy Processes and Robust
Long-Term Investing with Learning about Stock Return**

Master of Science, 2018

Andrew Na

Applied Mathematics

Ryerson University

ABSTRACT

In this work we propose a parametric model using the techniques of time-changed subordination that captures the implied volatility smile. We demonstrate that the Fourier-Cosine method can be used in a semi-static way to hedge for quadratic, VaR and AVaR risk. We also observe that investors looking to hedge VaR can simply hold the amount in a portfolio of mostly cash, whereas an investor hedging AVaR will need to hold more risky assets. We also extend ES risk to a robust framework. A conditional calibration method to calibrate the bivariate model is proposed.

For a robust long-term investor who maximizes their recursive utility and learns about the stock returns, as the willingness to substitute over time increases, the equity demand decreases and consumption-wealth ratio increases. As the preference for robustness increases the demand for risk decreases. For a positive correlation, we observe that learning about returns encourages the investor to short the bond at all levels of ψ and vice versa.

ACKNOWLEDGMENTS

This thesis concludes my research as a master's student at Ryerson University, from September 2016 to August 2018. The success of my candidacy was largely due to the continuous support of both my supervisors Dr. Pablo Olivares and Dr. Alexey Rubtsov. As well as Dr. Cornelius Oosterlee and Dr. Bowen Zhang, who, without reserve, were generous in their help. This work was possible thanks to the support of NSERC, the government of Canada, the government of Ontario and Ryerson University.

DEDICATION

The conclusion of this work marks the completion of another chapter in my life. The past two years went by in a blink of an eye, but at the same time it was filled with great memories at an institution that is always willing to give you a chance. This work has been at times challenging, at frustrating and brain wracking but most of all fulfilling. It has been the highlight of my life for the last two years and it feels gratifying to see it finished.

First and foremost, I would like to express my utmost gratitude to Dr. Pablo Olivares and Dr. Alexey Rubtsov. I have learned so many lessons from them both that I will carry forward as a researcher, academic and most importantly as a person. I would like to thank all of my instructors who have put tireless hours into our education and really helped me grow as a mathematician; Dr. Sebastian Ferrando, Dr. Jean-Paul Pascal, Dr. Silvana Ilie, Dr. Katrin Rohlf, Dr. Foivos Xanthos, and Dr. Niushan Gao. I would also like to express my gratitude to Steve Kanellis and Teresa Lee who worked tirelessly to support me and my colleagues and I would like to thank Ms. Kotlarenko and Ms. Evans who instilled in me a love for Mathematics. Lastly, I would like to express my gratitude to my friends Shannen and John for their encouragement and council and my family for their support and understanding.

TABLE OF CONTENTS

<i>Declaration</i>	ii
<i>Abstract</i>	iii
<i>Acknowledgments</i>	iv
<i>Dedication</i>	v
<i>List of Tables</i>	ix
<i>List of Figures</i>	xi
<i>List of Appendicies</i>	xii
1 Introduction	1
2 Background	5
2.1 Preliminaries	5
2.2 Arbitrage-free Pricing	7
2.3 Lévy Processes	8
2.3.1 Markov Property of Lévy Processes	9
2.3.2 Characteristic Function and Cumulants	9
2.4 Representation	10
2.4.1 Exponential Lévy Processes	13
2.4.2 Girsanov Transform	14
2.4.3 Feynman-Kac Formula	15
2.5 Time-Changed Lévy Process	16
2.5.1 Multivariate Subordination	16
3 A Time-Changed Exponential Lévy Model	18
3.1 Introduction	18
3.2 Model Setting and General Model	19
3.2.1 Characteristic Function and Lévy Triplets of the Model	19
3.3 Univariate Model	21

3.3.1	NIG Case	21
3.3.2	Variance Gamma(VG) Processes	23
3.4	Bivariate Model	24
3.4.1	NIG Case	24
3.4.2	VG Case	26
4	Asian Option Pricing	28
4.1	Introduction	28
4.2	Overview of Asian Options	29
4.2.1	Asian Basket Spread Options	29
4.3	Monte Carlo Pricing	30
4.3.1	Example of MC Simulation of Geometric Brownian Motion	30
4.4	Fourier-Cosine (COS) Method	31
4.4.1	Asian Fourier-Cosine (ASCOS) Method	32
4.5	Simulations	33
4.5.1	Univariate Simulations Results	35
4.5.2	Bivariate Simulations Results	42
5	Semi-Static Hedging Using Fourier Cosine Expansion	44
5.1	Introduction	44
5.2	Quadratic Hedging	45
5.2.1	Simulation of Quadratic Hedging	45
5.3	Quantile Hedging	49
5.3.1	Simulation of Quantile Hedging	53
5.3.2	Robust Risk Hedging	55
6	Calibration	57
6.1	Introduction	57
6.2	Model Parameter Calibration	58
6.3	Implied Volatility	62
7	Robust Consumption and Portfolio Choice with Stochastic Interest Rates and Learning about Stock Predictability	65
7.1	Introduction	65
7.2	Background	67
7.2.1	Expected Utility	67

7.2.2	Risk Aversion	68
7.2.3	Filtering	68
7.2.4	Overview of Stochastic Control	69
7.2.5	Dynamic Programming	70
7.3	Problem Formulation	71
7.4	Simulation and Results	81
8	Conclusion	88
	Appendices	91
	Bibliography	144

LIST OF TABLES

3.1	Table of Cummulants for Univariate IG Subordinated Lévy Process	22
3.2	Table of Cummulants for Univariate Gamma Subordinated Lévy Process	24
3.3	Table of Cummulants for Bivariate IG Subordinated Lévy Process	25
3.4	Table of Cummulants for Bivariate Gamma Subordinated Lévy Process	27
4.1	Summary of Monte Carlo pricing of Asian options by IG subordinated processes, $K = \$52, S_0 = \$57, dt = 1/250, T = 1/12$	40
4.2	Summary of Monte Carlo pricing of Asian options by Gamma subordinated pro- cesses, $K = \$52, S_0 = \$57, dt = 1/250, T = 1/12$	41
5.1	Summary of expected quadratic hedging losses, $S_0 = 57, K = 52, a = 1/250\kappa,$ $b = 1/\kappa, \mu = 0, \sigma = 0.02, \kappa = 0.1, n = 100000$	47
5.2	Optimal portfolios with quadratic hedging loss, $n = 100000, K = \$52, S_0 = \$57,$ $dt = 1/250, \kappa = 0.1$	48
5.3	Optimal portfolios with VaR hedging error, $n = 100000, K = \$52, S_0 = \$57,$ $dt = 1/250, \kappa = 0.1, \sigma = 0.02, \mu = 0, r = 0.036$	54
5.4	Optimal portfolios with AVaR hedging error, $n = 100000, K = \$52, S_0 = \$57,$ $dt = 1/250, \kappa = 0.1, \sigma = 0.02, \mu = 0, r = 0.036$	54
5.5	Expected hedging error, $\mathbb{E}^{\mathbb{Q}}[X_T - H_T]$, comparison between MC and COS, $n =$ $100000, K = \$52, S_0 = \$57, dt = 1/250, \kappa = 0.1, \sigma = 0.02, \mu = 0, r = 0.036$	54
6.1	Calibrated Parameters for NIG Process, $n = 100000, K^1 = K^2 = \$52, S_0^1 = \$68.58,$ $S_0^2 = \$74.88, dt = 1/12$	61

LIST OF FIGURES

4.1	The log return processes generated by (a) IG subordinated BM (b) Gamma subordinated BM and (c) GBM, using parameters $S_0 = 57$, $K = 52$, $a = 1/250\kappa$, $b = 1/\kappa$, $\mu = 0$, $\sigma = 0.02$, $\kappa = 0.02$. (d) is the historic log return of observed WTI Crude prices from 2014.	35
4.2	The QQ-plot from (a) IG subordinated process quantile, (b) Gamma subordinated process quantile, and (c) GBM quantile matched to the quantile of historic log return of observed WTI Crude prices from 2014 from sampled numbers.	36
4.3	The distribution curves of (a) IG subordinated process, (b) Gamma subordinated process, (c) GBM and (d) the historic log return of observed WTI Crude prices from 2014 from sampled numbers.	37
4.4	Monte Carlo pricing using parameters $S_0 = 57$, $a = 1/250\kappa$, $b = 1/\kappa$, $\mu = 0$, $\sigma = 0.02$, $\kappa = 0.02$	38
4.5	ASCOS pricing using parameters $S_0 = 57$, $a = 1/250\kappa$, $b = 1/\kappa$, $\mu = 0$, $\sigma = 0.02$, $\kappa = 0.02$	39
4.6	Monte Carlo pricing with confidence interval at $\alpha = 0.95$ shows the convergence of the option prices by the law of large numbers.	40
4.7	Simulation of dependent subordinator for $c = \{0, 0.5, 1\}$ values	42
4.8	Monte Carlo pricing using the bivariate model with parameters from calibration, the asset weights were set as $w_1 = 0.8$ and $w_2 = 0.2$	43
5.1	PDF of Monte Carlo simulation of quadratic hedging loss from (a) IG subordinated BM and (b) Gamma subordinated BM, using parameters $S_0 = 57$, $K = 52$, $a = 1/250\kappa$, $b = 1/\kappa$, $\mu = 0$, $\sigma = 0.02$, $\kappa = 0.1$, $n = 100000$	46
5.2	CDF of Monte Carlo simulation of quadratic hedging loss from (a) IG subordinated BM and (b) Gamma subordinated BM, using parameters $S_0 = 57$, $K = 52$, $a = 1/250\kappa$, $b = 1/\kappa$, $\mu = 0$, $\sigma = 0.02$, $\kappa = 0.1$, $n = 100000$	48

5.3	CDF of Monte Carlo simulation of (a) VaR and AVaR hedging error for NIG process and (b) VaR and AVaR Hedging error for VG process, using parameters $n = 100000$, $K = \$52$, $S_0 = \$57$, $dt = 1/250$, $\kappa = 0.1$, $\sigma = 0.02$, $\mu = 0$, $r = 0.036$	53
6.1	Current Option market prices of (a) WTI Asian call options (b) WTI Asian put options (c) Brent crude European call options. The market data was collected from Bloomberg's database via the Bloomberg Terminal	57
6.2	(a) Calibrated univariate model matched to current market price of WTI Asian Call Option (b) Calibrated bivariate model matched to current market price of WTI Asian Call Option and (c) Calibrated conditionally univariate model to current market price of Brent European Call Option expiring in 1 year.	62
6.3	Current market sensitivity of volatility in WTI Asian options to strike price.	63
6.4	Implied volatility smile of the model for different strike prices, the parameters at each strike price, K , were calibrated for both call and put options as described in Section 6.2.	64
7.1	(a) optimal stock allocation, (b) optimal bond allocation	82
7.2	Relationship between consumption-wealth ratio, ambiguity and EIS	83
7.3	(a) Myopic stock allocation, (b) Myopic optimal bond allocation	84
7.4	(a) Stock demand due to observable parameter hedging, (b) Bond demand due to observable parameter hedging	85
7.5	(a) Stock demand due to unobservable parameter hedging, (b) Bond demand due to unobservable parameter hedging	86
7.6	(a) Stock demand due to interest rate hedging, (b) Bond demand due to interest rate hedging	87

List of Appendices

Appendix A: Matlab Code	91
-------------------------------	----

Chapter 1

Introduction

In the year 1900, Bachelier published his thesis "The Theory of Speculation", see [3]. In this work, Bachelier introduced the use of Brownian motion with zero drift to model the dynamics of stock movement. Unfortunately, Bachelier's work was mostly ignored for many decades until, in 1973, when Black and Scholes published their paper "The Pricing of Options and Corporate Liabilities", see [5], where they derived the famous, Black-Scholes equation; which cleverly recognized that the fair value of the option, follows the heat equation. In parallel to Black and Scholes, Samuelson and Merton, were working on the same problem, in the same year, Merton published "The Theory of Rational Options Pricing", see [34], in which he used stochastic calculus arguments to derive the Black-Scholes equation using Brownian motion. Since then many market models have been developed to price derivatives of not just common stock, but commodities and currency as well. The hedging problem is actually an older problem, the mean-variance or quadratic hedging problem is a specific case of the utility maximization problem, which was first proposed in 1738. The mean-variance hedging problem was popularized by Merton's paper "An Intertemporal Capital Asset Pricing Model", see [33]. This was extended to the general case by Schweizer, see [45] and then to incomplete markets with Föllmer, see [24]. Almost a decade later, Leukert and Föllmer introduced the notion of hedging risk of quantiles and presented techniques to solve them, see [23].

Asian options were first introduced in the commodities market in Tokyo, 1987, by Standish and Spaighton, two traders at Bankers Trust, now owned by Deutsche. In this work, we look more specifically at Arithmetic average fixed strike Asian options. We follow the work of Zhang and Oosterlee, see [47], who introduced a Fourier-Cosine series expansion method to price arithmetic average asian options. Ruijter and Oosterlee, see [42], extend the Fourier-Cosine method of Fang and Oosterlee, see [22], to the two dimensional case relying on the independence of the

underlying assets. Pellegrino et al, see [38], further extend the method to three dimensions, again under the assumption that the underlying assets are independent. Li and Chen, see [28], propose a pricing and hedging method using the Edgeworth series expansion, for diffusion type models. Deelstra et al, see [17] propose a general functional for the Asian basket spread payoff, Albrecher and Predota, see [1], present a pricing method based on the characteristics of the NIG Lévy processes and Boughamora et al, see [7], price and hedge Asian options under the relative entropy measure.

In this work we explore the application of time changed Lévy process to price and hedge Asian options derived from oil commodities. We present a time changed model with a subordinator made from a linear combination of two independent subordinating processes. This introduces an inherent dependence structure in the model. The model assumes that there is a linear structure in time between underlying assets, this tells us that a portfolio will be more volatile if the assets are more correlated. Intuitively this means that the underlying assets have a common process driving them, i.e. macroeconomic cycle. The parametric nature of the model also let's us use heavy-tailed distributions to capture heavy tailedness in the data. In our work we use the Asian Fourier-Cosine method of Zhang and Oosterlee, see [47], to price arithmetic Asian options using the proposed time changed model. We derive the univariate and bivariate characteristic function of the time changed model as well as its moments which is summarised in Tables 3.1-3.4. The Fourier-Cosine method was compared with an efficient Monte Carlo algorithm and from the numerical simulations it was found that the Fourier-Cosine method is computationally faster than the efficient Monte Carlo by about an order of magnitude, i.e. 2 seconds vs. 20 seconds. For a detailed error analysis of the Fourier Cosine method we refer the reader to the work of Fang and Oosterlee, see [22], and Zhang and Oosterlee, see [47]. We also note that the Fourier-Cosine method, exhibit some instability around the strike price of \$0.

We also considered the application of the Fourier-Cosine method in different hedging strategies. Alonso et al, see [2], also looks at the application of the Fourier-Cosine method to the hedging problem, but they only show a numerical simulation of delta and gamma hedging and makes an incomplete note on quantile hedging. In our work we assume that the investor hedges in a semi static way, in otherwords the investors only rebalances their portfolio outside the time interval $[t, T]$. We look at quadratic hedging problem as solved by Kolkiewicz et al, see [27] and quantile hedging as seen in the work of Föllmer and Leukert, see [23]. We found that quadratic, VaR (value at risk) and AVaR (average value at risk) risk cannot be completely hedged in an

incomplete market, this is a result that is well known. We also observed that investors looking to hedge VaR can simply hold the amount in a portfolio of mostly cash, whereas an investor hedging AVaR will need to hold more risky assets to generate larger returns. We also extend AVaR to a robust framework which is intended to reduce modelling error.

Through calibration against market data, we observe that our model successfully produces an implied volatility smile. We propose a conditional calibration method for the bivariate model and we approach the inverse problem using the Pattern Search algorithm as shown in Torczon's work, see [46]. The key advantage of this algorithm is that it is a direct search method, which implies that the function does not necessarily need to be differentiable. Torczon, see [46], provides analytic background to justify global convergence for convex cost functions. We find that the Pattern Search method works particularly well in solving calibration problems for our model.

The theory of Utility Maximization has a long history in macro and financial economics. It's central premise is the fact that asset prices are determined by investor's risk preferences and by the distributions of asset's risky future payments. Until 1713 it was believed that individuals value risky assets based solely on higher expected payoffs, this was contradicted by Nicholas Bernoulli by posing the St.Petersberg Paradox and in 1738 Daniel Bernoulli explained the paradox and proposed the concept of expected utility, the concavity of the expected utility was explained as risk aversion of the investor. Pratt(1964) and Arrow(1971) independently introduced a measure of absolute risk aversion which was extended to relative risk aversion, which is independent of wealth, see [39]. In this work, we use a recursive relative risk aversion preference of the Epstein-Zin type, see [19]

The reader is directed to Pennachi, see [39], for a general overview on the theory of Utility Maximization and Campbell and Viceira, see [11], for a general overview on portfolio choice for long-term investors. Campbell and Viceira, see [10], use regression to maximize Epstein-Zin type preferences to solve the optimal asset allocation and consumption problem for a long lived investor. Campbell et al, see [9] extend the framework to a continuous time VAR model. Campbell, Chan and Viceira, see [8], extend the framework to a multivariate setting. Maenhout, see [32], introduces the robustness problem and solves it for the CRRA (Constant Relative Risk Aversion) and Power Utility case. Liu, see [31], extends Maenhout's work to the EZ preferences case. Ju and Miao, see [26], presents a general utility function incorporating ambiguity and learning, in a regime switching asset pricing model with dividends. Escobar et al, see [20],

present a framework which looks at the long-term investors problem with CRRA preferences where the investor learns about stock returns with two parameters, an observed parameter and an unobserved parameter, which is estimated by optimal non-linear filtering.

In this work we follow the work of Escobar et al, see [20] and Liu, see [31]. We use the filtration techniques of Shiryaev and Liptser, see [30] to estimate the unobserved parameter with optimal non-linear filtering from observed parameters. Using the robust representation of Anderson (2012) as shown Escobar et al, see [20] we derive an explicit expression for the optimal portfolio and consumption as an institutional investor in an infinite horizon setting with relative preference utility of the Epstein-Zin type. For a robust investor, the introduction of ambiguity aversion effectively increases the risk aversions. When time preferences and risk preferences are separated it is interesting to explore how ambiguity aversion affects the robust investor who learns about stock returns and is willing to substitute consumption. In general, we find that if the investors willingness to substitute over time increases, the equity demand decreases and consumption-wealth ratio increases. As the preference for robustness increases the demand for risk decreases. We observe that learning about returns encourages the investor to short the riskless asset at all levels of ψ .

This work is divided into the following chapters. The first portion of this work is contained in Chapters 2 through 6. In Chapter 2 we introduce an extensive background into various existing results in probability, to ensure the work is complete as possible. Chapter 3, we build the model from it's foundations, we derive the Lévy triplet, the characteristic function, its exponential function and the moments. Chapter 4 looks at the application of the model to price Asian options using Monte Carlo simulation and Asian Fourier-Cosine pricing. Chapter 5 looks at the application of the Fourier-Cosine method to quadratic and quantile hedging and extends risk hedging to a robust framework. Chapter 6 presents some results of model calibration, we see that the model can generate the implied volatility smile and present a conditional parameter calibration approach to calibrate the bivariate case using pattern search algorithms for a root mean square error(RMSE) cost function. Chapter 7 contains the second portion of the thesis. This chapter is designed to be self contained and is supplemented by the background in Chapter 2. In this chapter, we discuss the implications of robustness preferences and intertemporal substitution to a long-term investor who learns about stock returns. Chapter 8 is the conclusion of the thesis and summarizes key results in both portions.

Chapter 2

Background

2.1 Preliminaries

Throughout this work we assume the standard setup. We assume that we work in a probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \in [0, T]}, \mathbb{P})$ with a non-decreasing filtration $\{\mathcal{F}_t\}_{t \in [0, T]}$ that follows the usual assumptions.

Definition 2.1. Filtration

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be some probability space with the σ -algebra, \mathcal{F} . Then the filtration of \mathcal{F} generated by some right continuous process $Y_t(\omega)$ is defined as $\{\mathcal{F}_t\}_{t \in [0, T]} := \sigma\{Y_t(\omega)\}$ for $\omega \in \Omega$, is the smallest σ -algebra generated by Y_t . The σ -algebra generated by the process Y_t is non-decreasing, i.e $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \dots \subseteq \mathcal{F}_T$

We assume the probability measure, \mathbb{P} , is the real world probability and there exists an equivalent martingale measure (EMM) \mathbb{Q} such that \mathbb{Q} is absolutely continuous with respect to \mathbb{P} , denoted $\mathbb{Q} \ll \mathbb{P}$, and we assume that our EMM \mathbb{Q} is the risk neutral measure. The drift $\mu(t, y) = \mu$, variance $\sigma(t, y) = \sigma$, and interest rate $r(t, y) = r$ are assumed to be constants. The log price for each asset, $j = 1, 2$, is assumed to follow a Lévy process $\{Y_t^{(j)}\}_{t \geq 0}$ and the price dynamic for each asset is assumed to be given by

$$S_t^{(j)} = S_0^{(j)} \exp(Y_t^{(j)})$$

For simplicity we assume that no dividends are paid and there are no transaction costs. The preliminaries are presented without proofs as we assume the reader is familiar with the fundamentals of continuous time stochastic analysis.

Definition 2.2. C adl ag Function

A function is c adl ag if it is right continuous and has left limits, i.e. for a function $f : [0, T] \mapsto \mathbb{R}^d$,

for each $t \in [0, T]$ the limit $f(t^-) = \lim_{s \rightarrow t, s < t} f(s)$ exists, the limit $f(t^+) = \lim_{s \rightarrow t, s > t} f(s)$ exists and $f(t) = f(t^+)$.

Remark 2.3. Any continuous function is càdlàg, but càdlàg functions are not necessarily continuous and can have discontinuities.

If there exists a discontinuity point at t we have that $\Delta f(t) = f(t) - f(t^-)$, which is called a jump at t . It is important to note that a càdlàg function can have at most a countable number of jumps, see [14].

Stock prices are modelled using random càdlàg functions. If we look at the filtration \mathcal{F}_t of all possible stock prices, from $t = [0, T]$ we know stock price at time t is $S_t = S_{t^+}$, which is by definition càdlàg.

Definition 2.4. Stopping Time

A random variable $T : \Omega \mapsto [0, \infty)$ is a stopping time if $[T \leq t] \in \mathcal{F}_t$, see [40].

Definition 2.5. Martingale

An adapted process $\{Y_t\}_{t \in [0, T]}$ is called a martingale with respect to \mathcal{F}_t

- (1) $\mathbb{E}|Y_t| < \infty$
- (2) if $s \leq t$, then $\mathbb{E}[Y_t | \mathcal{F}_s] = Y_s$ *a.s.*

Definition 2.6. Semi-Martingale

A process $\{Y_t\}_{t \in [0, T]}$ is called a semimartingale with respect to \mathcal{F} , if and only if $Y_t = M_t + A_t$ where M_t is a martingale and A_t is of finite variation. Y_t is càdlàg, adapted and is continuous in probability, see [40], [18]

Example 2.7. Let S_t be the adapted process that describes the asset price and Φ_t be a simple predictable process with stopping time T , the the wealth process is given by

$$X_t = X_0 + \int_0^t \Phi_t dS_t = \Phi_0 S_0 + \sum_{j=1}^d \Phi_t^j S_t^j \quad (2.1)$$

The wealth process is an example of a semimartingale, see [40].

Definition 2.8. Quadratic Variation

Let X and Y be semimartingales, the quadratic variation process of X is $[X, X] = X^2 - 2 \int X^- dX$ and the quadratic covariation process of X and Y is $[X, Y] = XY - \int X^- dX - \int Y^- dY$, see [40].

Corollary 2.9. Polarization Identity

$$[X, Y] = \frac{1}{2}([X + Y, X + Y] - [X, X] - [Y, Y])$$

Proof. The corollary is a result of the definition of quadratic variation, see [40]. \square

Definition 2.10. Quadratic Pure Jump Process

A semimartingale Y is called quadratic pure jump if $[Y, Y]^c = 0$, i.e. the continuous martingale term is zero and $[Y, Y] = Y_0^2 + \sum_{0 < s \leq t} (\Delta Y_s)^2$

2.2 Arbitrage-free Pricing

Let's consider a market with d assets whose prices are modelled as real valued càdlàg vector processes $S_t = (S_t^0, \dots, S_t^d) \in [0, T] \times \mathbb{R}^d$. A portfolio is a real valued vector $\Phi_t = (\Phi_t^0, \dots, \Phi_t^d)$, such that the value of the portfolio is given by $X_t(S_t, \Phi_t) = \sum_{j=1}^d \Phi_t^j S_t^j = \langle \Phi_t, S_t \rangle$

Definition 2.11. Predictable Processes

The predictable σ -algebra is the σ -algebra generated on $(0, T] \times \Omega$ by all càglàd processes i.e processes that are left continuous with right limits. A measurable random variable $\Phi_t : [0, T] \times \Omega \mapsto \mathbb{R}^d$ is called a predictable process, i.e. For $t = 0, 1, \dots, M$, Φ_t^j is \mathcal{F}_{t-1} -measurable, see [14].

In financial terms, t gives us our transaction dates as a filtration made from our σ -algebra generated by our stopping times; which is a countable random partition on $[0, T]$. The portfolio is chosen based on the information available at $t - 1$, which shows that Φ_t is \mathcal{F}_{t-1} -measurable. This means when a broker executes a trade at t they are working with portfolio values at $t - 1$.

Definition 2.12. Simple Predictable Processes

A simple predictable process $\{\Phi_t\}_{t \in [0, T]}$ is given by

$$\Phi_t = \Phi_0 \mathbb{1}_{t=0} + \sum_{n=1}^M \Phi_{t_n} \mathbb{1}_{[t_n, t_{n+1}]}$$
 (2.2)

Where each $t_0 = 0 < t_1 < t_2, \dots < t_M = T$ is a stopping time and each Φ_{t_n} is a predictable bounded random variable, see [14].

Definition 2.13. Self-Financing Portfolio

Given the portfolio strategy Φ_t , we say the the portfolio is self financing if and only if the wealth process $dX_t = \Phi_t dS_t \iff X_t = X_0 + \int_0^t \Phi_t dS_t$ where S_t is a càdlàg random process, see [18]

Definition 2.14. Arbitrage Opportunity

An arbitrage opportunity is a self-financing portfolio with a wealth process that satisfies the following conditions

- (1) $X_0(\Phi_t) = 0$

$$(2) \mathbb{P}[X_T(\Phi_t) \geq 0] = 1$$

$$(3) \mathbb{P}[X_T(\Phi_t) > 0] > 0$$

Proof. Sondermann, shows that as long as an equivalent martingale measure exists no-arbitrage opportunities exists, see [18], Theorem 4.7.2. \square

Definition 2.15. Attainable Admissible Trading Strategy

An attainable admissible trading strategy is a self-financing trading strategy that admits no arbitrage. i.e. Let $\Phi \in \mathcal{U}$, where \mathcal{U} is a subspace of admissible portfolios such that Φ_t is self financing strategy. There exists an EMM \mathbb{Q} such that

$$\mathbb{E}^{\mathbb{Q}}[X_t(\Phi_t)] = X_0$$

see [18]

Definition 2.16. Expectation under Risk Neutral Measure

We consider the EMM \mathbb{Q} such that, $S_t = S_t e^{-rt+Y_t}$ is a \mathbb{Q} -martingale, and Y_t follows the dynamics of Eq.(2.15) then we have that

$$e^{-r(T-t)} \mathbb{E}[X_t(S_t)] = \mathbb{E}^{\mathbb{Q}}[X_t(S_t)]$$

is a \mathbb{Q} -martingale when $\sigma = 0$ and $\mu = r - \int_{-\infty}^{\infty} (e^y - 1 - y) \nu_Y(dy)$

2.3 Lévy Processes

Definition 2.17. Lévy Processes

Let $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \in [0, T]}, \mathbb{P})$ be a probability space with filtration \mathcal{F}_t , we define an adapted stochastic process $Y_t := \{Y(t) | t \in [0, T]\}$ as a Lévy process, if it satisfies the following properties:

- (1) $Y_0 = 0$
- (2) Y_t has independent increments, i.e. $Y_t - Y_s$ where $[s < t] \in [0, T]$ are independent of \mathcal{F}_s .
- (3) Y_t has stationary increments, i.e. $Y_{t_0}, Y_{t_1} - Y_{t_0}, \dots, Y_{t_n} - Y_{t_{n-1}}$ are independent
- (4) Y_t is continuous in probability i.e. For each $\epsilon > 0$, $\lim_{s \rightarrow t} \mathbb{P}(|Y_s - Y_t| \geq \epsilon) = 0$.

see, [14]

Remark 2.18. Every Lévy process has a càdlàg modification which is itself a Lévy process, see [40].

Then, we can say that $\{Y_t^{(j)}\}_{t \geq 0}$ is a Lévy process, for any $t > 0$ the distribution of $\{Y_t^{(j)}\}_{t \geq 0}$ is infinitely divisible, see [14], [40].

Definition 2.19. Infinite Divisibility

A probability distribution p_Y on \mathbb{R}^d is said to be infinitely divisible if for any integer $d \geq 2$, there exists M identically independent distributed (i.i.d) random variables Y_1, \dots, Y_M such that $Y = \sum_{i=1}^M Y_n \sim p_Y$, see [14].

Theorem 2.20. Infinite Divisibility of Lévy Processes

Let Y_t be a Lévy process. Then for every t , Y_t has an infinitely divisible distribution. We can also say that if p_Y is an infinitely divisible distribution then there exists a Y_t such that $Y_1 \sim p_Y$

2.3.1 Markov Property of Lévy Processes

Definition 2.21. Markov Processes

Let X be a random variable with filtration \mathcal{F}_t then X is called Markov, when:

$$\mathbb{E}[X|\mathcal{F}_t] = \mathbb{E}[Y|Y_t]$$

where $Y \in \mathcal{F}_t$ see [14].

By definition the Lévy processes satisfies $\phi(u)_{Y_{dt}} = (\phi(u))^{dt}$, where $\Delta t = dt := t_i - t_{i-1}$ is the time increment; this is sufficient enough to say that Y_t has the Markov property. In fact this is a stronger version of the Markov property since it holds for each t . The strong Markov property of Lévy processes allows us to replace dt with any càdlàg random time, see [14].

Definition 2.22. Transition Operator for Strong Markov Processes

The transition operator for Markov processes is defined as

$$P_t f(t, y) = \mathbb{E}[f(t + \Delta t, \Delta Y_t)]$$

The linearity of of Lévy processes gives us

$$P_{t+s} = P_t P_s$$

2.3.2 Characteristic Function and Cumulants

Definition 2.23. Characteristic Function

The characteristic function ϕ_Y of a random variable Y , is the Fourier-Stieltjes transform of the distribution function $F(y)$, i.e:

$$\phi_Y(u) = \mathbb{E}[e^{iuY}] = \int_{-\infty}^{\infty} e^{iuY} dF(y), \quad i = \sqrt{-1}$$

Definition 2.24. Exponential Moments

Similarly we can define the exponential moments

$$\mathbb{E}[\exp[uY]] = \exp[t\psi_Y(-iu)] \quad (2.3)$$

Definition 2.25. Independent Characteristic Functions

For independent random variables X and Y we have:

$$\phi_{X+Y}(u) = \phi_X(u)\phi_Y(u)$$

Definition 2.26. Characteristic Exponent

For a random variable Y and $u \in \mathbb{R}$ the characteristic exponent of Y is defined as

$$\psi_Y(u) = \frac{1}{t} \log(\mathbb{E}[\exp[iuY]]) = \frac{1}{t} \log(\phi_Y(u)) \quad (2.4)$$

Remark 2.27. By Definition 2.26 it is easy to see that

$$\phi_x(u) = \exp[t\psi(u)] \quad (2.5)$$

Definition 2.28. Cumulant Function

Let Y_t be a Lévy process with characteristic exponent given by 2.4 and $\mathbb{E}[|Y_t|^n] < \infty$ for each $t > 0$, then it's cumulant generating function is given by, see [14]

$$\xi_n = \frac{t}{i^n} \frac{\partial^n (\psi(u))}{\partial u^n} \Big|_{u=0} \quad (2.6)$$

$$\xi_1(Y_{t+dt} - Y_t) = dt\mathbb{E}[Y_t] \quad (2.7)$$

$$\xi_2(Y_{t+dt} - Y_t) = dtVar[Y_t] \quad (2.8)$$

$$s(Y_{t+dt} - Y_t) = \frac{\xi_3(Y_{t+dt} - Y_t)}{\xi_2(Y_{t+dt} - Y_t)^{3/2}} = \frac{s(Y_t)}{\sqrt{dt}} \quad (2.9)$$

$$k(Y_{t+dt} - Y_t) = \frac{\xi_4(Y_{t+dt} - Y_t)}{\xi_2(Y_{t+dt} - Y_t)^2} = \frac{k(Y_t)}{dt} \quad (2.10)$$

The cumulants of Y_t increase linearly with t , i.e. $\xi_{Y_t} = t\xi_{Y_1}$, which holds because Y_t is infinitely divisible, see [14].

2.4 Representation

Definition 2.29. Jump measure

For a Lévy process Y_t with càdlàg paths and $0 \notin \bar{A}$ where $A \subset \mathcal{F}_t$ bounded away from 0, then the jump measure is given by:

$$N_t^A = \sum_{0 < s \leq t} \mathbf{1}_A(Y_{s^-} - Y_s) = \sum_{0 < s \leq t} \mathbf{1}_A(\Delta Y_s)$$

The jump measure is simply a counting measure, that counts the number of jumps in the path process, see [40].

Definition 2.30. Lévy measure

Let $\{Y_t\}_{t \in [0, T]}$ be a Lévy process on \mathcal{F}_t . Then, the Lévy measure ν of Y_t on \mathbb{R}^n is defined as:

$$\nu(A) = \mathbb{E}[N_1^A] = \mathbb{E}\left[\sum_{0 < s \leq 1} \mathbb{1}_A(\Delta Y_s)\right]$$

$\nu(A)$ is the expected number of jumps in A .

the definition tells us that the Lévy measure is the expected number of jumps that occur in our path process. Protter, [40], presents some nice results that we will use later.

Theorem 2.31. *Let $A \subset \mathcal{F}_t$, and g a measurable function on A , then*

$$\int_A g(y) N_t(ds, dy) = \sum_{0 < s \leq t} g(\Delta Y_s) \mathbb{1}_A(\Delta Y_s)$$

From the fact that N_t^A has independent and stationary increments, see [40], Theorem 34.

Corollary 2.32. *Given $A \subset \mathcal{F}_t$ with $0 \notin \bar{A}$ and let g be measurable on A , then*

$$\int_A g(y) N_t(ds, dy)$$

is a Lévy process, see [40].

Theorem 2.33. *Let $A \subset \mathcal{F}_t$, and ν be the Lévy measure of Y , and let $g \mathbb{1}_A \in L^2(d\nu)$. Then*

$$\mathbb{E}\left[\int_A g(y) N_t(ds, dy)\right] = t \int_A g(y) \nu(dx)$$

and

$$\mathbb{E}\left[\left(\int_A g(y) N_t(ds, dy) - t \int_A g(y) \nu(dx)\right)^2\right] = t \int_A g(y)^2 \nu(dx)$$

Proof. see, [40], Theorem 38 □

Theorem 2.34. Lévy-Itô Decomposition

Let Y_t be a Lévy process on \mathcal{F}_t and ν its Lévy measure. Then the Lévy process can be decomposed to:

$$Y_t = \mu(t, y)t + \sigma(t, y)B_t + J_t^l + \tilde{J}_t^c \tag{2.11}$$

$$J_t^l = \int_{|y| < 1} y N_t(dt, dy) \tag{2.12}$$

$$\begin{aligned} \tilde{J}_t^c &= \int_{\mathbb{R}} y (N_t(dt, dy) - t\nu(dy)) \\ &= \int_{\mathbb{R}} y \tilde{N}_t(dt, dy) \end{aligned} \tag{2.13}$$

where $\mu(t, y)$ is the drift coefficient, $\sigma(t, y)$ is the diffusion volatility and the two integral terms are independent Lévy jumps, see [14] [40]. The \tilde{J}_t^ε term captures small movements and the J_t^l term captures large movements. In tempered stable Lévy processes the parameter α emphasizes how important these movements are. When $\alpha = 0$ (Gamma) the process favours large movements and when $\alpha = 1/2$ (Inverse Gaussian) the process favours small movements, see [14].

Theorem 2.35. Itô Formula

Let Y_t be a semi-martingale process and let $f(t, Y_t)$ be a $C^{1,2}$ function, then the Itô formula

$$\begin{aligned} df(t, Y_t) &= \frac{\partial f}{\partial t} dt + \frac{\partial f}{\partial y} [\mu dt + \sigma dB_t] + \frac{\sigma^2}{2} \frac{\partial^2 f}{\partial y^2} dt \\ &+ \int_{|y|<1} [f(t, Y_t) - f(t, Y_{t-}) - y \frac{\partial f}{\partial y}] \nu(dy) dt \\ &+ \int_{\mathbb{R}} [f(t, Y_t) - f(t, Y_{t-})] \tilde{N}_t(dt, dy) \end{aligned} \quad (2.14)$$

Theorem 2.36. Lévy-Itô Isometry

Let Y_t , a semi-martingale with stopping time T . Then the solution given by applying the Itô formula to equation (2.11) gives us

$$\begin{aligned} Y_t &= Y_0 + \int_0^T \mu dt + \int_0^T \sigma dB_t + \int_0^T \int_{|y|<1} y N_t(dt, dy) \\ &+ \int_0^T \int_{\mathbb{R}} y \tilde{N}_t(dt, dy) \end{aligned} \quad (2.15)$$

and has the differential representation

$$\begin{aligned} dY_t &= \mu dt + \sigma dB_t + \int_{|y|<1} y N_t(dt, dy) \\ &+ \int_{\mathbb{R}} y \tilde{N}_t(dt, dy) \end{aligned} \quad (2.16)$$

Assuming $\mathbb{E}[Y_t] < \infty$ and if Y_t has the stopping time T , $Y_0 = 0$ and $\mu(t, y) = 0$ then

$$\mathbb{E}[Y_T^2] = \mathbb{E}\left[\int_0^T \sigma^2 dt + \int_0^T \int_{\mathbb{R}} y^2 \nu(dy) dt\right] \quad (2.17)$$

see [37]

Theorem 2.37. Lévy Khintchine Representation

Given a Lévy process Y and its Lévy triplet (μ, σ^2, ν) . The characteristic exponent of Y satisfies the following formulation:

$$\psi_Y(u) = iu\mu - \frac{u^2\sigma^2}{2} + \int_{\mathbb{R}} (e^{iuy} - 1 - iuy\mathbf{1}_{|y|<1}) \nu(dy) \quad (2.18)$$

$\mu \in \mathbb{R}$, $\sigma^2 \in \mathbb{R}_+$, and ν is a measure on $\mathbb{R} \setminus \{0\}$.

Remark 2.38. The Lévy Khintchine form is quite useful in determining the expected value of exponential Lévy processes since $\mathbb{E}[\exp(Y_t)] = \exp(t\varphi_Y(-i))$

The Lévy triplets are known for some exponential Lévy processes as seen in [44]. μ is the drift coefficient, σ is the volatility and ν is the Lévy measure. If the Lévy measure has the form $\nu(dy) = u(y)dy$, then $u(y)$ is the Lévy density, which has the same characteristics as the probability density but does not need to be integrable and is zero at the origin.

2.4.1 Exponential Lévy Processes

Definition 2.39. Stochastic Exponential

For a Lévy process S_t , $dS_t = S_t dY_t$, the stochastic exponential of S_t is given by

$$Z_t = \mathcal{E}(Y_t) = \exp\left(Y_t - \frac{1}{2}[Y, Y]_t^c\right) \quad (2.19)$$

Where $[Y, Y]_t^c = 0$ is the path-continuous part of $[Y, Y]$ of Y_t , for a proof see [40], Theorem 28.

Under pure jump processes when $\int_{-1}^1 |y| \nu_Y(dy) < \infty$ the stochastic exponential becomes

$$\begin{aligned} Z_t &= \mathcal{E}(Y) = e^{Y_t - \frac{\sigma^2 t}{2}} \prod_{0 < s \leq t} (1 + \Delta Y_s) e^{-\Delta Y_s} \\ dZ_t &= dY_t + \frac{\sigma^2}{2} dt + \sum_{0 \leq s \leq t} (e^{\Delta Y} - 1 - \Delta Y) \end{aligned} \quad (2.20)$$

Applying the Itô formula with $f(t, y) = e^{y - \frac{\sigma^2 t}{2}} \prod_{0 < s \leq t} (1 + y) e^{-y}$ we get that $dZ_t = Z_t dY_t$ where dY_t is given by

$$dY_t = \left(\mu_Y - \frac{\sigma^2}{2}\right) dt + \int_0^T \int_{-1}^1 (e^y - 1 - y) \nu_Y(dy) dt + \int_0^T \int_{\mathbb{R}} (e^y - 1) \tilde{N}_t(dt, dy) \quad (2.21)$$

Theorem 2.40. Ordinary and Stochastic Exponential

- (1) Let X_t be a Lévy process with Lévy triplet $(\mu_X, \sigma_X^2, \nu_X)$ and $Z_t = \mathcal{E}(X_t)$. If $X_t > 0$ a.s. then there exists another Lévy process Y_t such that $Z_t = e^{Y_t}$ where

$$Y_t = \log(X_t) = X_t - \frac{\sigma^2 t}{2} + \sum_{0 < s \leq t} [\log(1 + \Delta X_s) - \Delta X_s] \quad (2.22)$$

With Lévy triplets $(\mu_Y, \sigma_Y^2, \nu_Y)$

$$\mu_Y = \mu_X - \frac{\sigma^2}{2} + \int_{[-1,1]} (\log(1+x) - x) \nu(dx)$$

$$\sigma_Y = \sigma_X$$

$$\nu_Y = \int \mathbf{1}_{|\log(1+x)| \leq 1} \nu_X(dx)$$

(2) *Conversely*

$$X_t = Y_t + \frac{\sigma^2 t}{2} + \sum_{0 < s \leq t} [1 + \Delta Y_s - e^{\Delta Y_s}] \quad (2.23)$$

With Lévy triplets $(\mu_X, \sigma_X^2, \nu_X)$

$$\mu_X = \mu_Y + \frac{\sigma^2}{2} + \int_{\mathbb{R}} [e^y - 1 - y] \nu_Y(dy)$$

$$\sigma_X = \sigma_Y$$

$$\nu_X = \int \mathbb{1}_{|e^y - 1| \leq 1} \nu_Y(dy)$$

Under the EMM \mathbb{Q} and stopping time T , the evolution of asset prices is usually described by an exponential Lévy model with the form $S_t = e^{-r(T-t)} \mathbb{E}[S_t]$ and $S_t = \exp(Y_t)$, where Y_t is a Lévy process given by equation (2.15). By applying the Itô formula the process S_t is given by

$$\begin{aligned} S_t &= e^{-rT} S_{t-} \left[\int_0^T (\mu_Y - r) dt \right. \\ &\quad + \int_0^T \int_{-1}^1 (e^y - 1 - y) \nu_Y(dy) dt \\ &\quad \left. + \int_0^T \int_{\mathbb{R}} (e^y - 1) \tilde{N}_t(dt, dy) \right] \end{aligned} \quad (2.24)$$

To check if an exponential Lévy process is a martingale, we need to check that $\mathbb{E}^{\mathbb{Q}}[S_t] < \infty$.

Under the expectation we get

$$\mathbb{E}^{\mathbb{Q}}[\mathcal{F}_t] = e^{-rT} S_{t-} (A_t + M_t) \quad (2.25)$$

$$A_t = \int_0^T \left[\mu_Y - r - \frac{\sigma_Y^2}{2} + \int_{-1}^1 (e^y - 1 - y) \nu_Y(dy) \right] dt \quad (2.26)$$

$$M_t = \int_0^T \sigma_Y dB_t + \int_0^T \int_{\mathbb{R}} (e^y - 1) \tilde{N}_t(dt, dy) \quad (2.27)$$

Where M_t is the martingale term and A_t is a drift term. Under the EMM \mathbb{Q} , we get a martingale by setting $A_t = 0$, i.e. for some interest rate $r \in \mathbb{R}$ and $\sigma_Y = 0$

$$\mu_Y - r + \int_{-1}^{\infty} (e^y - 1 - y) \nu(dy) = 0 \quad (2.28)$$

The integral term $m = \int_{-1}^{\infty} (e^y - 1 - y) \nu(dy)$ is given for different Lévy processes.

2.4.2 Girsanov Transform

Definition 2.41. Radon-Nikodym Derivative

Let \mathbb{Q} be another probability measure on $(\Omega, \mathcal{F}, \mathcal{F}_t)$ such that $\mathbb{Q} \ll \mathbb{P}$. Then there exists $Z_t \in \mathcal{L}^1(\Omega, \mathcal{F})$ with $d\mathbb{Q} = Z d\mathbb{P}$, i.e.

$$\mathbb{Q}(A) = \int_A Z_t(\omega) \mathbb{P}(d\omega)$$

for each $A \subset \mathcal{F}_t$. Solving for Z we get

$$Z_t = \frac{d\mathbb{Q}}{d\mathbb{P}} \implies \mathbb{E}^{\mathbb{Q}}[\mathbf{1}_A] = \mathbb{E}[Z_t | \mathcal{F}_t] = \mathbb{E}\left[\frac{d\mathbb{Q}}{d\mathbb{P}} \middle| \mathcal{F}_t\right] \quad (2.29)$$

Z_t is a right continuous martingale such that $\mathbb{E}[Z_t | \mathcal{F}_t] = 1$ and

$$(1) Z_t(\omega) > 0 \text{ } \mathbb{Q}\text{-a.s.}$$

$$(2) Z_t = \frac{d\mathbb{Q}_t}{d\mathbb{P}_t} \text{ on } \mathcal{F}_t$$

see, [18]

Theorem 2.42. Girsanov Transform

Let $\mathbb{Q} \ll \mathbb{P}$, $Z = \frac{d\mathbb{Q}}{d\mathbb{P}}$ continuous, and M_t be a \mathbb{P} -martingale, then the \mathbb{Q} -martingale \tilde{M}_t is given by

$$\tilde{M}_t = M_t - \int_0^t \frac{1}{Z_s} d[M, Z]_s$$

When $\mathbb{Q} \stackrel{d}{=} \mathbb{P}$, for Lévy process Y such that $Z = \mathcal{E}(Y)$ we get that

$$d\mathbb{Q} = \mathcal{E}(Y) d\mathbb{P} \quad (2.30)$$

and

$$d\mathbb{P} = \frac{1}{\mathcal{E}(Y)} d\mathbb{Q} \quad (2.31)$$

Note that $\mathbb{E}[Z_0] = 1$ a.s on measure ν

Lemma 2.43. For some underlying asset price given by the a pure jump Lévy process $dS_t = S_t - dY_t$, where $dY_t = \mu_Y dt + \int_{\mathbb{R}} (\log(1+y) - y) N_t(dt, dy) + \int_{\mathbb{R}} \log(1+y) \tilde{N}_t(dt, dy)$ applying the Girsanov transform for some stopping time T , the discounted price, given by $\hat{S}_t = e^{-r(T-t)} S_t$ for interest rate $r > 0$ and $\sigma_{\mu_Y} = 0$. We define

$$Z_t = \exp\left(\int_0^T \int_{-1}^{\infty} (\log(1+y) - y) \tilde{N}_t(dt, dy) + \int_0^T (\mu_Y - r) dt\right) = \exp(rt + Y_t) \quad (2.32)$$

2.4.3 Feynman-Kac Formula

Let Y_t be a Lévy process with the Lévy decomposition given by equation (2.11). The Feynman-Kac Formula considers the problem

$$\frac{\partial f(t, Y_t)}{\partial t} + \mathcal{A}f(t, y) = 0 \quad (2.33)$$

where $\mathcal{A}f$ is the infinitesimal generator of f , r is a positive real constant and $(t, y) \in [0, T] \times \mathbb{R}$ with boundary condition $f(T, y) = H(y)$

Definition 2.44. Infinitesimal Generator of Lévy Processes

$\mathcal{A}f$ is the infinitesimal generator of $f(t, y)$, it is given by

$$\begin{aligned}
\mathcal{A}f &= \lim_{dt \rightarrow 0} \frac{1}{dt} (P_t f(t, y) - f(t, y)) \\
&= \lim_{t \rightarrow 0} \frac{1}{t} (E[f(t + dt, \Delta Y_t)] - f(t, y)) \\
&= \mu(t) \frac{\partial f(t, y)}{\partial y} + \frac{\sigma^2(t)}{2} \frac{\partial^2 f(t, y)}{\partial y^2} \\
&\quad + \int_{\mathbb{R}} [f(t, \Delta Y_t) - f(t, y) - y \mathbb{1}_{|y| < 1} \frac{\partial f(t, y)}{\partial y}] \nu(dy)
\end{aligned} \tag{2.34}$$

Remark 2.45. The solution of the Feynman-Kac formula, if it exists is given by

$$f(t, y) = \mathbb{E} \left[\exp \left[- \int_t^T r ds \right] H(Y_T) \middle| Y_t = y \right] \tag{2.35}$$

2.5 Time-Changed Lévy Process

There are many Lévy processes that are used to price options. The most well known is Brownian motion, which has been used quite successfully in Finance, but present some drawbacks. One of the drawbacks of Brownian motion is that it fails to capture the jumps in asset prices, it also fails to capture the heavy-tail property and the skewness in distributions seen in Financial data of asset returns. Another class of Lévy processes, the jump-diffusion process is a valid candidate for pricing options, but it does not capture stochastic volatility. This is what motivated the use of stochastic volatility models first by Clark, see [12], to model asset returns. Time-changed processes is one way to include stochastic volatility in a model and it's underlying distributions are rich enough to capture the properties missing in other models. This is what motivated the study into time changed Lévy Processes.

2.5.1 Multivariate Subordination

Multivariate subordination was introduced by Barndorff-Nelson, Persen and Sato (2015), they proved that the linear combination of n independent subordinated processes is again a Levy process, see [4]. Benth and Kruhner, extended their work to generalize the results of Barndorff-Nelson, Persen and Sato. We will restate some results from Benth and Kruhner, see [4]. Similar results can be found in a reference text by Cont and Takov, see [14].

Definition 2.46. Subordinators

Let $\{R_t\}_{t \in [0, T]}$ be a càdlàg positive increasing Lévy process, we call such processes subordinators, see [14].

Theorem 2.47. Ch.f. of Subordinated Lévy

Let be Y_t a subordinated Lévy process, R_t the subordinator and X_t the Lévy process being subordinated; X_t is independent of R_t . Then the characteristic function of Y_t is given by

$$\phi_{Y_t}(u) = \exp(t\psi_{R_t}(-i\psi_{X_t}(u))) = \exp(tl_{R_t}(\psi_{X_t}(u))) \quad (2.36)$$

Proof. Let $Y_t = X_{R_t}$ be a subordinated Brownian motion then

$$\begin{aligned} \phi_{Y_t}(u) &= \mathbb{E}[\exp(iuY_t)] \\ &= \mathbb{E}[\mathbb{E}[\exp(iu(\mu R_t + \sigma B_{R_t})) | R_t]] \\ &= \mathbb{E}[\exp(R_t(iu\mu - \frac{u^2\sigma^2}{2}))] \text{ Let } v = iu\mu - \frac{u^2\sigma^2}{2} \\ &= \mathbb{E}[\exp(R_tv)] \\ &= \exp(t\psi_{R_t}(-iv)) \\ &= \exp(t\psi_{R_t}(-i\psi_{X_t}(u))) \end{aligned}$$

and the rest follows. \square

Theorem 2.48. For a Lévy process $Y_t = c_0X_t^0 + c_jX_t^j$, $j = 1, 2$, where X_t^0 and X_t^j are independent Lévy processes and c_0, c_j constants, then $\{Y_t\}_{t \geq 0}$ has the Lévy triplets given by:

$$\mu_Y = c_0\mu_{X^0} + c_j\mu_{X^j} + \int_{\mathbb{R}^n} y[\mathbb{1}_{|y| \leq 1} - \mathbb{1}_{[c_0y + c_jy ||y| \leq 1]}] \rho_Y(dy) \quad (2.37)$$

$$\rho_Y = \rho_{L_t^0}(c_0x) + \rho_{L_t^j}(c_jx) \quad (2.38)$$

$$\sigma_Y = c_0\sigma_{L_t^0} + c_j\sigma_{L_t^j} \quad (2.39)$$

Proof. see [14], Theorem 4.1 \square

Theorem 2.49. Let $Y_t = X_{R_t}$ be a subordinated Lévy process and assume $\int_{\mathbb{R}_+^n} |Y_t| \rho_R(dR_t) < \infty$, X_t and R_t have Lévy triplets (μ_X, σ_X, ν_X) and $(\mu_R, 0, \rho)$ then

$$\sigma_Y = \mu_R \sigma_X \quad (2.40)$$

$$\nu_Y = \mu_R \nu_X(A) + \int_0^\infty p_{X_t}(A) \rho_R(dt) \quad (2.41)$$

$$\mu_Y = \mu_R \mu_X + \int_0^\infty \rho_R(ds) \int_{|x| \leq 1} xp_{X_s}(dx) \quad (2.42)$$

where $A \subset \mathcal{F}_t$ and ρ is the Lévy measure of the subordinator.

Proof. see [4], Theorem 2.4 \square

Chapter 3

A Time-Changed Exponential Lévy Model

3.1 Introduction

Lévy processes are a class of processes that are widely used in asset pricing. It contains processes such as Brownian motion, compound Poisson process, jump-diffusion process, α -stable process and many others. There are many different methods to construct Lévy processes, one of which is by subordination. We present a parametric model which adds dependence through the subordinator in a multi-asset portfolio. The resulting Lévy process created from the subordination of another independent Lévy process, i.e. subordinated Brownian motion, is called a time-changed process, first introduced by Clark, see [12]. The price of the assets are the exponential of time-changed processes where the subordinator is given by the Inverse Gaussian process and Variance Gamma process. The process being subordinated is a Brownian motion independent of the subordinator process.

Our model for the log-price of an asset return is a subordinated Brownian motion where the subordinator is constructed from the linear combination of two independent Lévy processes; the new subordinator has all the properties of a Lévy processes due to the independence of Lévy measures, see [14].

We choose to work with subordinated Brownian motion because it is rich enough that captures small and large jumps in asset prices as well as capture the heavy-tail distribution of real asset returns, see [14]. Time-changed processes incorporate stochastic volatility in the model by setting time to be stochastic, i.e. Time runs faster in periods of high volatility, see [44].

This effect allows us to reproduce the volatility smile, which the classical Black-Scholes model cannot, see [14]. Next we extend the model to two dimensions to price Asian basket spread options.

3.2 Model Setting and General Model

Consider a risk neutral market with assets whose prices are represented by the vector $\mathbf{S}_t = [S_t^1, S_t^2]^\top$, where $j = 1, 2$, are risky assets whose dynamics are given by:

$$\mathbf{S}_t = \begin{bmatrix} S_0^1 e^{Y_t^1} \\ S_0^2 e^{Y_t^2} \end{bmatrix} \quad (3.1)$$

where each Y_t^j , $j = 1, 2$ is a subordinated Brownian motion given by:

$$Y_t^j = \sigma_Y^j B_{R_t^j}^j + \mu_Y^j R_t^j, \quad j = 1, 2 \quad (3.2)$$

where σ_Y^j is the volatility of the j^{th} risky asset and μ_Y^j is the drift of the j^{th} risky asset. The subordinator of each asset j is a linear combination of two independent subordinators given by

$$R_t^j = L_t^0 + c_j L_t^j, \quad j = 1, 2 \quad (3.3)$$

where L_t^0 is the shared subordinator for all assets $j = 1, 2$, L_t^j . L_t^j is an independent increasing Lévy process unique to each asset, with a coefficient $c_j > 0$; the coefficient c_j must be positive because the subordinator process is by definition always increasing. In financial terms c_j acts to "speed up" business time, mathematically the coefficient adds a fractional gain and introduces a linear structure in time. Notice that for the special case where $c_j = 0$ we get the standard subordinated Lévy process.

We can also represent the model as a pure jump process, by the Lévy-Itô decomposition:

$$\begin{aligned} Y_t^j &= \mu_Y^j t + J_t^{l,j} + \tilde{J}_t^{\epsilon,j}, \quad j = 1, 2 \\ J_t^{l,j} &= \int_{|y| < 1} y N_t(dt, dy) \\ \tilde{J}_t^{\epsilon,j} &= \int_{\mathbb{R}} y (N_t(dt, dy) - t\nu_{Y^j}(dy)) \end{aligned} \quad (3.4)$$

3.2.1 Characteristic Function and Lévy Triplets of the Model

Theorem 3.1. *For each asset $j = 1, 2$, let Y_t , be a subordinated Brownian motion with μ and σ real valued constants to be calibrated. Then Y_t follows equation (3.2) and its characteristic function is given by:*

$$\phi_{Y_t}(u) = \exp(t\psi_{L_t^0}^0(-iv)) \exp(t\psi_{c_j L_t^j}^j(-iv)) \quad (3.5)$$

where $v = iu\mu - \frac{1}{2}u^2\sigma^2$

Proof. The Theorem follows directly from applying Theorem 2.36 So by using the towering property

$$\begin{aligned}
\phi_{Y_t}(u) &= \mathbb{E}[\mathbb{E}[e^{iu(\mu R_t + \sigma B_{R_t})} | R_t]] \\
&= \mathbb{E}[e^{R_t(-i(iu\mu - \frac{1}{2}u^2\sigma^2))}] \\
&= e^{t\psi_{R_t}(-iv)}, \quad v = iu\mu - \frac{1}{2}u^2\sigma^2 \\
&= e^{t(\psi_{L_t^0}(-iv) + \psi_{c_j L_t^j}(-iv))} \\
&= e^{t\psi_{L_t^0}(-iv)} e^{t\psi_{c_j L_t^j}(-iv)}
\end{aligned} \tag{3.6}$$

and the rest follows. \square

Theorem 3.2. For each asset $j = 1, 2$, let Y_t be the subordinated Lévy process, X_t a Brownian motion and $R_t = L_t^0 + c_j L_t^j$ the subordinator with Lévy triplets $(\mu_Y, \sigma_Y^2, \nu_Y), (\mu_X, \sigma_X^2, 0), (0, 0, \rho_R)$ respectively.

$$\begin{aligned}
\mu_Y &= \int_0^\infty \int_{-1}^1 \frac{x}{\sigma_{X^0} \sqrt{2\pi s}} \exp\left[-\frac{(x - \mu_{X^0})^2}{2t\sigma_{X^0}^2}\right] dx \rho_{L^0}(ds) \\
&\quad + \int_0^\infty \int_{-1}^1 \frac{x}{\sigma_{X^j} \sqrt{2\pi s}} \exp\left[-\frac{(x - \mu_{X^j})^2}{2t\sigma_{X^0}^2}\right] dx \rho_{L^j}(c_j ds)
\end{aligned} \tag{3.7}$$

$$\sigma_Y = 0 \tag{3.8}$$

$$\begin{aligned}
\nu(y) &= \int_0^\infty \frac{1}{\sigma_{X^0} \sqrt{2\pi t}} \exp\left[-\frac{(y - \mu_{X^0} t)^2}{2t\sigma_{X^0}^2}\right] \rho_{L^0}(dt) \\
&\quad + \int_0^\infty \frac{1}{\sigma_{X^j} \sqrt{2\pi t}} \exp\left[-\frac{(y - \mu_{X^j} t)^2}{2t\sigma_{X^j}^2}\right] \rho_{L^j}(c_j dt)
\end{aligned} \tag{3.9}$$

Proof. The Lévy triplets of Y_t are computed by applying Theorem 2.48 and Theorem 2.49

$$\begin{aligned}
\rho_R(x) &= \rho_{L_t^0}(x) + \rho_{L_t^j}(c_j x) \\
\mu_Y &= \int_0^\infty \rho_L(ds) \int_{-1}^1 x p_{X_s}(dx) \\
&= \int_0^\infty \int_{-1}^1 \frac{x}{\sigma_X \sqrt{2\pi s}} \exp\left[-\frac{(x - \mu_X s)^2}{2t\sigma_X^2}\right] dx \rho_R(ds) \\
&= \int_0^\infty \int_{-1}^1 \frac{x}{\sigma_X \sqrt{2\pi s}} \exp\left[-\frac{(x - \mu_X s)^2}{2t\sigma_X^2}\right] dx [\rho_{L^0}(ds) + \rho_{L^j}(c_j ds)] \\
&= \int_0^\infty \int_{-1}^1 \frac{x}{\sigma_{X^0} \sqrt{2\pi s}} \exp\left[-\frac{(x - \mu_{X^0})^2}{2t\sigma_{X^0}^2}\right] dx \rho_{L^0}(ds) \\
&\quad + \int_0^\infty \int_{-1}^1 \frac{x}{\sigma_{X^j} \sqrt{2\pi s}} \exp\left[-\frac{(x - \mu_{X^j})^2}{2t\sigma_{X^j}^2}\right] dx \rho_{L^j}(c_j ds) \\
\sigma_Y &= \mu_R \sigma_X \\
&= 0 \\
\nu_Y(y) &= \mu_R \mu_X + \int_0^\infty p_X(y) \rho_R(dt)
\end{aligned}$$

$$\begin{aligned}
&= \int_0^\infty \frac{1}{\sigma_X \sqrt{2\pi t}} \exp\left[-\frac{(y - \mu_X t)^2}{2t\sigma_X^2}\right] \rho_R(dt) \\
&= \int_0^\infty \frac{1}{\sigma_X \sqrt{2\pi t}} \exp\left[-\frac{(y - \mu_X t)^2}{2t\sigma_X^2}\right] [\rho_{L^0}(dt) + \rho_{L^j}(c_j dt)] \\
&= \int_0^\infty \frac{1}{\sigma_{X^0} \sqrt{2\pi t}} \exp\left[-\frac{(y - \mu_{X^0} t)^2}{2t\sigma_{X^0}^2}\right] \rho_{L^0}(dt) \\
&+ \int_0^\infty \frac{1}{\sigma_{X^j} \sqrt{2\pi t}} \exp\left[-\frac{(y - \mu_{X^j} t)^2}{2t\sigma_{X^j}^2}\right] \rho_{L^j}(c_j dt)
\end{aligned}$$

□

Using the Lévy triplets we can also derive an alternate representation of the characteristic function using the Lévy-Khintchine representation

Theorem 3.3. *Let Y_t be a subordinated Brownian motion, with Lévy triplets $(\mu_Y, \sigma_Y^2, \nu_Y)$ represented as equation (3.4), it's characteristic function is given by*

$$\phi_Y(u) = e^{t\psi_Y(u)} \quad (3.10)$$

where

$$\psi_Y(u) = iu\mu_Y + \int_{\mathbb{R}} [e^{iuy} - 1 - iuy\mathbf{1}_{|y|<1}] \nu_Y(dy) \quad (3.11)$$

Proof. This is the result of directly applying Theorem 2.37 with Lévy triplets $(\mu_Y, 0, \nu_Y)$. □

3.3 Univariate Model

The univariate case is a special case where the subordinator R_t is a real-valued process and $c_j = 0$. The characteristic function and Lévy triplets of Y_t for the NIG and VG processes are known in closed form, see [14], Table 4.5.

3.3.1 NIG Case

Definition 3.4. IG Processes

Let $L_t \sim IG_t(a, b)$ be a real valued process, where $a = \frac{t}{\sqrt{\kappa}}$, $b = \frac{1}{\sqrt{\kappa}}$ then $R_t = L_t$ is an Inverse Gaussian process with the Lévy density

$$\rho_R(x) = \begin{cases} \frac{1}{\sqrt{2\pi\kappa}} \frac{e^{-\frac{x}{2\kappa}}}{x^{3/2}} & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

where κ is the variance of the IG process. The exponential moment is given by

$$\mathbb{E}[e^{uR_t}] = \exp\left(-\frac{t}{\sqrt{2\kappa}}(\sqrt{1/2\kappa - u} - \sqrt{1/2\kappa})\right) \quad (3.12)$$

see [14], Table 4.4

The NIG process is an infinite variation process with stable-like behavior of small jumps, as shown by Cont and Tankov, see [14]. As a consequence, this model gives more emphasis to local small movement of asset prices.

Definition 3.5. NIG Processes

Let Y_t be a subordinated Brownian motion with an IG subordinator, then the process $Y_t \sim NIG_t(\mu_X, \sigma_X, \kappa)$, as shown by Schoutens, see [44]. In the univariate case the Lévy measure, characteristic exponent and cumulants are known in closed form, where X_t is a Brownian motion with drift μ_X and volatility σ_X^2 . The Lévy measure is given by

$$\begin{aligned}\nu_Y(x) &= \frac{C}{|x|} e^{Ax} K_1(B|x|) \\ A &= \frac{\mu_X}{\sigma_X^2} \\ B &= \frac{\sqrt{\mu_X^2 + \sigma_X^2/\kappa}}{\sigma_X^2} \\ C &= \frac{\sqrt{\mu_X^2 + \sigma_X^2/\kappa}}{2\pi\sigma_X\sqrt{\kappa}}\end{aligned}$$

K_1 is the modified Bessel function of the second kind which is given by

$$\begin{aligned}K_w(z) &= \frac{\pi}{2} \frac{I_{w-}(z) - I_w(z)}{\sin(\pi w)} \\ I_{w-}(z) &= I_w(z) \\ I_w(z) &= \sum_{k=0}^{\infty} \frac{(z/2)^{w+k}}{k!\Gamma(k+w+1)}\end{aligned}$$

Following Schoutens, see [44], the characteristic function is given by

$$\phi_Y(-iv) = \exp(-at(\sqrt{-2i(-iv) + b^2} - b)) \quad (3.13)$$

Where $a = \frac{1}{\sqrt{\kappa}}$, $b = \frac{1}{\sqrt{\kappa}}$ and $v = iu\mu_X - \frac{1}{2}u^2\sigma_X^2$ then it's characteristic exponent is given by

$$\psi_Y(-iv) = \log(\phi_Y(u))/t = -a(\sqrt{-2i(-iv) + b^2} - b) \quad (3.14)$$

and it's first two cumulants are given by

Table 3.1: Table of Cummulants for Univariate IG Subordinated Lévy Process

Cummulants	
$\mu_Y = \xi_1$	$t\mu_X$
$\sigma_Y^2 = \xi_2$	$t(\sigma_X^2 + \mu_X^2\kappa)$
s	$t(3\sigma_X^2\mu_X\kappa + 3\mu_X^3\kappa^2)$
k	$t(3\sigma_X^4\kappa + 15\mu_X^4\kappa^3 + 18\sigma_X^2\mu_X^2\kappa^2)$

see [14], Section 4.4.3

3.3.2 Variance Gamma(VG) Processes

Like the univariate NIG case, the univariate VG case is a special case where the subordinator $R_t = G_t \sim \Gamma(a, b)$ is a real-valued process and $c_j = 0$. The characteristic function and Lévy triplets of the NIG, Y_t are known in closed form, see [14], Table 4.5.

Definition 3.6. Gamma Processes

Let $G_t \sim \Gamma_t(a, b)$ be a real valued process, where $a = \frac{t}{\kappa}$, $b = \frac{1}{\kappa}$. Then $R_t = G_t$ is a Gamma process with Lévy density

$$\rho_R(x) = \begin{cases} \frac{1}{\kappa} \frac{e^{-\frac{x}{\kappa}}}{x} & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

where κ is the variace of the Gamma process. The exponential moment of the Gamma process is given by

$$\mathbb{E}[e^{uR_t}] = (1 - \kappa u)^{-t/\kappa} \quad (3.15)$$

The Variance Gamma process is a finite variation process with low infinite activitiy of small jumps, see [14]. As a consequence, this model gives more emphasis to larger jumps in asset prices.

Definition 3.7. VG Processes

Let Y_t be a subordinated Brownian motion with a Gamma subordinator, then the process $Y_t \sim VG(\mu_X, \sigma_X, \kappa)$, see [44]. In the univariate case the Lévy measure, characteristic exponent and cumulants are known in closed form, where X_t is a Brownian motion with drift μ_X and volatility σ_X^2 . The Lévy measure is given by

$$\begin{aligned} \nu_Y(x) &= \frac{1}{\kappa |x|} e^{Ax - B|x|} \\ A &= \frac{\mu_X}{\sigma_X^2} \\ B &= \frac{\sqrt{\mu_X^2 + \sigma_X^2/\kappa}}{\sigma_X^2} \end{aligned}$$

Following Schoutens, see [44], the characteristic function is given by

$$\phi_Y(-iv) = \left(1 - \frac{i(-iv)}{b}\right)^{-at} \quad (3.16)$$

where $a = \frac{1}{\kappa}$, $b = \frac{1}{\kappa}$ and $v = iu\mu_X - \frac{1}{2}u^2\sigma_X^2$ then it's characteristic exponent is given by

$$\psi_Y(-iv) = \log(\phi_Y(-iv))/t = -a \log\left(1 - \frac{i(-iv)}{b}\right) \quad (3.17)$$

and it's first two cummulants are given by

Table 3.2: Table of Cummulants for Univariate Gamma Subordinated Lévy Process

Cummulants	
$\mu_Y = \xi_1$	$t\mu_X$
$\sigma_Y^2 = \xi_2$	$t(\sigma_X^2 + \mu_X^2\kappa)$
s	$t(3\sigma_X^2\mu_X\kappa + 2\mu_X^3\kappa^2)$
k	$t(3\sigma_X^4\kappa + 6\mu_X^4\kappa^3 + 12\sigma_X^2\mu_X^2\kappa^2)$

see [14], Section 4.4.3

3.4 Bivariate Model

We extend the univariate case ($c_j = 0$) to a bivariate case by using the scaling properties of the Inverse Gaussian distribution and the Gamma distribution for a constant $c_j > 0$, see [44].

3.4.1 NIG Case

Lemma 3.8. *Let $R_t = L_t^0 + c_j L_t^j$, $j = 1, 2$, $c_j > 0$ be a real valued process and $L_t^0 \sim IG_t(a_0, b_0)$, where $a_0 = \frac{t}{\sqrt{\kappa_0}}$, $b_0 = \frac{1}{\sqrt{\kappa_0}}$ and $c_j L_t^j \sim IG_{\sqrt{c_j}t}(\sqrt{c_j}a_j, b_j/\sqrt{c_j})$, where $a_j = \frac{t}{\sqrt{\kappa_j}}$, $b_j = \frac{1}{\sqrt{\kappa_j}}$.*

Proof. This result follows from scaling property of the Inverse Gaussian process, see [44]. \square

Proposition 3.9. Lévy Measure of Bivariate NIG Process

Let $Y_t \sim NIG(\mu_X, \sigma_X, \kappa_0, \kappa_j, c_j)$ be a subordinated Brownian motion with the subordinator $R_t = L_t^0 + c_j L_t^j$, $j = 1, 2$, with $c_j L_t^j \sim IG_{\sqrt{c_j}t}(\sqrt{c_j}a_j, b_j/\sqrt{c_j})$, $\lambda_j = 1/2c\kappa_j$, and $\gamma_j = 1/\sqrt{2\pi\kappa_j}$. $L_t^0 \sim IG_t(a_0, b_0)$, $\lambda_0 = 1/2\kappa_0$, and $\gamma_0 = 1/\sqrt{2\pi\kappa_0}$, then the Lévy measure of Y_t is given by

$$\begin{aligned} \nu_Y(x) &= e^{Ax} \left[\frac{C_0}{|x|} K_1(B_0 |x|) + \frac{C_j}{|x|} K_1(B_j |x|) \right] \\ A &= \frac{\mu_X}{\sigma_X^2} \\ B_0 &= \frac{\sqrt{\mu_X^2 + \sigma_X^2/\kappa_0}}{\sigma_X^2} \\ C_0 &= \frac{\sqrt{\mu_X^2 + \sigma_X^2/\kappa_0}}{2\pi\sigma_X\sqrt{\kappa_0}} \\ B_j &= \frac{\sqrt{\mu_X^2 + \sigma_X^2/\kappa_0}}{\sigma_X^2} \sqrt{c_j\kappa_j} \end{aligned}$$

$$C_j = \frac{\sqrt{\mu_X^2 + \sigma_X^2 / \sqrt{c_j} \kappa_j}}{2\pi \sigma_X c_j \sqrt{\kappa_j}}$$

Proof. using equation (3.9)

$$\begin{aligned} \nu_Y(x) &= \frac{\gamma_0}{\sigma_X \sqrt{2\pi}} \int_0^\infty e^{-\frac{(x-\mu_X t)^2}{2t\sigma_X^2} - \lambda t} \frac{dt}{t^2} \\ &\quad + \frac{\gamma_j}{\sigma_X \sqrt{2\pi} c_j} \int_0^\infty e^{-\frac{(x-\mu_X t)^2}{2t\sigma_X^2} - \lambda \sqrt{c_j} t} \frac{dt}{t^2} \end{aligned}$$

Using the integral representation of the modified Bessel function

$$\begin{aligned} \nu_Y(x) &= e^{\frac{\mu_X x}{\sigma_X^2}} \left[\frac{1}{2\pi \sigma_X \sqrt{\kappa_0}} \left(\frac{\sqrt{\mu_X^2 + \sigma_X^2 / \kappa_0}}{|x|} \right) K_1 \left(\frac{\sqrt{\mu_X^2 + \sigma_X^2 / \kappa_0}}{\sigma_X^2} |x| \right) \right. \\ &\quad \left. + \frac{1}{2\pi \sigma_X c_j \sqrt{\kappa_j}} \left(\frac{\sqrt{\mu_X^2 + \sigma_X^2 / \sqrt{c_j} \kappa_j}}{|x|} \right) K_1 \left(\frac{\sqrt{\mu_X^2 + \sigma_X^2 / \sqrt{c_j} \kappa_j}}{\sigma_X^2} |x| \right) \right] \end{aligned}$$

□

Proposition 3.10. Characteristic Function of Bivariate NIG Process

Let $Y_t \sim NIG(\mu_X, \sigma_X, \kappa_0, \kappa_j, c_j)$ be a subordinated Brownian motion with the subordinator $R_t = L_t^0 + c_j L_t^j$, $j = 1, 2$, with $c_j L_t^j \sim IG_{\sqrt{c_j} t}(\sqrt{c_j} a_j, b_j / \sqrt{c_j})$ and $L_t^0 \sim IG_t(a_0, b_0)$. Following Schoutens, see [44], the characteristic function of Y_t is given by

$$\phi_Y(-iv) = \mathbb{E}[\exp(i(-iv)Y_t)] = e^{-a_0 t (\sqrt{-2i(-iv) + b_0^2} - b_0)} e^{-\sqrt{c_j} a_j t (\sqrt{-2i(-iv) + b_j^2 / c_j} - b_j / \sqrt{c_j})}$$

where $a_0 = \frac{1}{\sqrt{\kappa_0}}$, $b_0 = \frac{1}{\sqrt{\kappa_0}}$, $a_j = \frac{1}{\sqrt{\kappa_j}}$, $b_j = \frac{1}{\sqrt{\kappa_j}}$, $v = iu\mu_X - \frac{1}{2}u^2\sigma_X^2$ and $c_j > 0$ is a constant.

Proof. Applying Theorem 3.1 we directly get the result. □

Remark 3.11. We can recover the characteristic exponent by $\psi_{Y_t}(u) = \log(\phi_Y(u))/t$ which gives us

$$\psi(-iv) = -a_0 (\sqrt{-2i(-iv) + b_0^2} - b_0) - \sqrt{c_j} a_j (\sqrt{-2i(-iv) + b_j^2 / c_j} - b_j / \sqrt{c_j})$$

Remark 3.12. The above theorem imposes that $\mu_Y = 0$, since and $\mu_X, \sigma_X \in \mathbb{R}$.

Table 3.3: Table of Cummulants for Bivariate IG Subordinated Lévy Process

Cummulants	
$\mu_Y^j = \xi_1$	$t\mu_X + c_j t\mu_X$
$(\sigma_Y^j)^2 = \xi_2$	$t((\kappa_0 + c_j^2 \kappa_j)\mu_X^2 + (1 + c_j)\sigma_X^2)$
s	$t(3(\kappa_0^2 + c_j^3 \kappa_j^2)\mu_X^3 + 3(\kappa_0 + c_j^2 \kappa_j)\mu_X \sigma_X^2)$
k	$t(15(\kappa_0^3 + c_j^4 \kappa_j^3)\mu_X^4 + 18(\kappa_0^2 + c_j^3 \kappa_j^2)\mu_X^2 \sigma_X^2 + 3(\kappa_0 + c_j^2 \kappa_j)\sigma_X^4)$

3.4.2 VG Case

Lemma 3.13. *Let $R_t = G_t^0 + c_j G_t^j$, $j = 1, 2$, $c_j > 0$ a constant. be a real valued process. Let $G_t^0 \sim \Gamma_t(a_0, b_0)$, where $a_0 = \frac{t}{\kappa_0}$, $b_0 = \frac{1}{\kappa_0}$ and $c_j G_t^j \sim \Gamma_t(a_j, b_j/c_j)$, where $a_j = \frac{t}{\kappa_j}$, $b_j = \frac{1}{\kappa_j}$.*

Proof. This result follows from scaling property of the Gamma process, see [44]. \square

Proposition 3.14. Lévy Measure of Bivariate VG Process

Let $Y_t \sim VG(\mu_X, \sigma_X, \kappa_0, \kappa_j, c_j)$ be a subordinated Brownian motion with the subordinator $R_t = G_t^0 + c_j G_t^j$, $j = 1, 2$, with $c_j G_t^j \sim \Gamma_t(a_j, b_j/c_j)$, $b_j = 1/\kappa_j$, and $a_j = t/\kappa_j$. $G_t^0 \sim \Gamma(a_0, b_0)$. $b_0 = 1/\kappa_0$, and $a_0 = t/\kappa_0$ then the Lévy density of Y_t is given by

$$\begin{aligned} \nu_Y(x) &= \frac{1}{\kappa |x|} e^{Ax - B_0|x|} + \frac{1}{c_j^{3/2} \kappa_j |x|} e^{Ax - B_j|x|} \\ A &= \frac{\mu_X}{\sigma_X^2} \\ B_0 &= \frac{\sqrt{\mu_X^2 + \sigma_X^2/\kappa_0}}{\sigma_X^2} \\ B_j &= \frac{\sqrt{\mu_X^2 + \sigma_X^2/c_j \kappa_j}}{\sigma_X^2} \end{aligned}$$

Proof. The proof is similar to Proposition 3.9, using equation (3.9) and completing the square and solving the integral gives us the proposition. \square

Proposition 3.15. Characteristic Function of Bivariate VG Process

Let $Y_t \sim VG(\mu_X, \sigma_X, \kappa_0, \kappa_j, c_j)$ be a subordinated Brownian motion with the subordinator $R_t = G_t^0 + c_j G_t^j$, $j = 1, 2$, with $c_j G_t^j \sim \Gamma(a_j, b_j/c_j)$ and $G_t^0 \sim \Gamma(a_0, b_0)$, Following Schoutens, see [44], the characteristic function of Y_t is given by

$$\phi_Y(-iv) = \mathbb{E}[\exp(i(-iv)Y_t)] = (1 - i(-iv)/b_0)^{-a_0 t} (1 - ic_j(-iv)/b_j)^{-a_j t}$$

where $a_0 = \frac{t}{\kappa_0}$, $b_0 = \frac{1}{\kappa_0}$, $a_j = \frac{t}{\kappa_j}$, $b_j = \frac{1}{\kappa_j}$, $v = iu\mu_X - \frac{1}{2}u^2\sigma_X^2$ and $c_j > 0$ is a constant.

Proof. Applying Theorem 3.1 we directly get the result. \square

Remark 3.16. We can recover the characteristic exponent by $\psi_{Y_t}(u) = \log(\phi_Y(u))/t$ which gives us

$$\psi(-iv) = -a_0 \log(1 - i(-iv)/b_0) - a_j \log(1 - ic_j(-iv)/b_j)$$

Remark 3.17. The above proposition imposes that $\mu_Y = 0$, since $\mu_X, \sigma_X \in \mathbb{R}$.

Table 3.4: Table of Cummulants for Bivariate Gamma Subordinated Lévy Process

Cummulants	
$\mu_Y^j = \xi_1$	$t\mu_X + c_j t\mu_X$
$(\sigma_Y^j)^2 = \xi_2$	$t((\kappa_0 + c_j^2 \kappa_j)\mu_X^2 + (1 + c_j)\sigma_X^2)$
s	$t(2(\kappa_0^2 + c_j^3 \kappa_j^2)\mu_X^3 + 3(\kappa_0 + c_j^2 \kappa_j)\mu_X \sigma_X^2)$
k	$t(6(\kappa_0^3 + c_j^4 \kappa_j^3)\mu_X^4 + 12(\kappa_0^2 + c_j^3 \kappa_j^2)\mu_X^2 \sigma_X^2 + 3(\kappa_0 + c_j^2 \kappa_j)\sigma_X^4)$

Chapter 4

Asian Option Pricing

4.1 Introduction

How can we determine that a contract is fairly priced? This is one of the main questions in asset pricing and many academics have approached this problem from a mathematical and financial point of view. Literature regarding this topic is vast and carries many different opinions on how to value assets correctly. For interested readers we refer to these texts, [14], and [40], which show pricing from both a stochastic and a partial differential equations point of view. One of the most common methods to compute prices is to use the Monte Carlo method. Monte Carlo methods take advantage of the Law of Large Numbers to converge to a mean and output an expected price.

Alternatively, the relation between stochastic differential equations (SDE) and partial differential equations (PDE) through the Feynman-Kac representation allows us to use numerical techniques to solve pricing problems. Many different techniques have been developed to solve these PDEs, such as the method of Carr and Madan, which uses a damped Fourier series expansion to solve the PDE, [14], finite difference methods and a Fourier-Cosine expansion method developed by Fang and Oosterlee, [22]. It is well known that finite difference methods have issues in pricing high dimensional portfolios. The Fourier-Cosine method was extended to the bivariate case by Ruijter and Oosterlee, see [42] and the Asian Fourier-Cosine Method was developed by Zhang and Oosterlee, see [47]. The method was extended to three dimensions and applied to pricing and hedging multi-asset spread contracts by Pellegrino and Piergiacomo, see [38]. We present here an implementation of the Asian Fourier-Cosine method under one-dimensional time-changed models and compare with MC techniques. A two-dimensional time-changed model is presented and simulated using MC techniques.

4.2 Overview of Asian Options

Asian options were first introduced in Tokyo, 1987, for oil commodities. They are classified as path dependent options as their payoff is typically calculated as the geometric or arithmetic average of the underlying asset price at maturity [47]. These option styles are known in a broader scope as exotic options. They are usually traded over the counter (OTC) and are often illiquid. The path dependence of Asian options presents a challenge in pricing and hedging, which we try to address in this work.

Definition 4.1. Asian Option

Let $\{S_t^j\} \in \mathbb{R} \times [0, T]$ be the real valued matrix of risk neutral asset price processes, where $t = 0, 1, \dots, M$, $j = 1, 2$ and M is the sampling frequency. The geometric average Asian option takes the geometric average of the price process of each asset. The geometric average is given by

$$\bar{S}_t^G = \left(\prod_{n=0}^M S_n^j \right)^{\frac{1}{M+1}} \quad (4.1)$$

Similarly, the arithmetic average can be used and is given by

$$\bar{S}_t^A = \frac{1}{M+1} \sum_{n=0}^M S_n^j \quad (4.2)$$

Consider an Asian option maturing at time T based on \bar{S}_t with an F_T -measurable payoff H . Then, the payoff of the European style Asian call option is given by:

$$H(t, S_t)^k = [\bar{S}_t^{G,A} - K^j]^+ \quad (4.3)$$

where $(S, 0)^+ = \max(S, 0)$, see [44].

4.2.1 Asian Basket Spread Options

Definition 4.2. Asian Basket Spread Options

Let S_{t_1} be the asset price process of a basket asset $j = 1$ and S_{t_2} be the asset price process of a basket of assets $j = 2$ then payoff of the Asian spread option is given by the following:

$$H(S_{t_1}, S_{t_2}, t) = [(w_1 \bar{S}_t^1 - w_2 \bar{S}_t^2) - K]^+ \quad (4.4)$$

where \bar{S}_t^j , $j = 1, 2$ is the arithmetic averages of the respective assets, see [17] and w_j is the weight of each asset such that $\sum_{j=1}^2 w_j = 1$.

4.3 Monte Carlo Pricing

Monte Carlo (MC) method is a statistical method that directly applies the Central Limit Theorem by simulating $n \in \mathbb{N}$ number of trajectories, that follow a prescribed distribution in an independent way. Random number generators (RNG) are used to sample n independent random numbers from a known distributions and relies on the Law of Large Numbers to converge to a mean, see [41]. In our work we assume we are working in a risk-neutral measure \mathbb{Q} , to compute no arbitrage prices of Asian options. We employ efficient algorithms to reduce computation time of MC simulations by utilizing matrix operations on large generated data.

Proposition 4.3. Risk-Neutral Pricing

Let $S_t(Y_t)$ be the price process of the underlying asset driven by the Lévy process Y_t and let $H(t, Y_t)$ be the payoff of the Asian option under \mathbb{P} , then the risk neutral price of the option under the risk neutral probability \mathbb{Q} is given by

$$\mathbb{E}^{\mathbb{Q}}[H(t, Y_t)] = e^{-r(T-t)} \mathbb{E}[H(t, Y_t)] \quad (4.5)$$

where T , is the maturity time.

Proof. see [18], Theorem 4.2.1 □

4.3.1 Example of MC Simulation of Geometric Brownian Motion

The Monte Carlo algorithm outlined by Cont, see [14] is presented below.

Algorithm 1: Geometric Brownian Motion

```

while  $t \leq T$  do
   $dt = 1/250$ 
  Generate  $Z \sim N(0, 1)$ 
   $\Delta Y = \sigma \sqrt{dt} Z + \mu dt$ 
   $Y = \sum_{i=0}^N \Delta Y$ 
  update  $i$ 
  update  $t$ 
end
 $S = S_0 \exp(Y)$ 

```

Algorithm 1 simulates the geometric Brownian motion. It assumes that each increment is independent identically distributed (i.i.d) and draws random numbers from the standard normal distribution.

Algorithm 2: Asian Option Pricing

```

while  $i \leq N$  do
  |  $H(S(i), T) = \exp(-rT) \max(\text{mean}(\hat{S}_T) - K, 0)$ 
  | update  $i$ 
end

```

Where $N = T/dt$ for stopping time T and dt is the sampling frequency i.e. daily sampling $dt = 1/250$. Algorithm 2 gives us the discounted price of the Asian option by taking the average of the trajectories generated from the geometric Brownian motion in Algorithm 1.

4.4 Fourier-Cosine (COS) Method

The Fourier-Cosine (COS) method is a transformation type numerical method developed by Fang and Oosterlee, see [22]. The method is extended to the bivariate case by Ruijter and Oosterlee, see [42]. The Asian Fourier-Cosine method (ASCOS) is an extension of the COS method developed by Zhang and Oosterlee, see [47]. The method is extended to three independent random variables and applied to pricing and hedging multi-asset spread contracts by Pellegrino and Piergiacomo, see [38].

The COS method calculates an approximation to the value of the option by discounting the risk neutral price. It is used to approximate the transitional probability distribution using the transitional characteristic function on a truncated interval $[a, b] \in \mathbb{R}^2$. Under the risk neutral probability measure \mathbb{Q} , we denote the independent Lévy process as Y_t and risk neutral price process as S_t , where $Y_t = \log(S_t)$ and $Z_t = \log(X_t)$ such that $H(t, z) = (z - K)^+$. For example, for the geometric Asian option $X_t = \left(\prod_{t=0}^M S_t\right)^{\frac{1}{M+1}}$. We can approximation of the conditional probability density of Z_t by

$$\begin{aligned}
 f(z|y) &= \frac{2}{b-a} \int_a^b \text{Re}[\varphi_{Z_T}\left(\frac{k\pi}{b-a}|y\right) \exp(-ik\pi \frac{a}{b-a})] \cos(k\pi \frac{z-a}{b-a}) dz \\
 f(z|y) &\approx \frac{2}{b-a} \sum_{k=0}^{N-1} \text{Re}[\varphi_{Z_T}\left(\frac{k\pi}{b-a}|y\right) \exp(-ik\pi \frac{a}{b-a})] \cos(k\pi \frac{z-a}{b-a})
 \end{aligned} \tag{4.6}$$

similarly we can approximate the non conditional probability density for Y_t by

$$\begin{aligned}
 f(y) &= \frac{2}{b-a} \int_a^b \text{Re}[\phi_{Y_T}\left(\frac{k\pi}{b-a}\right) \exp(-ik\pi \frac{a}{b-a})] \cos(k\pi \frac{y-a}{b-a}) dy \\
 f(y) &\approx \frac{2}{b-a} \sum_{k=0}^{N-1} \text{Re}[\phi_{Y_T}\left(\frac{k\pi}{b-a}|y\right) \exp(-ik\pi \frac{a}{b-a})] \cos(k\pi \frac{y-a}{b-a})
 \end{aligned} \tag{4.7}$$

where $\text{Re}[\cdot]$ is the real argument of the input, $\varphi(u|y)$ is the conditional characteristic function of the independent Lévy process, \sum' is truncated a sum where the first term is truncated by $\frac{1}{2}$

and $k = 0, 1, \dots, N - 1$. The shifted characteristic function $\varphi(u|y)$ is given by

$$\varphi_{Z_t}(u|y) = \exp(iuy)\phi_{Y_t}(u) \quad (4.8)$$

Where ϕ_{Y_t} is the characteristic function given for each process Y_t^j . The option price is approximated by

$$H(t, y) \approx \exp[-r(T-t)] \sum_{k=0}^{N-1} \text{Re}[\varphi_{Z_T}(\frac{k\pi}{b-a}|y) \exp(-ik\pi \frac{a}{b-a})] V_k \quad (4.9)$$

where V_k is given by

$$V_k = \frac{2}{b-a} \int_a^b H(t, X_t) \cos(k\pi \frac{x-a}{b-a}) dx \quad (4.10)$$

see [22] and [42].

4.4.1 Asian Fourier-Cosine (ASCOS) Method

The Asian Fourier-Cosine method was developed by Zhang and Oosterlee is an extension of the COS method to price Asian options, see [47]. Again, we fix the complete probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \in [0, T]}, \mathbb{P})$, and \mathbb{Q} is the risk neutral probability measure then the price process has the dynamics given by

$$S_t = S_0 \exp(Y_t) \quad (4.11)$$

Where Y_t is a Lévy process.

Arithmetic Average Asian Option

We let the log price of the asset be $Y_t = \log(S_t)$, $t = 0, \dots, M$ and we the log return be defined as $R_t = \log(\frac{S_{t+1}}{S_t})$, then if we let $Z_1 = R_M$ we get that for $t = 2, \dots, M$ we have that

$$\begin{aligned} Z_t &= \log\left(\frac{S_{M-t+1}}{S_{M-t}} + \frac{S_{M-t+2}}{S_{M-t}} + \dots + \frac{S_M}{S_{M-t}}\right) \\ &= R_{M+1-t} + \log(1 + \exp(Z_{t-1})) = R_{M+1-t} + W_{t-1} \end{aligned} \quad (4.12)$$

from this Zhang and Oosterlee derived the characteristic function of X_t and approximates the characteristic function using the Clenshaw-Curtis Quadrature, see [47]. The characteristic function of Z_t is given by

$$\phi_{W_{t-1}} = \int_{-\infty}^{\infty} (e^y + 1)^{iu} f_{Z_{t-1}}(y) dy \quad (4.13)$$

and is approximated by

$$\phi_{W_{t-1}}(u, dt) \approx \frac{2}{b-a} \text{Re}[\phi_{Z_{t-1}}(\frac{l\pi}{b-a}|dt) e^{\frac{-il\pi a}{b-a}}] \int_a^b (e^y + 1)^{iu} \cos(l\pi \frac{y-a}{b-a}) dy \quad (4.14)$$

the Clenshaw-Curtis quadrature, see [13], is used to approximate

$$\mathcal{Q}(k, l) = \int_a^b (e^y + 1)^{\frac{ik\pi}{b-a}} \cos(l\pi \frac{y-a}{b-a}) dy \quad (4.15)$$

This gives us the recursive approximation of $\phi_{W_{t-1}}$ as

$$\phi_{W_{t-1}}(u, dt) \approx \frac{2}{b-a} \text{Re}[\phi_{Z_{t-1}}(\frac{l\pi}{b-a} | dt) \exp(\frac{-il\pi a}{b-a})] \mathcal{Q}(k, l) \quad (4.16)$$

where $l, k = [1, 2, \dots, 256]$. Substituting in Eq.(4.12) we get that

$$\frac{1}{M+1} \sum_{t=0}^M S_t = \frac{(1 + e^{Z_M}) S_0}{M+1} \quad (4.17)$$

and the payoff of the arithmetic average is given by

$$V_k^{call} = \left(\frac{(1 + e^z) S_0}{M+1} - K \right)^+ \quad (4.18)$$

$$V_k^{put} = \left(K - \frac{(1 + e^z) S_0}{M+1} \right)^+ \quad (4.19)$$

then,

$$V_{k_{call}} = \frac{2}{b-a} \left[\frac{S_0}{M+1} (\eta_{call} + \chi_{call}) - K \eta_{call} \right] \quad (4.20)$$

$$V_{k_{put}} = \frac{2}{b-a} \left[K \eta_{put} - \frac{S_0}{M+1} (\eta_{put} + \chi_{put}) \right] \quad (4.21)$$

$$\chi_{call}(y^*, b) = \int_{y^*}^b \exp(z) \cos(k\pi \frac{z-a}{b-a}) dz \quad (4.22)$$

$$\eta_{call}(y^*, b) = \int_{y^*}^b \cos(k\pi \frac{z-a}{b-a}) dz \quad (4.23)$$

$$\chi_{put}(a, y^*) = \int_a^{y^*} \exp(z) \cos(k\pi \frac{z-a}{b-a}) dz \quad (4.24)$$

$$\eta_{put}(a, y^*) = \int_a^{y^*} \cos(k\pi \frac{z-a}{b-a}) dz \quad (4.25)$$

where $y^* = \log(\frac{K(M+1)}{S_0} - 1)$ is the execution point bounded by $[a, b]$, i.e. $a \leq y^* \leq b$. The integration range $[a, b]$ is given by

$$[a, b] = [\min(\log(M) + \xi_1 - L\sqrt{M\xi_2}), \max(\log(M) + \xi_1 + L\sqrt{M\xi_2})] \quad (4.26)$$

the detailed derivation can be seen in [47].

4.5 Simulations

To use the Monte Carlo method to simulate Inverse Gaussian (IG) subordinated processes, we first needed to generate IG random numbers. IG random numbers were generated using Michael, Schucany and Hass' algorithm as shown in [35] and is outlined in Algorithm 3.

Similarly, Gamma random numbers were generated using Jonks algorithm as shown in [44] and is outlined in Algorithm 4. Using the generated numbers we simulate the subordinator process via Algorithm 5, then the underlying price is simulated from Algorithm 1, where $dt = dR = R(t + dt) - R(t)$.

Algorithm 3: IG Random Number Generator

a, b are inputs representing the scale and shape parameter

Generate $V \sim N(0, 1)^2$

$$W = aV$$

$$C = \frac{a}{2b}$$

$$X = a + C(W - \sqrt{W(4b + W)})$$

Generate $Y \sim U(0, 1)$

$$Z = \frac{a}{a+1}$$

if $Y \geq Z$ **then**

$$| \quad X = \frac{a^2}{X}$$

else

$$| \quad X = X$$

end

Algorithm 4: Gamma Random Number Generator

Generate $Y \sim U(0, 1)$

$Z = \frac{a}{a+1}$, a is the input representing the scale parameter

while $X + Y > 1$ **do**

$$| \quad U \sim U(0, 1)$$

$$| \quad V \sim U(0, 1)$$

$$| \quad X = U^{\frac{1}{a}}$$

$$| \quad Y = V^{\frac{1}{1-a}}$$

$$| \quad X = X$$

end

$$W \sim U(0, 1)$$

$$Z = -\log(W)$$

$$X = \frac{ZX}{X+Y}$$

Algorithm 5: Subordinator Process Simulation

```

while  $t \leq T$  do
  Generate  $X \sim IG(a, b)$  or  $X \sim \Gamma(a)$ 
  Generate  $Y \sim IG(a, b)$  or  $Y \sim \Gamma(a)$ 
   $L_0 = L_0 + X$ 
   $L_j = L_j + Y$ 
   $R = L_0 + c_j L_j$ 
  update  $t$ 
end

```

4.5.1 Univariate Simulations Results

Through numerical simulation we first study the univariate model. The purpose of this section is to study the validity of subordinated Lévy processes in capturing jumps in asset prices and study the stability of the numerical methods. We also present some observations about the ASCOS method compared and Monte Carlo method. The parameters were initially chosen heuristically based on the parameters presented in literature. We also present some calibrated parameters in Chapter 6.

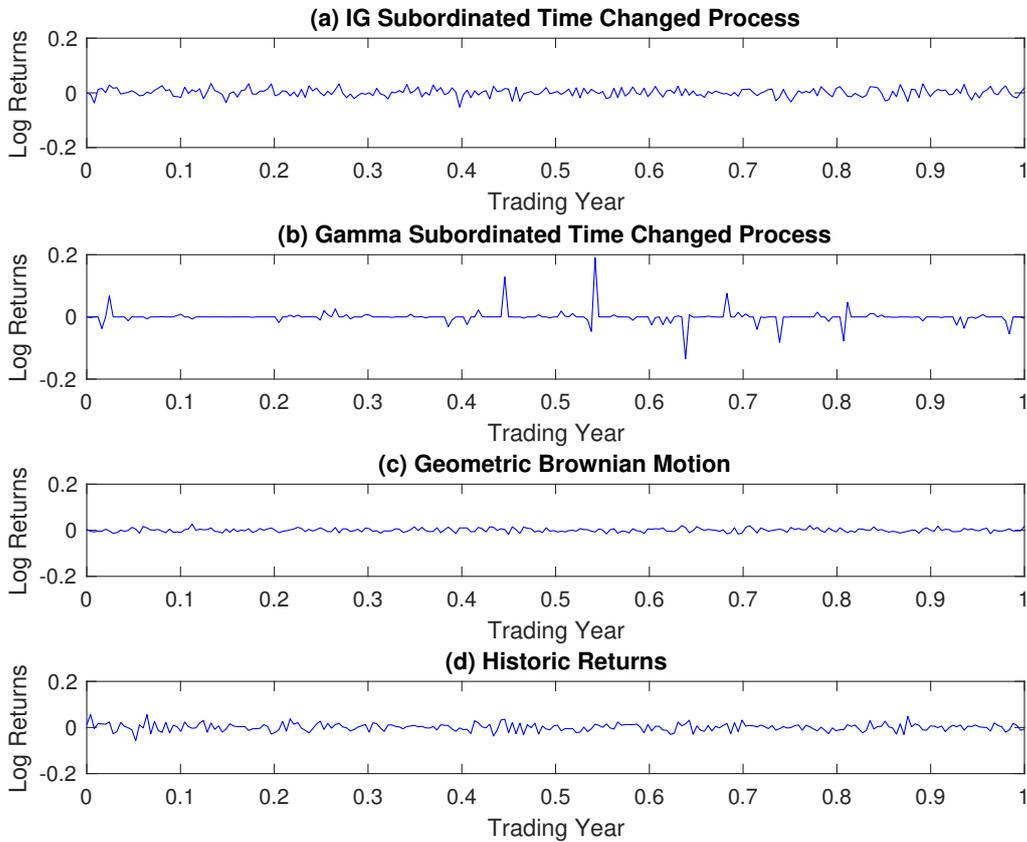


Figure 4.1: The log return processes generated by (a) IG subordinated BM (b) Gamma subordinated BM and (c) GBM, using parameters $S_0 = 57$, $K = 52$, $a = 1/250\kappa$, $b = 1/\kappa$, $\mu = 0$, $\sigma = 0.02$, $\kappa = 0.02$. (d) is the historic log return of observed WTI Crude prices from 2014.

To illustrate the characteristics of the time changed model we look at Figure 4.1, which shows a process generated by (a) the IG subordinated Brownian Motion, (b) the Gamma subordinated Brownian Motion and (c) the Geometric Brownian Motion (GBM). It is clear from Figure 4.1 that Fig.4.1(a) and Fig.4.1(b) show much more volatility than Fig.4.1(c) for the same average result. The IG subordinated process captures small jump movements and does not generate large jumps. Whereas the Gamma subordinated process, lacks the fine details of the IG subordinated process but produces severe jumps. Fig.4.1(d) shows a one year historic log return of observed WTI crude oil prices from 2014. It is clear that the observed historic data the NIG process emulates Oil returns the closest.

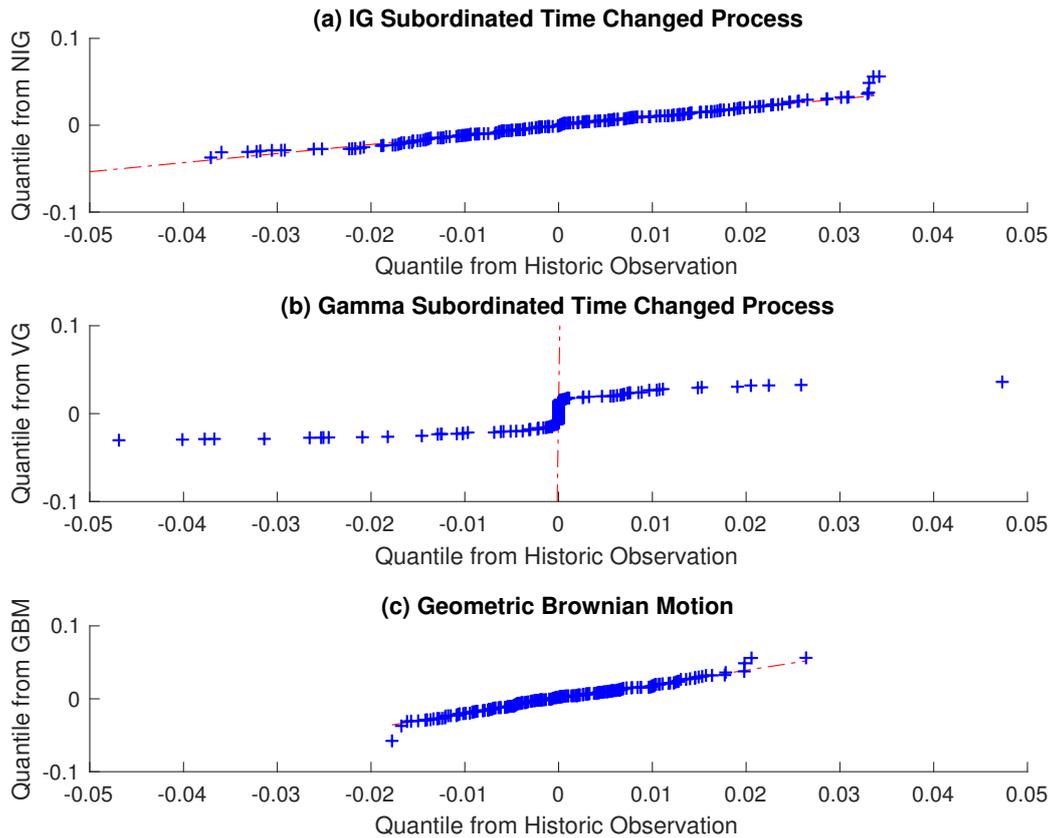


Figure 4.2: The QQ-plot from (a) IG subordinated process quantile, (b) Gamma subordinated process quantile, and (c) GBM quantile matched to the quantile of historic log return of observed WTI Crude prices from 2014 from sampled numbers.

Fig.4.2(a) shows us that random numbers sampled from the IG subordinated process is the best fit to the historic data as it lies mostly on the normal line generated by the quantiles from

random numbers sampled from the IG subordinated process and the historical data. Fig.4.2(b) clearly shows that the Gamma subordinated process is not a good fit to the historical data, as there are extreme values that lie off the normal line. Fig.4.2(c) is a closer fit than Fig.4.2(b) but it still does not fit the data as well as Fig.4.2(a), as the lower tail of the distribution lies off the normal line.

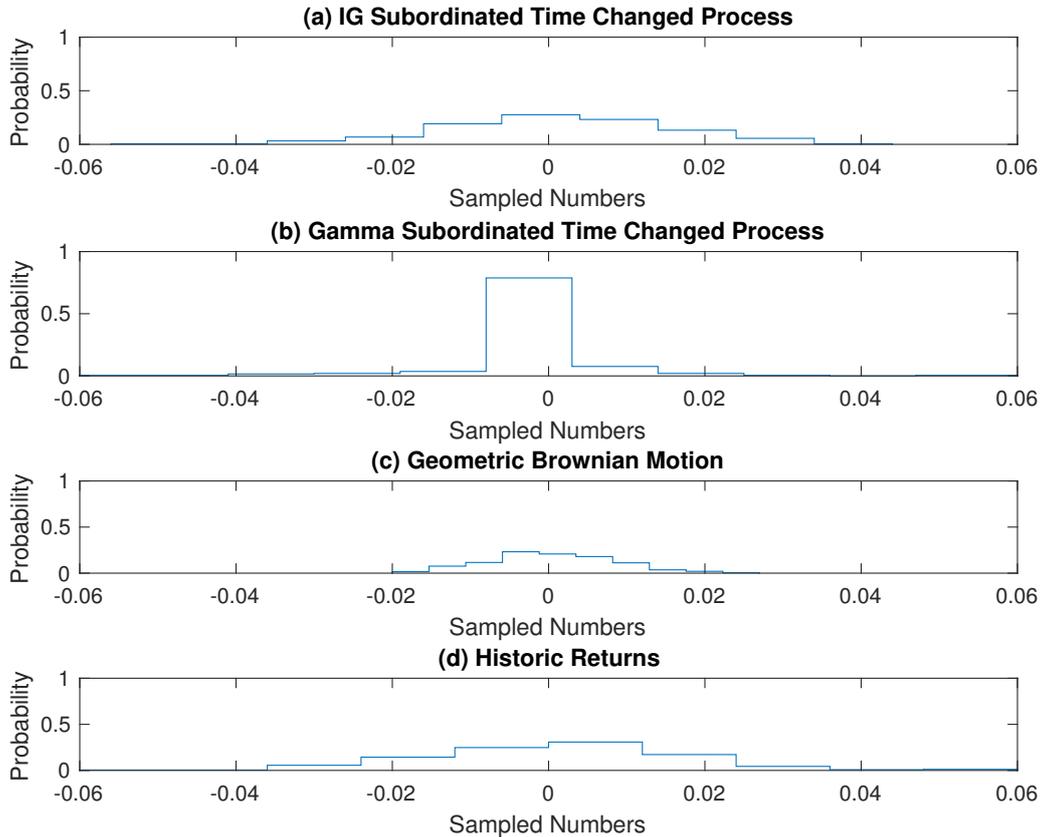


Figure 4.3: The distribution curves of (a) IG subordinated process, (b) Gamma subordinated process, (c) GBM and (d) the historic log return of observed WTI Crude prices from 2014 from sampled numbers.

Next, we plot the numbers sampled from the path to see their underlying distribution as shown in Figure 4.3. Comparing Fig.4.3(a) to Fig.4.3(c) we can clearly see that Fig.4.3(a) generates numbers in outside the tails of Fig.4.3(c) and is mesokurtic as you'd expect from a heavy tailed distribution. Comparing Fig.4.3(b) to Fig.4.3(c) we also observe that Fig.4.3(b) generates numbers outside the tails of Fig.4.3(c) but displays leptokurtic behavior. This is to be expected from Figure 4.1, as Fig.4.1(b) exhibits large jumps at a modest rate. Fig.4.3(d) shows us the distribution from the historic observation. Visually, it supports the argument that the NIG process is the distribution that closely emulates the historic characteristics.

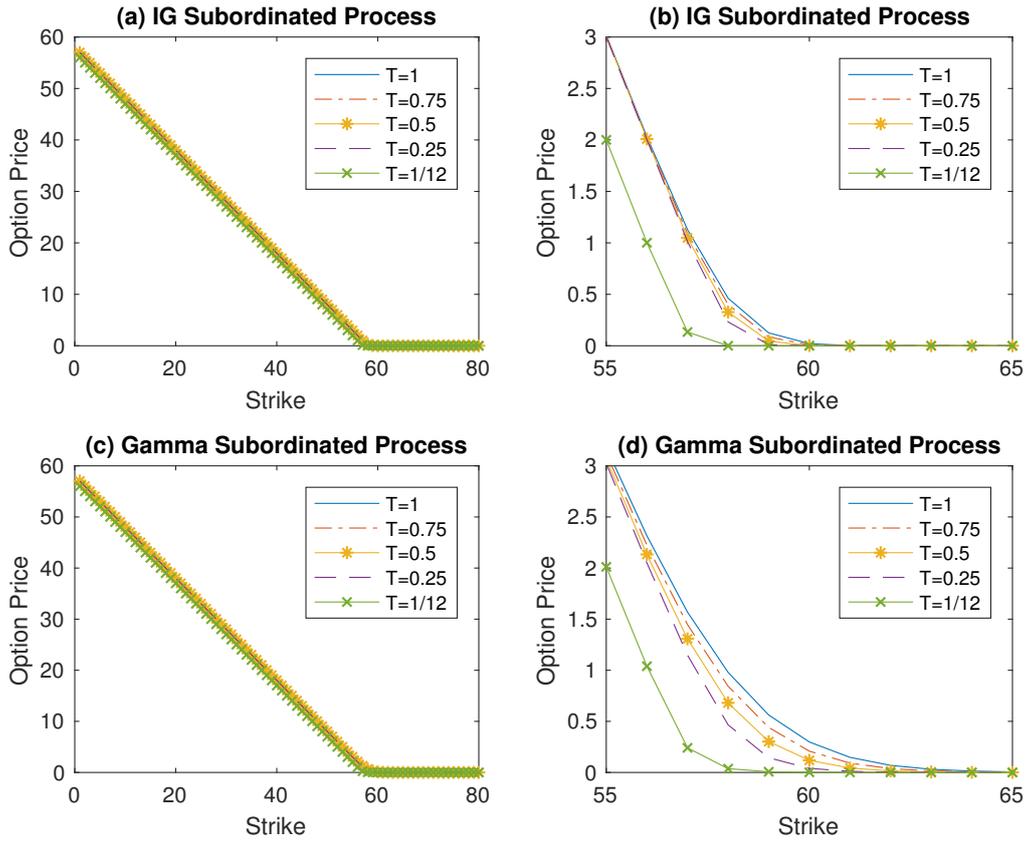


Figure 4.4: Monte Carlo pricing using parameters $S_0 = 57$, $a = 1/250\kappa$, $b = 1/\kappa$, $\mu = 0$, $\sigma = 0.02$, $\kappa = 0.02$.

In Figure 4.4 we plot the sensitivity of the MC simulated option price with respect to the strike price. Fig.4.4(b) and Fig.4.4(d) are magnifications of Fig.4.4(a) and Fig.4.4(c), respectively. We observe that as the maturity date decreases, the at-the-money strike price shifts towards the initial price. This is attributed to the exponential nature of the discount factor. MC simulations smooth out the payoff of options because of averaging.

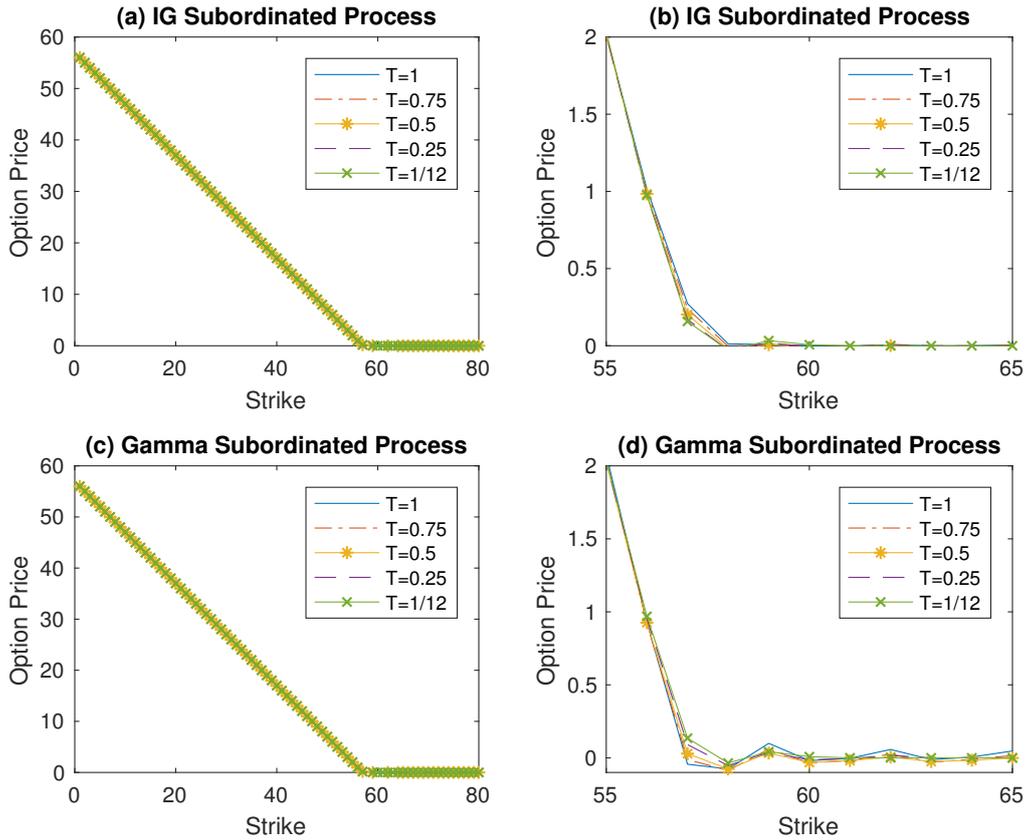


Figure 4.5: ASCOS pricing using parameters $S_0 = 57$, $a = 1/250\kappa$, $b = 1/\kappa$, $\mu = 0$, $\sigma = 0.02$, $\kappa = 0.02$.

The sensitivity of the ASCOS option price with respect to the strike price is shown in Figure 4.5. Fig.4.5(b) and Fig.4.5(d) are magnifications of Fig.4.5(a) and Fig.4.5(c), respectively. As the maturity time increases the payoff curve does not show the monotonic behavior shown in the MC simulation. The periodic behaviour of the method is due to the sinusoidal characteristic of the Fourier-Cosine expansion. The method also exhibits instability at a strike price of 0, this happens because the y^* shown in Equations (4.19)-(4.25) is an imaginary value and does not exist within the range of integration.

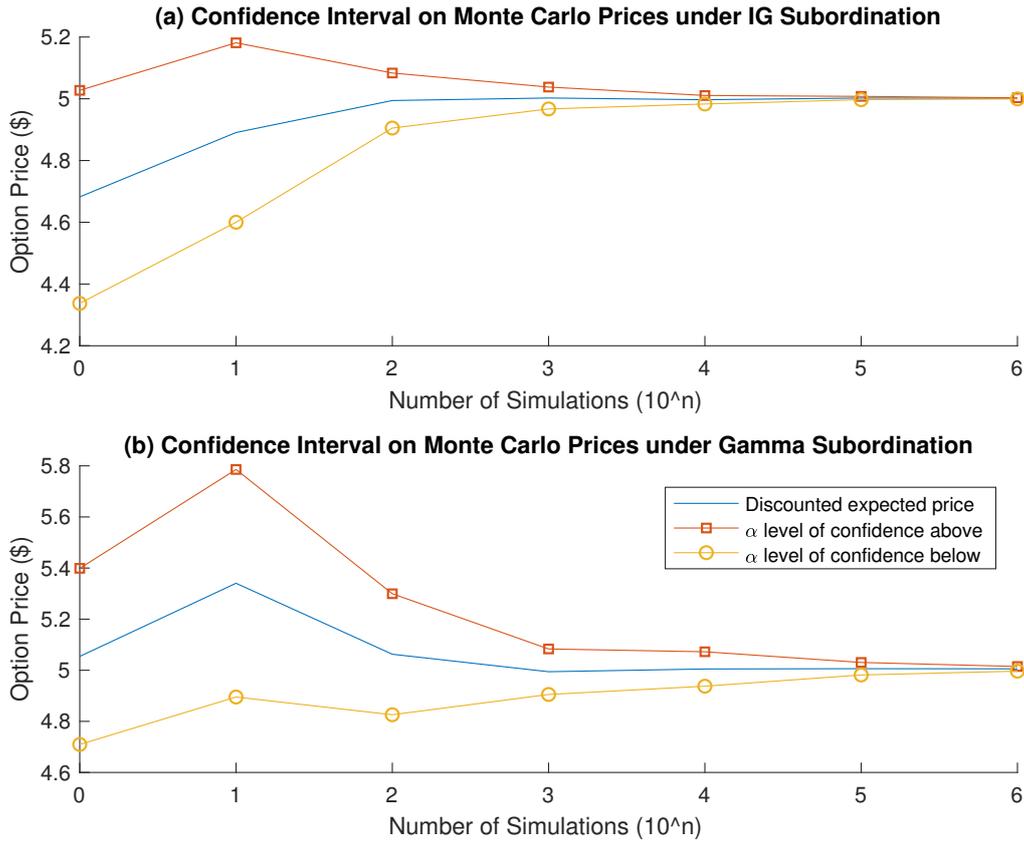


Figure 4.6: Monte Carlo pricing with confidence interval at $\alpha = 0.95$ shows the convergence of the option prices by the law of large numbers.

Table 4.1: Summary of Monte Carlo pricing of Asian options by IG subordinated processes, $K = \$52$, $S_0 = \$57$, $dt = 1/250$, $T = 1/12$

Option Price (\$)	Confidence Interval ($\alpha = 0.95$)	Number of Simulations (10^n)	Computation Time (s)
4.6823	± 0.3450	$n = 0$	0.026497
4.8907	± 0.2906	$n = 1$	0.053204
4.9943	± 0.0891	$n = 2$	0.048611
5.0025	± 0.0304	$n = 3$	0.151043
4.9969	± 0.0138	$n = 4$	0.738055
5.0024	± 0.0054	$n = 5$	8.5354010
5.0013	± 0.0018	$n = 6$	221.478561

Table 4.2: Summary of Monte Carlo pricing of Asian options by Gamma subordinated processes, $K = \$52$, $S_0 = \$57$, $dt = 1/250$, $T = 1/12$

Option Price (\$)	Confidence Interval ($\alpha = 0.95$)	Number of Simulations (10^n)	Computation Time (s)
5.0542	± 0.3447	$n = 0$	0.026497
5.3405	± 0.4453	$n = 1$	0.053204
5.0627	± 0.2368	$n = 2$	0.048611
4.9910	± 0.1626	$n = 3$	0.151043
5.0050	± 0.0674	$n = 4$	0.738055
5.0061	± 0.0243	$n = 5$	8.5354010
5.0056	± 0.0091	$n = 6$	221.478561

In Figure 4.6 we show that for Monte Carlo simulations the convergence of errors at a confidence level of 0.95. As the number of simulation increases the error bounds decrease. This experiment shows that a tight boundary of prices is achieved at $n \geq 5$. We present a summary of computed option prices up to the nearest percent in Table 4.1. The numbers are generated by the Michael, Schucany and Hass algorithm. To achieve the required accuracy using an efficient MC method we require 221.48 seconds, while Using the Fourier cosine expansion for IG subordinated processes we get an option price of \$5.0119 with a computation time of 3.04554 seconds. Table 4.2 is a summary of computed option prices up to the nearest cent with numbers generated by Jonk's algorithm. To achieve the required accuracy using an efficient MC method we require 221.48 seconds, using the Fourier cosine expansion for Gamma subordinated processes we get an option price of \$5.0141 with a computation time of 2.998690 seconds.

4.5.2 Bivariate Simulations Results

Through numerical simulations we study the impact of the coefficient c in the bivariate model given by Eq.(3.2) and (3.3). We use the calibrated parameters in Table 6.1 to price the risk-neutral Asian basket spread options and make some observations.

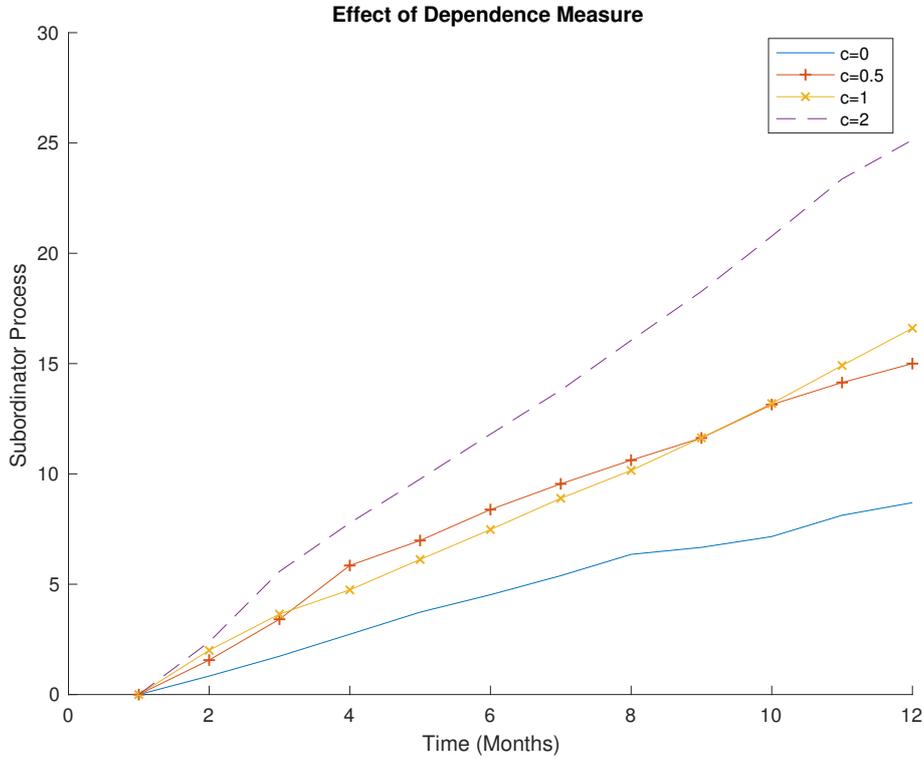


Figure 4.7: Simulation of dependent subordinator for $c = \{0, 0.5, 1\}$ values

We show that as c increases the slope of the subordinated process gets steeper as shown in Figure 4.7. The slope of the line will be given by $m = \frac{dR_t}{dt} = \frac{L_{0,dt} + cL_{dt}}{dt}$. We can immediately see that for $c > 0$ the slope is proportional to c in a linear way and this implies that the stochastic time change occurs at faster rates, adding to the volatility in the simulated price process.

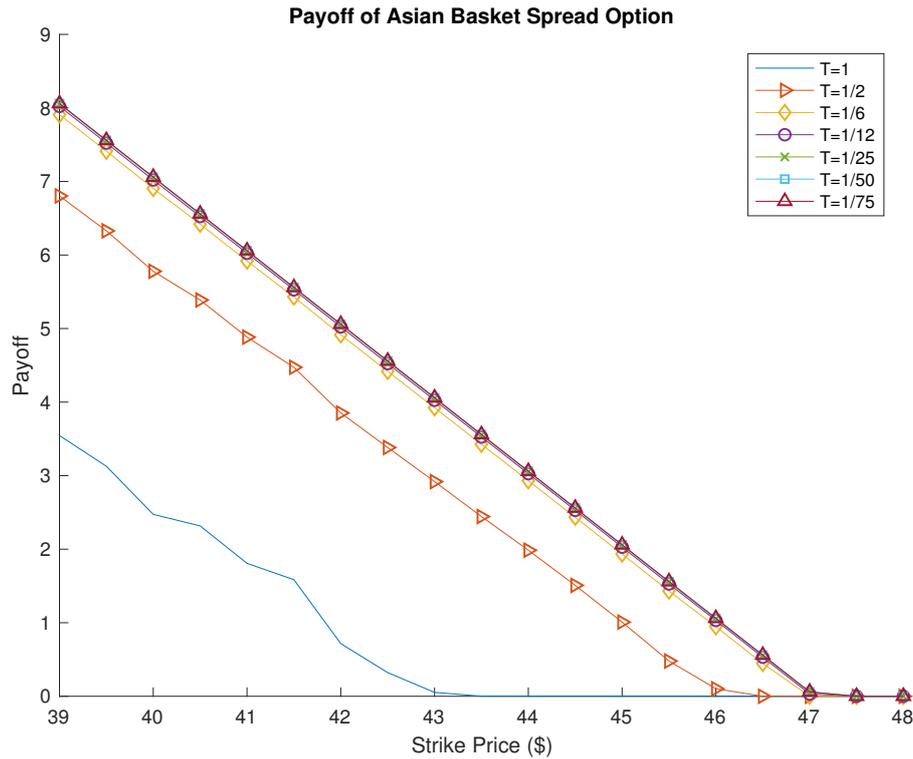


Figure 4.8: Monte Carlo pricing using the bivariate model with parameters from calibration, the asset weights were set as $w_1 = 0.8$ and $w_2 = 0.2$.

In Figure 4.8 we plot the sensitivity of the MC simulated Asian basket spread option price with respect to the strike price. As the maturity time increases the payoff curve moves away from the strike axis around the at-the-money point. We also observe the same monotonic behavior observed in the payoff curve in the univariate case. Prices shift right as the maturity times increase, this shows us that the spread is actually increasing as maturity increases. We do not use the Fourier-Cosine method to price the bivariate model. This is because the dependence structure between the two assets are not known. However, this dependence structure can be estimated by Lévy copulas.

Chapter 5

Semi-Static Hedging Using Fourier Cosine Expansion

5.1 Introduction

Hedging strategies are used by investors and writers of financial contracts to minimize the loss on returns due to exposure to market risk. These risks are mitigated by creating a replicating portfolio that replicates the payoff of the option. In this work we assume that the investor only re-balances (hedges) their portfolio outside some time interval $[t, T]$, this is called semi-static hedging. Kolkiewicz and Liu, [27], argue that continuous dynamic trading is only beneficial in a complete market setting with perfect replication. They propose that static hedging, where a hedging portfolio is created at $t = 0$ and held to maturity $t = T$, is more advantageous since it reduces transaction costs. Kolkiewicz and Liu, show that for path dependent derivatives, the semi-static hedging strategy can be applied at $t = T$. We follow in Kolkiewicz and Liu's reasoning and look at the semi-static hedging problem.

The semi-static hedging strategies described in this chapter are based on the strategies outlined in Föllmer and Schweizer, [24], and Föllmer and Leukert, [23], respectively. Kolkiewicz and Liu, [27], apply Föllmer and Schweizers results to hedging with path dependent options, see [27]. Alonso-García et al, [2], tried to use the COS-method to delta hedging, they tried to extend the framework to quadratic risk and quantile risk, but did not show any numerical evidence.

We replicate the payoff of an Asian options given by equation (4.9) with the predictable wealth process

$$X_t = X_0 + \int_0^T \beta r dt + \int_0^T \alpha dS_t$$

To follow the notation of Kolkiewicz and Liu, see [27], we let $\alpha = \pi_s X_t$ and $\beta = X_0 - S_0 \alpha$. Where π_s is the fraction of wealth invested in equity that is predictable for $t \in [0, T]$ and $\pi_p = 1 - \pi_s$ is the fraction of wealth invested in the cash account, and r is the constant interest rate.

The price dynamic of the underlying asset is a discounted cádlág process given by $S_t = S_0 e^{Y_t}$; where Y_t is given by Eq.(3.2) and (3.3). In our work we prescribe two subordinating distributions; the IG subordinator and the Gamma subordinator.

5.2 Quadratic Hedging

For payoff function H_t , and wealth process X_t we can define the quadratic hedging error as

$$J = \mathbb{E}^{\mathbb{Q}}[|X_t - H_t|^2] \quad (5.1)$$

with maturity, $t = T$. Assuming that $X_0 = e^{-r(T-t)} \mathbb{E}[H_T] = H_0$, then the cost function J is given by

$$\begin{aligned} J &= \mathbb{E}^{\mathbb{Q}}[H_0^2 + 2H_0 S_T \alpha - 2H_0 H_T + S_T^2 \alpha^2 - 2H_T S_T \alpha + H_T^2] \\ &= H_0^2 + 2H_0 \alpha \mathbb{E}^{\mathbb{Q}}[S_T] - 2H_0 \mathbb{E}^{\mathbb{Q}}[H_T] + \alpha^2 \mathbb{E}^{\mathbb{Q}}[S_T^2] - 2\alpha \mathbb{E}^{\mathbb{Q}}[H_T S_T] + \mathbb{E}^{\mathbb{Q}}[H_T^2] \end{aligned}$$

Then taking the derivative with respect to α , setting it to zero and solving for α^* we get

$$\alpha^* = \frac{\text{Cov}[H_T, S_T]}{\text{Var}[S_T]} \quad (5.2)$$

this is the same result given as shown in Kolkiewicz and Liu, see [27], it interesting to note that the optimal hedging portfolio is the ratio $\sigma_{H_t}/\sigma_{S_t}$, which gives a relative measure of risk in quadratic variation.

5.2.1 Simulation of Quadratic Hedging

We look at simulating the quadratic hedging loss using MC simulation and the ASCOS method. To hedge the option by MC simulation, first we price the option using MC simulation, then we use Eq.(5.2) to determine the optimal stock portfolio α^* , then we find $\pi_s^* = \alpha^*/X_T$ and $\pi_p^* = 1 - \pi_s^*$.

Similarly, to hedge the option by COS method, we get the option payoff process from

$$H_t \approx \exp[-r(T-t)] \text{Re}[\varphi_{Z_T}(\frac{k\pi}{b-a}|y) \exp(-ik\pi \frac{a}{b-a})] V_k$$

for $k = [0, \dots, N - 1]$. We next generate the price process from

$$S_t = S_0 e^{Y_t} \\ \approx \frac{2}{b-a} \operatorname{Re} \left[\phi_Y \left(\frac{k\pi}{b-a} \right) \exp \left(-ik\pi \frac{a}{b-a} \right) \right] V_k$$

where V_k is given by

$$V_k = S_0 \frac{2}{b-a} \int_a^b e^x \cos \left(k\pi \frac{x-a}{b-a} \right) dx$$

where ϕ_Y is the characteristic function of Y_t . Then using Eq.(5.2) we find α^* which is used to find $\pi_s^* = \alpha^*/X_T$ and $\pi_p^* = 1 - \pi_s^*$.

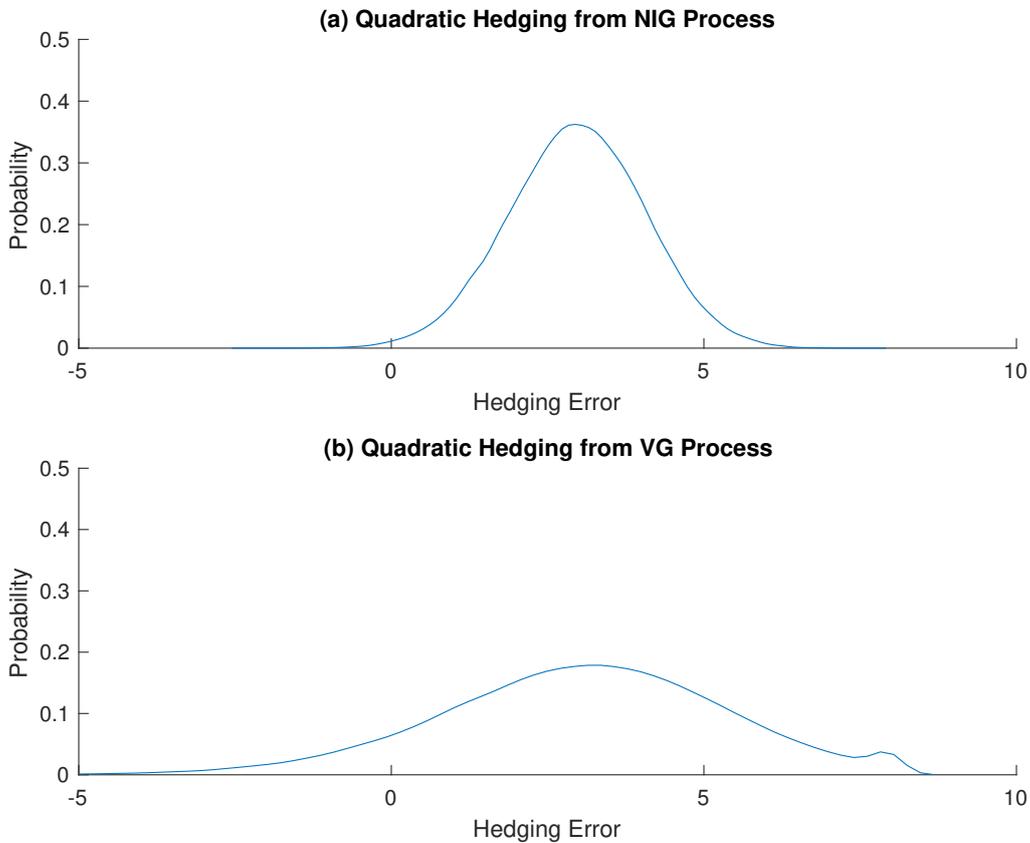


Figure 5.1: PDF of Monte Carlo simulation of quadratic hedging loss from (a) IG subordinated BM and (b) Gamma subordinated BM, using parameters $S_0 = 57$, $K = 52$, $a = 1/250\kappa$, $b = 1/\kappa$, $\mu = 0$, $\sigma = 0.02$, $\kappa = 0.1$, $n = 100000$.

We start our analysis with the quadratic hedging problem using Monte Carlo simulation. Figure 5.1 shows the quadratic hedging error distribution from MC simulations. Fig.5.1(a) shows the hedging error where the underlying distribution is assumed to follow the NIG process. Fig.5.1(b) shows the hedging error where the underlying distribution is assumed to follow the VG process. We observe two things; the first is the the hedging loss from the VG process generates a

wider distribution. We can explain this by looking at Fig.4.3(b), which shows that the Gamma subordinated process generates numbers outside the tail of IG subordinated process, and generates hedging errors in the extremes. The second observation is the hump in the upper tail of Fig.5.1(b). This is likely caused by the slight skewness in the Gamma subordinated process generating asymmetric jumps, as shown in Fig.4.3(b).

We observe that the COS method is more efficient at generating the expected cost. The MC simulation for $n = 100000$ NIG processes took 3.228895 seconds while the COS method took 2.977460 seconds. The MC simulation for $n = 100000$ VG processes took 5.240039 seconds while the COS method took 2.961511 seconds.

Table 5.1: Summary of expected quadratic hedging losses, $S_0 = 57$, $K = 52$, $a = 1/250\kappa$, $b = 1/\kappa$, $\mu = 0$, $\sigma = 0.02$, $\kappa = 0.1$, $n = 100000$

Distribution	$\mathbb{E}^{\mathbb{Q}}[X_T - H_T]$	$\mathbb{E}^{\mathbb{Q}}[X_{T,COS} - H_{T,COS}]$
NIG	2.9646	2.9465
VG	3.0491	2.9545

We summarize the expected hedging error simulated from MC and the COS method in Table 5.1. We observe that the expected hedging error from quadratic hedging is non zero for all methods, this is to be expected because quadratic hedging strategies cannot be perfectly hedged in an incomplete market. From Figure 5.2, we observe for both IG subordinated processes, Fig.5.2(a) and Gamma subordinated processes.

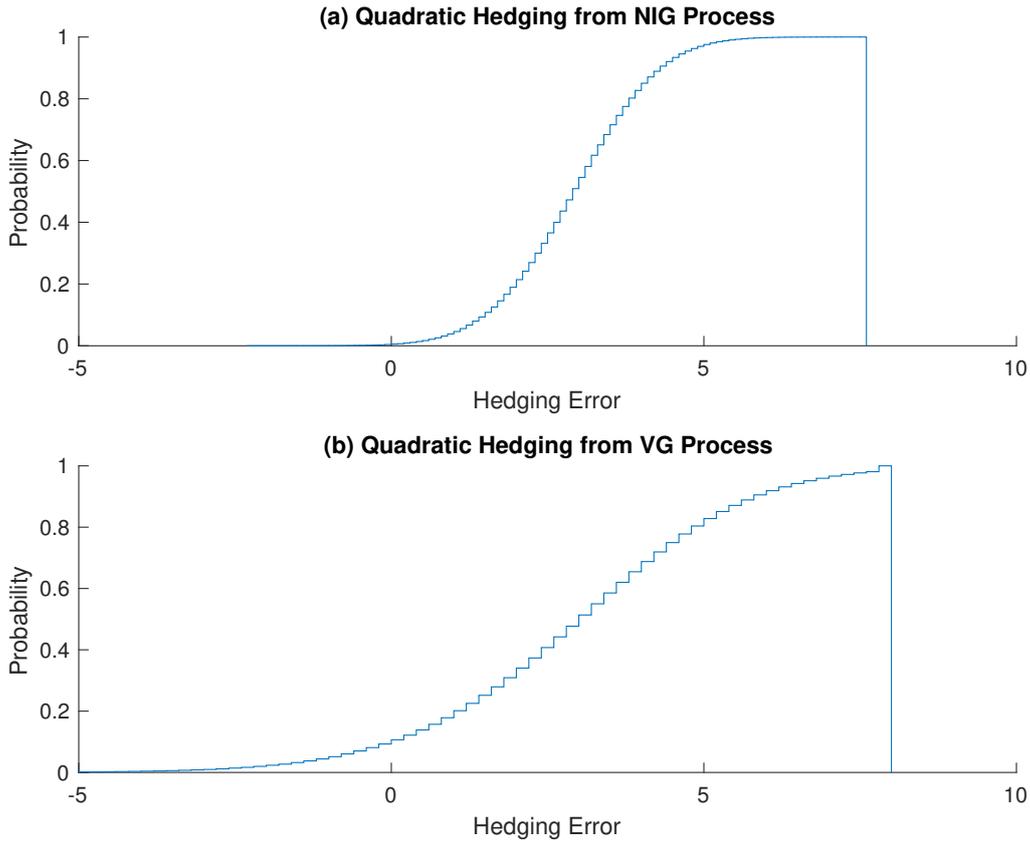


Figure 5.2: CDF of Monte Carlo simulation of quadratic hedging loss from (a) IG subordinated BM and (b) Gamma subordinated BM, using parameters $S_0 = 57$, $K = 52$, $a = 1/250\kappa$, $b = 1/\kappa$, $\mu = 0$, $\sigma = 0.02$, $\kappa = 0.1$, $n = 100000$.

Table 5.2: Optimal portfolios with quadratic hedging loss, $n = 100000$, $K = \$52$, $S_0 = \$57$, $dt = 1/250$, $\kappa = 0.1$

Distributions	π_s	$\pi_{s,COS}$	π_p	$\pi_{p,COS}$
NIG	0.1236	0.1240	0.8764	0.8760
VG	0.1210	0.1237	0.8790	0.8763

We show the comparison of portfolio allocations simulated from MC and the COS method in Table 5.2. The portfolio allocations tell us that for an investor who want to hedge quadratic variation risk, they will invest most of their wealth in the cash account and less than 15% in the underlying.

5.3 Quantile Hedging

With the financial crisis in 2008 regulators started requiring financial institutes to hold enough reserves to offset losses they may occur from portfolio returns. This led to the development of risk measures such as Value at Risk, average Value at Risk and etc. We want to now develop a hedging portfolio that minimizes these risk measures. For this work we assume we are working in risk-neutral probability. Intuitively, we can interpret the risk given by Value at Risk as the risk that the smallest loss the portfolio will incur given a specified probability. AVaR risk has a similar interpretation, but captures more losses than VaR. The hedging VaR and AVaR risks are similar, they both try to replicate the risks using the underlying and cash account.

Definition 5.1. Quantile Function

A function $q_X : (0, 1) \rightarrow \mathbb{R}$ is a quantile function for X if

$$F_X(q_X(s)^-) \leq s \leq F_X(q_X(s)), \quad \forall s \in (0, 1)$$

The left and right continuous inverse functions of F_X

$$q_X^-(s) = \inf\{x \in \mathbb{R} | F_X(x) \geq s\}$$

and

$$q_X^+(s) = \sup\{x \in \mathbb{R} | F_X(x) \leq s\}$$

are called the lower and upper quantile functions respectively, $q_X(u)$ at a given level $u \in [0, 1]$ is called an u -quantile, see [43].

Value at risk is a quantile function, intuitively, it is the minimum amount of cash that an investor must hold to offset unlikely loss in a portfolio

Definition 5.2. Value at Risk (VaR)

For some fixed level $u \in [0, 1]$ and a return function X we define the loss as $L = -X$, then the value at risk of level u is defined as

$$VaR_u(L) := \inf\{c \in \mathbb{R} | \mathbb{P}[L > c] \leq 1 - u\}$$

see, [43]

We redefine VaR_u as the smallest amount of initial capital X_0 required to ensure the hedging error is greater than or equal to zero with a probability higher than u conditioned on a risk neutral setting.

Definition 5.3. VaR Hedging

For some fixed high level $u \in [0, 1]$ and wealth process X , the average value at risk of level u is defined as

$$VaR_u^{\mathbb{Q}}(X_t|\mathcal{F}_t) := \inf\{H_t \in \mathbb{R} | \mathbb{Q}[C_t \geq 0] \geq u\}$$

where $C_t = \mathbb{E}^{\mathbb{Q}}[X_t - H_t|\mathcal{F}_t]$, for readability we drop \mathbb{Q} and \mathcal{F}_t on $VaR_u^{\mathbb{Q}}$ as we move forward.

Intuitively, this can be interpreted as the cheapest replicating portfolio that results in positive hedging error with probability greater than or equal to u . Now we can define the success set to minimize the smallest payoff as $A = \{H_t \mathbf{1}_{C_t \geq 0}\}$. Generally, VaR is not a convex measure and penalizes diversification which is not an desirable feature. However, it is the minimum requirement in the industry which justifies studying it.

The problem of VaR hedging was originally addressed by Föllmer and Leukert [23] and solved using the Neyman-Pearson lemma. Bouchard, Elie and Touzi, see [6], solved the problem by reformulating it as a stochastic target problem by introducing a new random process. Moreau further extended Bouchard, Elie and Touzi's analysis to jump diffusion models, see [36]. In our work, we use a semi-static hedging strategy to hedge for quantile risk. The following proposition is taken from Leukert and Föllmer and is presented without proof

Proposition 5.4. Quantile Hedging

For $C = \{V_T \geq H\}$, let $\bar{C} \in \mathcal{F}_t$ be a solution of the problem

$$\mathbb{P}[C] = \max$$

under the constraint

$$\mathbb{E}^{\mathbb{Q}}[H \mathbf{1}_C] \leq \bar{V}_0$$

where \mathbb{Q} is a unique EMM. Let $\bar{\xi}$ denote the hedge for the option $\bar{H} = H \mathbf{1}_C \in L^1(\mathbb{Q})$, i.e.

$$\mathbb{E}^{\mathbb{Q}}[H \mathbf{1}_{\bar{C}}|\mathcal{F}_t] = \mathbb{E}^{\mathbb{Q}}[H \mathbf{1}_{\bar{C}}] + \int_0^T \bar{\xi}_t dX_t \text{ P-a.s.}$$

Then $(\bar{V}_0, \bar{\xi})$ solves the optimization problem given by

$$\mathbb{P}[V_0 + \int_0^T \xi_t dX_t] = \max$$

under the constraint

$$V_0 \leq \bar{V}_0$$

Proof. see Leukert and Föllmer, [23]. □

Corollary 5.5. *Building on Proposition 5.4 of Leukert and Föllmer, see [23]. Let π_s and π_p be the fraction of wealth invested in a stock and risk free asset respectively. An investor who wants to minimize VaR at level $u \in [0, 1]$ will have the optimal portfolio allocation given by*

$$\begin{aligned}\pi_s^* &= \alpha^*/X_T \\ \pi_p^* &= 1 - \pi_s^*\end{aligned}$$

by solving the following system of equations

$$\begin{aligned}X_t &= e^{-rT} A(n) + e^{rT} \beta + \alpha S_T \\ \beta &= H_0 - \alpha S_0 \\ C_t &= 0\end{aligned}$$

at the n^{th} element in the set $A = H_t \mathbb{1}_{C_t \geq 0}$, where $n = \text{floor}[N(1 - u)] + 1$, and N is the size of the set $A(1, 2, \dots, n)$.

Proof. Let $A = \mathbb{E}^{\mathbb{Q}}[H_t \mathbb{1}_{C_t \geq 0}]$ be the success set, then the optimal VaR, VaR_u^* , is given by

$$VaR_u^* := VaR_u(n)$$

To find the optimal portfolio we want to find the success set A such that $C_t \geq 0$ and order the payoffs in ascending order. Then we find the index $n = \text{floor}[Nu] + 1$, where N is the length of the set $[1, 2, \dots, n]$, then the optimal VaR is given by $VaR_u^* = VaR_u(n)$. By replicating the portfolio for VaR_u^* we get the portfolio weights

$$\begin{aligned}VaR_u^* &= X_t \\ &= e^{-rT} A(n) + e^{rT} \beta + \alpha S_T \\ \beta &= H_0 - \alpha S_0\end{aligned}$$

then solving for α such that $C_t = 0$ we get the optimal α^* , where $C_t = e^{-rT}(X_t - H_t)$

$$\begin{aligned}\pi_s^* &= \alpha^*/X_T \\ \pi_p &= 1 - \pi_s^*\end{aligned}$$

□

We can extend this to find a hedging portfolios under average value at risk (AVaR).

Definition 5.6. Average Value at Risk (AVaR) Hedging

The average value at risk at level $u \in [0, 1]$ of a payoff function $X \in L^1(\Omega)$ is given by

$$AVaR_u^{\mathbb{Q}}(X_t | \mathcal{F}_t) = \frac{1}{u} \int_u^1 VaR_s^{\mathbb{Q}}(X_t | \mathcal{F}_t) ds$$

again for readability we drop \mathbb{Q} and \mathcal{F}_t on $AVaR_u^{\mathbb{Q}}$ as we move forward.

Then our success set to minimize is again given by A

Corollary 5.7. *Building on Proposition 5.4 of Leukert and Fömer, see [23]. Let π_s and π_p be the fraction of wealth invested in a stock and risk free asset respectively. An investor who wants to minimize AVaR at level $u \in [0, 1]$ will have the optimal portfolio allocation given by*

$$\begin{aligned}\pi_s^* &= \alpha^*/X_T \\ \pi_p^* &= 1 - \pi_s^*\end{aligned}$$

by solving the following system of equations

$$\begin{aligned}X_t &= e^{-rT} \frac{1}{Nu} \sum_{i=1}^{n-1} A(i) + e^{-rT} A(n) + e^{rT} \beta + \alpha S_T t \\ \beta &= H_0 - \alpha S_0 \\ C_t &= 0\end{aligned}$$

where the n^{th} index is found as in VaR and N is the size of the set $A(1, \dots, n)$, we denote M as the last index of the set A .

Proof. The discretized AVaR at level u is given by

$$AVaR_u = \frac{1}{Nu} \sum_{i=1}^{n-1} VaR_u(i) + VaR_u(n)$$

then we can replicate the AVaR by

$$AVaR_u^* = e^{-rT} \frac{1}{N(1-u)} \sum_{i=1}^{n-1} A(i) + e^{-rT} A(n) + e^{rT} \beta + \alpha S_T$$

then solving the system of equations

$$\begin{aligned}AVaR_u^* &= X_t \\ &= e^{-rT} \frac{1}{Nu} \sum_{i=1}^{n-1} A(i) + e^{-rT} A(n) + e^{rT} \beta + \alpha S_T \\ \beta &= H_0 - \alpha S_0 \\ C_t &= 0\end{aligned}$$

solving for α such that $C_t = 0$ we get the optimal α^* , where $C_t = e^{-rT}(X_t - H_t)$

$$\begin{aligned}\pi_s^* &= \alpha^*/X_T \\ \pi_p^* &= 1 - \pi_s^*\end{aligned}$$

□

5.3.1 Simulation of Quantile Hedging

We simulate the quantile hedging loss using MC simulation and COS method. We put into practice the method to determine the optimal wealth to hedge quantile risk presented by Leukert and Föllmer, see [23]. To adjust the method to the COS method we assume that the wealth process follows the same distribution as the underlying asset, and the CDF of X_t is given by

$$F_X \approx \operatorname{Re}[\phi_Y(\frac{k\pi}{b-a}) \exp(-ik\pi \frac{a}{b-a})] V_k$$

where V_k is given by

$$V_k = \frac{2}{b-a} \int_a^b e^x \cos(k\pi \frac{x-a}{b-a}) dx$$

where Y_t is the underlying Lévy process, and $\phi_Y(\cdot)$ its characteristic function. To determine the index from the CDF, we looked for the first CDF index that resulted in a value greater than u , where $u \in [0, 1]$. We determine the optimal hedging portfolio for VaR and AVaR as shown in Corollary 5.5 and Corollary 5.7. We observe that the Fourier-Cosine method is an efficient method, but Monte Carlo method is more robust and tractable. A Monte Carlo simulation with 100000 trajectories was efficiently computed in 5.015140 seconds. On the other hand the COS method completed its calculations in 2.940627 seconds.

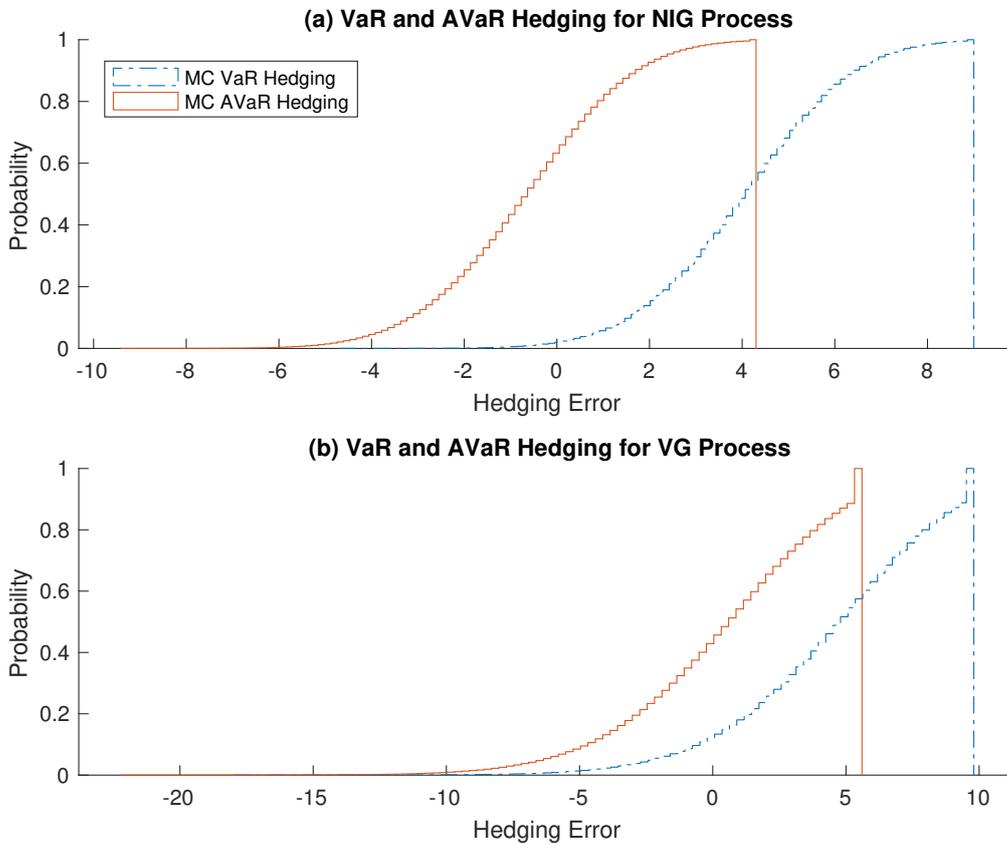


Figure 5.3: CDF of Monte Carlo simulation of (a) VaR and AVaR hedging error for NIG process and (b) VaR and AVaR Hedging error for VG process, using parameters $n = 100000$, $K = \$52$, $S_0 = \$57$, $dt = 1/250$, $\kappa = 0.1, \sigma = 0.02$, $\mu = 0$, $r = 0.036$.

The CDF of the hedging error for VaR and AVaR are shown in Figure 5.3. In a more intuitive sense, hedging VaR is equivalent to creating a replicating portfolio for the smallest positive hedge that results in a positive hedge at a high probability. This gives us the smallest underlying holding we need to hedge this option with a positive hedging error. Table 5.3 shows that when hedging VaR the investor is only required to hold a small portion of wealth in the underlying and a large amount in cash. This implies that to hedge VaR risk, we can simply hedge the option by holding mostly cash.

Table 5.3: Optimal portfolios with VaR hedging error, $n = 100000$, $K = \$52$, $S_0 = \$57$, $dt = 1/250$, $\kappa = 0.1, \sigma = 0.02$, $\mu = 0$, $r = 0.036$

Distributions	$\pi_{s,VaR}^{MC}$	$\pi_{s,VaR}^{COS}$	$\pi_{p,VaR}^{MC}$	$\pi_{p,VaR}^{COS}$
NIG	0.0287	0.1671	0.9713	0.8329
VG	0	0.1681	1	0.8319

Table 5.4: Optimal portfolios with AVaR hedging error, $n = 100000$, $K = \$52$, $S_0 = \$57$, $dt = 1/250$, $\kappa = 0.1, \sigma = 0.02$, $\mu = 0$, $r = 0.036$

Distributions	$\pi_{s,AVaR}^{MC}$	$\pi_{s,AVaR}^{COS}$	$\pi_{p,AVaR}^{MC}$	$\pi_{p,AVaR}^{COS}$
NIG	0.5931	0.5143	0.4069	0.4857
VG	0.3773	0.5191	0.6227	0.5191

Counter intuitively, hedging AVaR requires a more risky portfolio as seen in Table 5.4. However, this is to be expected because to hedge a large loss, simply holding cash cannot provide sufficient returns. Thus the investor must seek higher returns by looking for more risky investments.

Table 5.5: Expected hedging error, $\mathbb{E}^Q[X_T - H_T]$, comparison between MC and COS, $n = 100000$, $K = \$52$, $S_0 = \$57$, $dt = 1/250$, $\kappa = 0.1, \sigma = 0.02$, $\mu = 0$, $r = 0.036$

Distributions	VaR: MC	VaR: COS	AVaR: MC	AVaR: COS
NIG	4.5785	2.2064	-0.6249	-0.2416
VG	6.4388	4.3868	0.4197	-0.0883

We summarize the discounted expected hedging errors in Table 5.5. For an investor looking to hedge VaR they should expect their returns to reflect the payoff of the option as most of their hedging investments are allocated to the cash account and will at most lose a fraction of

any negative movements in the underlying asset. The investor hedging AVaR is actually more exposed to risk as they need to generate higher returns to hedge risk in the entire tail.

5.3.2 Robust Risk Hedging

We can extend AVaR hedging to a robust setting. Following Föllmer et al, see [23], we present risk hedging where the risk measure is robust. Intuitively, this can be thought of as hedging the expected loss regardless of the probability measure used.

Definition 5.8. Robust Risk Measure

For a random variable $X \in \mathcal{L}^1(\Omega)$ and $Q^* \in \mathcal{Q}$ is a robust probability measure in a set of probability measures \mathcal{Q} and is absolutely continuous w.r.t. some probability measure \mathbb{P} . A robust risk measure is given by

$$\rho(X) = \sup_{Q^* \in \mathcal{Q}} \mathbb{E}^{Q^*}[X - \alpha(Q)]$$

Proposition 5.9. *Let $Z^* = \frac{dQ^*}{dQ}$ be the Radon-Nikodym derivative such that $\mathbb{E}^Q[Z|\mathcal{F}_t] = 1$ then setting $\alpha(Q) = 0$ and $Q^* = \{Q \in \mathcal{Q} | Z^* \geq \frac{1}{u}\}$. For some $u \in [0, 1]$, $C_t = \mathbb{E}^Q[X_t - H_t | \mathcal{F}_t]$ and success set A , the optimal portfolio allocation is given by*

$$\begin{aligned} \pi_s^* &= \alpha^* / X_T \\ \pi_p^* &= 1 - \pi_s^* \end{aligned}$$

by solving the following system of equations

$$\begin{aligned} X_t &= e^{-rT} \widehat{AVaR}_u + e^{rT} \beta + \alpha S_T t \\ \beta &= H_0 - \alpha S_0 \\ C_t &= 0 \end{aligned}$$

where \widehat{AVaR}_u is given by

$$\widehat{AVaR}_u = \frac{1}{u} \left[A(n) \left(u - \sum_{i=1}^{n-1} q(i) \right) + \sum_{i=1}^{n-1} A(i) q(i) \right]$$

Proof. From the Neyman-Pearson lemma, see [23], Definition (2.32) and subsequent examples, we get that

$$Z^* = \frac{1}{u} (\mathbb{1}_{X_T > H_T} + \epsilon \mathbb{1}_{X_T = H_T})$$

where

$$\epsilon = \begin{cases} 0 & \mathbb{Q}[X_T = H_T] = 0 \\ \frac{u - \mathbb{Q}[X_T > H_T]}{\mathbb{Q}[X_T = H_T]} & \text{otherwise} \end{cases}$$

then

$$\begin{aligned}
\overline{AVaR}_u^{\mathbb{Q}}(X_t|\mathcal{F}_t) &= \sup_{Q^* \in \mathcal{Q}} \mathbb{E}^{\mathbb{Q}}[ZVaR_u(X_t|\mathcal{F}_t)] = \mathbb{E}^{\mathbb{Q}}[Z^*VaR_u(X_t|\mathcal{F}_t)] = \\
&= e^{-rT} \frac{1}{u} \left[VaR_u(n) \left(u - \sum_{i=1}^{n-1} q(i) \right) + \sum_{i=1}^{n-1} VaR_u(i)q(i) \right] + e^{rT} \beta + \alpha S_T \\
&= e^{-rT} \frac{1}{u} \left[A(n) \left(u - \sum_{i=1}^{n-1} q(i) \right) + \sum_{i=1}^{n-1} A(i)q(i) \right] + e^{rT} \beta + \alpha S_T \\
\beta &= H_0 - \alpha S_0 \\
C_t &= 0
\end{aligned}$$

solving for α such that $C_t = 0$ we get the optimal α^* , where $C_t = e^{-rT}(X_t - H_t)$

$$\begin{aligned}
\pi_s^* &= \alpha^* / X_T \\
\pi_p^* &= 1 - \pi_s^*
\end{aligned}$$

where $q(i) = \mathbb{Q}[X_T(i) = H_T(i)]$, is a counting probability of the number of times the option payoff greater the value at risk of at indices $i = n, \dots, M$, where n is found as in VaR. \square

Chapter 6

Calibration

6.1 Introduction

The problem of solving for unknown parameters given current data is known as the calibration or the inverse problem. We refer the reader to Cont and Tankov for an overview, see [15]. When historic data is used to solve the inverse problem it is known called parameter estimation. In this section we show that our model can reproduce the volatility smile observed in empirical data and present a conditional calibration method to calibrate the bivariate model.

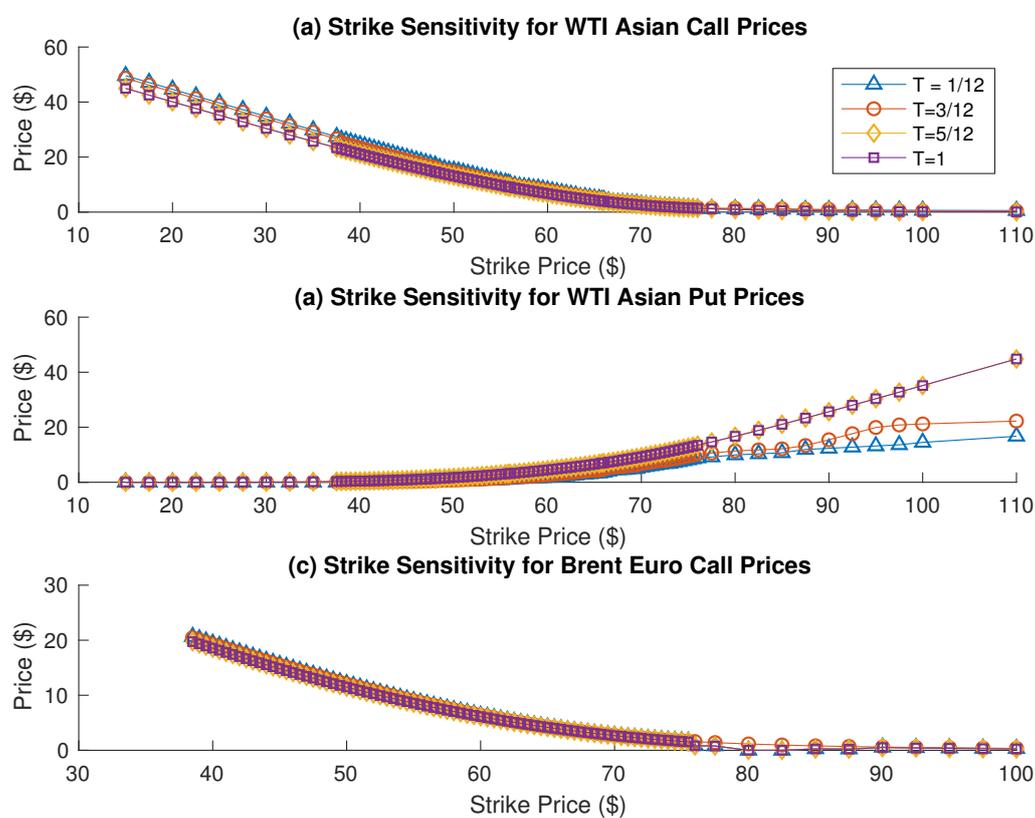


Figure 6.1: Current Option market prices of (a) WTI Asian call options (b) WTI Asian put options (c) Brent crude European call options. The market data was collected from Bloomberg's database via the Bloomberg Terminal

Some paths of current option prices for two different underlying assets, WTI crude oil, $j = 1$, and Brent crude oil, $j = 2$ are shown in Figure 6.1. We observe that the actual market price of the option is different from the theoretical prices. The market price also doesn't follow the monotonic behavior of the time value of money. These differences are likely caused by market friction, randomness in other variables, i.e. interest rate and a phenomenon known as the volatility smile.

6.2 Model Parameter Calibration

A common way of calibrating pricing models is to set the cost function as some distance function, $d(\cdot)$, to minimize. This results in the following minimization problem.

$$J(y) = \min_{y \in \mathcal{U}} \{d(H(y) - H^{mkt})\} \quad (6.1)$$

where y and x are vectors of parameters in a solution set \mathcal{U} .

Definition 6.1. Calibration Problem

Let \mathcal{U} be a set of solutions such that, there exists a collection of optimal solutions $(y^*, x^*) \in \mathcal{U}$ that solves the minimization problem. Given prices of a call option $H^j(y)$ where $j \in [1, 2]$ is the j^{th} . We construct the Lévy process Z_t such that the discounted asset price $S_t = S_0 \exp(Z_t)$ is a martingale.

$$H^j(y) = e^{-rT} \mathbb{E}^{\mathbb{Q}}[(S^j(y) - K)^+ | \mathcal{F}_t] \quad (6.2)$$

Where $S^j(y)$ is some functional that describes the price process, i.e. $S^j(y) = \frac{1}{M+1} \sum_{t=0}^M S_t(y)$ for arithmetic Asian options. Then the calibration problem is the solution of the following minimization problem

$$y^* = \min_y \sqrt{\frac{\sum_{i=1}^N (H_i^j(y) - H^{mkt})^2}{N}}, \text{ for } j = 1, 2 \quad (6.3)$$

We follow Schoutens, see [44], and apply the root mean squared error (RMSE) as our calibration minimization criteria, on market option price of that day.

Remark 6.2. The cost function given by the RMSE $:= \sqrt{\frac{\sum_{i=1}^N (X_i - \hat{X}_i)^2}{N}}$ is convex for $[x, y] \in X$ a convex set and $z \in \mathbb{R}$.

Proof. We can reduce the RMSE to the SE as follows

$$\begin{aligned}\text{RMSE} &= \sqrt{\text{MSE}} \\ \text{MSE} &= \sum_{i=1}^N (\text{SE})/N \\ \text{SE} &= (X_i - \hat{X})^2\end{aligned}$$

then let $f(x) := (x - z)^2$ and $f(y) := (y - z)^2$. Without loss of generality, set $\lambda = 1/2$ and $[x, y] \in X$, then

$$\begin{aligned}f(\lambda x + (1 - \lambda)y) &= f\left(\frac{1}{2}x + \frac{1}{2}y\right) \\ &= \left(\frac{1}{2}x + \frac{1}{2}y - z\right)^2 \\ &= \left(\frac{1}{2}x - \frac{1}{2}z + \frac{1}{2}y - \frac{1}{2}z\right)^2 \\ &= \left(\frac{1}{2}(x - z) + \frac{1}{2}(y - z)\right)^2\end{aligned}$$

by the Triangular inequality

$$\begin{aligned}&\leq \frac{1}{2}(x - z)^2 + \frac{1}{2}(y - z)^2 \\ &= \lambda f(x) + (1 - \lambda)f(y)\end{aligned}$$

Since $f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$ the cost function is convex. \square

Proposition 6.3. *Under the risk neutral assumption the conditional calibration method looks to solve the following problem. Let*

$$\begin{aligned}H_1(y) &= e^{-rT} \mathbb{E}^{\mathbb{Q}} \left[\left(\frac{\sum_{t=0}^M S_t^1(y)}{M+1} - K \right)^+ \right] \\ H_2(y, x|y) &= e^{-rT} \mathbb{E}^{\mathbb{Q}} \left[(S_T^2(y, x) - K)^+ \mid y = y^* \right]\end{aligned}$$

Where $H^1(y)$ is the modelled arithmetic Asian call option payoff for the first underlying asset, and $H^2(y, x)$ is the modelled European call option payoff for the second underlying asset, in a basket of assets. y is a vector of independent parameters to be calibrated and x is a vector of conditionally independent parameters to be calibrated. Then the calibration problem becomes the following optimization problem

$$\begin{aligned}J_1(y) &= \sqrt{\frac{\sum_{i=1}^N (H_i^1(y) - H^{1, mkt})^2}{N}} \\ J_2(y, x|y = y^*) &= \sqrt{\frac{\sum_{i=1}^N (H_i^2(y, x) - H^{2, mkt})^2}{N}}\end{aligned}$$

such that

$$y^* = \min_{y \in \mathcal{U}} \{J_1(y)\} \quad (6.4)$$

$$x^* = \min_{x \in \mathcal{U}} \{J_2(y, x) \mid y = y^* \in \mathcal{U}\} \quad (6.5)$$

where $[y^*, x^*] \in \mathcal{U}$ are the pair of optimal solutions to the inverse problem 6.4 and 6.5 respectively. This optimal solution lives in the space of solution sets \mathcal{U}

Proof. This follows from the fact that the set of solutions for $x^* \in \mathcal{U}$ is disjoint from $y^* \in \mathcal{U}$ when conditioned on $y = y^* \in \mathcal{U}$, i.e. for a function $\inf_{x,y} f(x, y) = \inf_x \tilde{f}(x)$, where $\tilde{f}(x) = \inf_y f(x, y)$ \square

Remark 6.4. To model

$$H_2(y, x | y = y^*) = e^{-rT} \mathbb{E}^{\mathbb{Q}} \left[\left(S_T^j(y, x) - K \right)^+ \mid y = y^* \right]$$

we require the conditional characteristic function of the second independent subordinated processes. To do this we apply Bayes rule to get

$$\phi_{L_j | L_0}(u) = \frac{\phi_{L_0}(u) \phi_{c_j L_j}(u)}{\phi_{L_0}(u)} = \phi_{c_j L_j}(u)$$

Proof. This follows from the direct application of Bayes rule and the independence of the subordinating processes \square

For univariate models, the vector $\mathbf{y} = [\kappa_0, \mu_X, \sigma_X]$ are the three parameters that needs to be calibrated. In the bivariate case, things get more complicated, $\mathbf{y} = [\kappa_0, \kappa_1, c_1, \mu_X^1, \sigma_X^1]$ and $\mathbf{x} = [\kappa_2, c_2, \mu_X^2, \sigma_X^2]$. We can interpret σ_X^j , for $j = 1, 2$, as the volatility in the independent BM, and μ_X^j is interpreted as it's mean. We can recover the μ_Y^j and σ_Y^j by computing the moments in Table 3.3 and Table 3.4. The parameter κ_k , where $k = 0, 1, 2$ is interpreted as the variance of the subordinating process. As $\kappa_k \rightarrow 0$ the process becomes normal, thus for our simulation $\kappa_k \neq 0$. As subordinating processes are monotonically increasing functions, we can see that $\kappa_k > 0$. It is important to note that the parameters above are not directly observable, that is why we treat them as arbitrary parameters to be calibrated.

To solve the minimization problem, we look at an approach known as the Generalized Pattern Search (GPS) algorithm with constraints. The general idea of the GPS algorithm is to evaluate the cost function at the initial point, evaluate the cost function at perturbed points and if one the evaluated perturbed points has a value less than the initial, it becomes the new initial point and the perturbation radius is doubled. If the perturbation fails to yield a value less than the initial point, the perturbation radius is halved. The convergence is considered in the following

paper by Torczon, see [46]. The algorithm is readily available in Matlab as the function `patnsearch`. This approach has the advantage in that it is not a gradient method, so it doesn't require the function to be differentiable; therefore it is quite ideal to use for stochastic processes. The figure below, shows the calibrated payoff and the market payoff at each strike. We can see....

Table 6.1 is shows a snap shot of solutions to the calibration problem for the univariate and bivariate model. Parameters were calibrated using data collected from the Bloomberg terminal. The underlying asset for $j = 1, 2$ was chosen to be WTI crude (June 2018 - June 2019) and Brent crude (June 2018 - June 2019), respectively. The initial points for Calibration was set to $y_0 = [\kappa_0, \mu_X, \sigma_X] = [0.25, 0.3, 0]$ in the univariate case, and the pair $(y_0, x_0) = ([\kappa_0, \kappa_1, c_1, \mu_X^1, \sigma_X^1], [\kappa_2, c_2, \mu_X^2, \sigma_X^2]) = ([0.25, 0.25, 0.2, 0.3, 0], [0.25, 1, 0, 0])$ in the bivariate case.

Table 6.1: Calibrated Parameters for NIG Process, $n = 100000$, $K^1 = K^2 = \$52$, $S_0^1 = \$68.58$, $S_0^2 = \$74.88$, $dt = 1/12$

Model	κ_0	κ_j	c_j	μ_X^j	σ_X^j
Univariate	0.8744	(-, -)	(-, -)	(-0.1875, -)	(0.0005, -)
Bivariate	0.9829	(0.7500, 0.6777)	(0.0633, 0.2)	(-0.2499, -0.9988)	(0.00047, 0.00047)

Fig.6.2(a) and Fig.6.2(b) show the calibrated models against market prices. The models were priced using Monte carlo pricing. At each strike point 1000000 paths were generated. The pattern search algorithm was run until the mesh grid size was less than 1e-6.

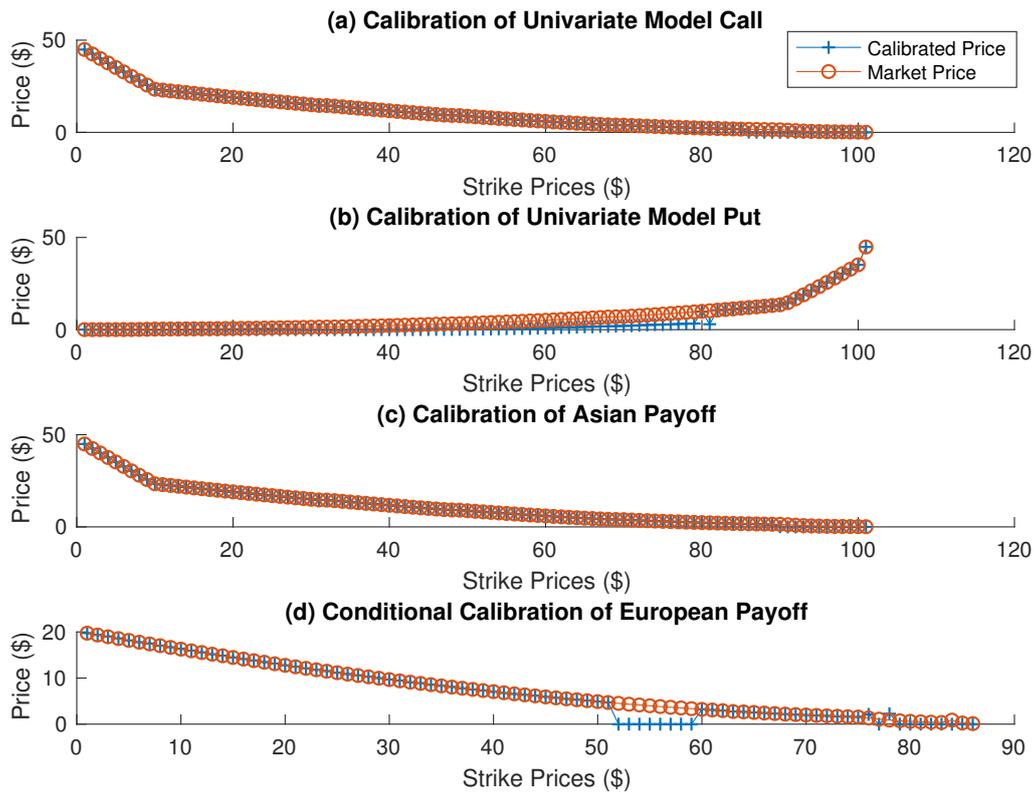


Figure 6.2: (a) Calibrated univariate model matched to current market price of WTI Asian Call Option (b) Calibrated bivariate model matched to current market price of WTI Asian Call Option and (c) Calibrated conditionally univariate model to current market price of Brent European Call Option expiring in 1 year.

Figure 6.2 shows a general good fitness of the calibrated price to the market price. This is one of the advantages to parametric models with more degrees of freedom. However, we can pose the question, how many degrees of freedom is appropriate? As one would expect the run time of the calibration process shows that the more free parameters there are in a model the longer the calibration process. The calibration of parameters for the univariate model took 4820.758297 seconds, over 101 different strikes and 11529.775153 seconds for the bivariate model.

6.3 Implied Volatility

Determining the implied volatility is a key calibration problem for financial models. A pricing model needs to be able to show a volatility smile to ensure that it accounts for the change in volatility with strike price. This phenomenon is observed in empirical data as shown in Figure 6.3.

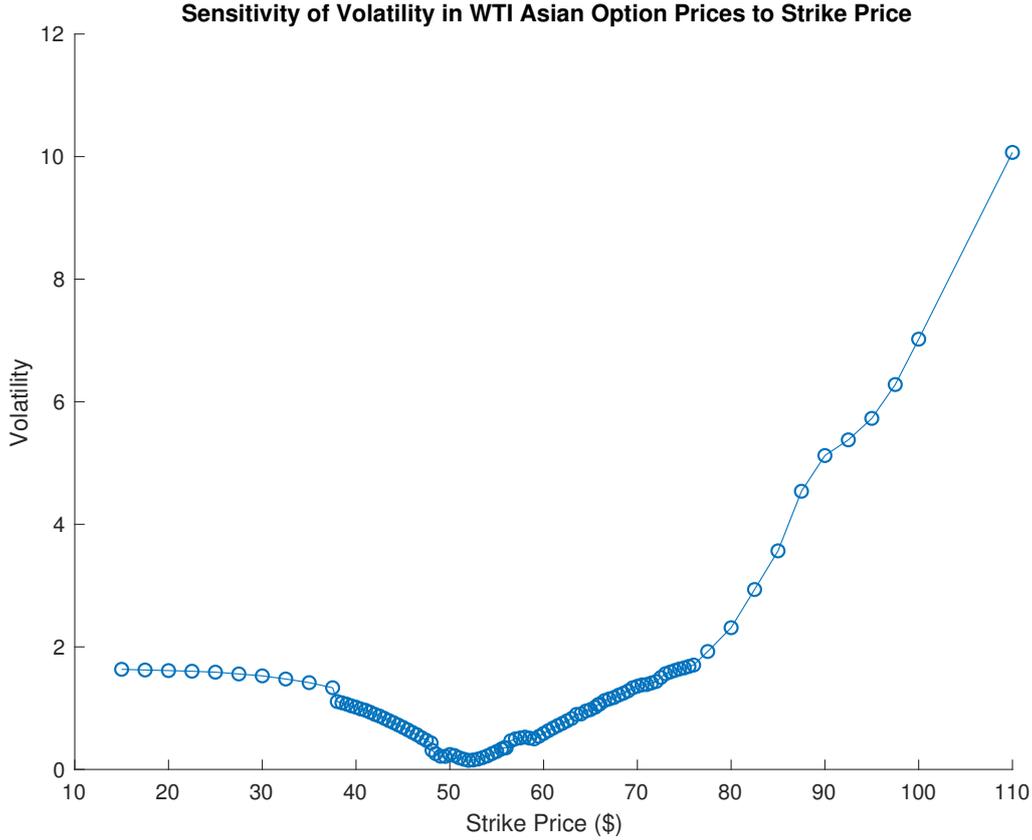


Figure 6.3: Current market sensitivity of volatility in WTI Asian options to strike price.

It is well known that models that like the Black-Scholes model overprices options. This is thought to be the consequence of implied volatility, see [21]. Implied volatility is found by solving the inverse problem for σ , i.e. for the price of an option given by $H(\sigma, K)$ and the market payoff $H^{mkt}(t, K)$, the implied volatility (σ) is found by solving $H(\sigma, K) = H^{mkt}(t, K)$ for σ for a fixed S_0 , T and each K . The main argument against the Black-Scholes model is that it assumes a constant implied volatility.

We approximate the implied volatility of Arithmetic Asian option payoff following the work of Ewald, et al, see [21]. Ewald, et al, argue that σ also depends on the type of option, which is normally neglected. They show that to price general Asian options, fixed strike Asian options should be used to compute implied volatility instead of European options, see [21]. First we need to find vega, denoted $\nu := \frac{\partial H}{\partial \sigma}$. Vega is the sensitivity of the option with respect to volatility. In our work we will use the classical method to solve for vega. Then for call options vega is given by

$$\frac{\partial}{\partial \sigma} \mathbb{E}^{\mathbb{Q}} \left[\left(\frac{1}{M+1} \sum_{t=0}^M S_t - K \right)^+ \right] = \mathbb{E}^{\mathbb{Q}} \left[\mathbf{1}_{\bar{S}_t > K} \frac{1}{M+1} \sum_{t=0}^M S_t (B_{R_t} - \sigma R_t) \right]$$

where \bar{S}_t is the average price of the underlying asset and for put options we look for $\mathbb{1}_{\bar{S}_t < K}$, see [21]. Next we use Newton's method to approximate implied volatility at each strike price K , and $t = T$ is the maturity time of the option. Newton's method is given by

$$\sigma_{k+1} = \sigma_k - \frac{H(\sigma_k, K) - H^{mkt}(T, K)}{\frac{\partial}{\partial \sigma} H(\sigma_k, K)}$$

For call options the initial point is approximated by

$$\sigma_0 \approx \sqrt{\frac{2}{M} \left| \frac{\mathbb{E}^{\mathbb{Q}}[S_t]}{K} \right|}$$

and for put options

$$\sigma_0 \approx \sqrt{\frac{2}{M} \left| \frac{K}{\mathbb{E}^{\mathbb{Q}}[S_t]} \right|}$$

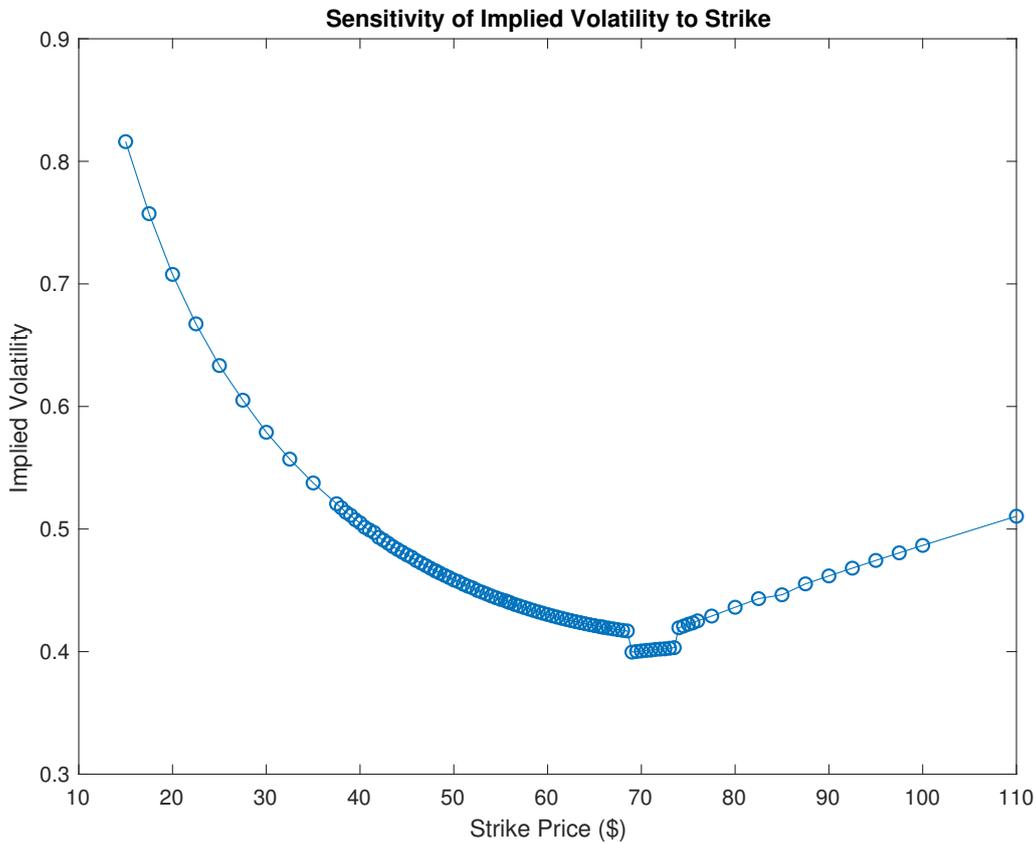


Figure 6.4: Implied volatility smile of the model for different strike prices, the parameters at each strike price, K , were calibrated for both call and put options as described in Section 6.2.

After solving the calibration problem, we can see that our model produces an implied volatility smile, as shown in Figure 6.4. This shows that our model is rich enough to capture the non-linearity of the implied volatility.

Chapter 7

Robust Consumption and Portfolio Choice with Stochastic Interest Rates and Learning about Stock Predictability

7.1 Introduction

We consider an institutional investor who wants to maximize their expected utility over an infinite horizon. We also assume that the investor is willing to substitute consumption of the risky asset with the risk free asset, in a more intuitive sense the investor is willing to have lower returns to for safer investments. The investor is assumed to trade in two risky assets (a stock and a bond) and one risk-free asset (a money market account). The investor is assumed to learn about stock returns using two parameters; λ_t (observable) and β_t (unobservable). β_t only predicts the expected return, while the observed parameter, λ_t , predicts both the expected return and volatility. This is to reflect the fact that estimating the conditional expected returns cannot fully explain the variations in stock risk premium. Like Escobar et al, see [20], we assume in the model that the stock risk premium is driven by an affine combination of function of λ_t and β_t . We assume the investor estimates the unobservable parameter by Bayesian learning and that the stock return volatility depends on the observed parameter; this implies that a stochastic volatility model is a special case of our general model. We assume that an investor has an Epstein-Zin type recursive preferences, see [19]. This preference disentangles the the inverse relationship between the coefficient of relative risk aversion and the elasticity

of intertemporal substitution (EIS), implied by the CRRA (Constant Relative Risk Aversion) preferences. Furthermore, the investor worries that the model describing the investment opportunities is subject to model misspecification which follows the robust optimization problem of Anderson (2002) and outlined in Escobar et al, see [20], this implies the investor is willing to seek robust consumption and portfolio choice. We extend the work of Escobar et al, see [20], by considering Duffie and Epstein recursive preferences, see [19] in an infinite investment horizon. We extend the work of Liu, see [31] by using a bond portfolio to hedge interest risk and the inclusion of unobservable parameters in stock risk premiums.

We summarize some of the existing literature. Campbell and Viceira, see [10], shows that i) at high levels of risk aversion, γ , the optimal consumption-wealth ratio increase with EIS. At low levels of EIS, i.e $\psi < 1$, the optimal ratio rises with risk aversion. While at high levels of ψ , i.e $\psi > 1$, it declines with risk aversion. ii) If the investor is highly risk averse, then they hold almost all wealth in the bond and earn low returns, if they have low to moderate risk aversion they borrow at the riskless rate to earn high leveraged returns. iv) Highly risk-averse investors choose safe portfolios with low average returns, so a higher ψ corresponds to a higher average consumption-wealth ratio. v) Investor with a low risk aversion coefficient is more invested in the risky asset. Maenhout, see [32], shows that robustness in the model reduces equity demand, and effectively acts like risk aversion. He also concludes that robustness decreases the risk free rate. Liu, see [31], shows that when stock returns and expected returns are negatively correlated the robustness actually causes the total equity demand to increase. This is due the increase in hedging demand from state variables, and the decrease in myopic demand (mean-variance hedging). Ju and Miao, see [26], conclude that for EIS, $\psi > 1$, the consumption-wealth ratio is a convex function of robustness and for EIS, $\psi < 1$, the consumption-wealth ratio is a concave function of robustness. They report a calibrated ambiguity aversion, $\theta = 8.86$ and show that ambiguity aversion helps generate variation in consumption-wealth ratio. They also observed that as ambiguity aversion increases, consumption-wealth ratio increases. Escobar et al, see [20], shows that stock return predictability significantly impacts the optimal bond portfolio in the sense that hedge components for the observed and unobserved variables are substantial. The hedge components are larger in the bond portfolio compared with the respective hedge components in the stock portfolio. They also observe that the correlation between bonds and learning parameters determines the investors position in the risky assets. For $\rho_{\lambda P} > 0$ and $\rho_{P\beta} > 0$ the investor will short the bond and for $\rho_{\lambda P} < 0$ and $\rho_{P\beta} < 0$ the investor will buy the bond.

7.2 Background

7.2.1 Expected Utility

Why do we choose to optimize an investors expected utility over a more natural measure such as the expected returns? This is because in 1713, Nicholas Bernoulli pointed out a critical flaw in choosing expected returns by posing the "St. Petersburg Paradox". The paradox goes like this:

Peter tosses a coin and continues to do so until it lands "heads". He pays Paul one ducat if he gets "heads" on the very first throw, two ducats if he gets it on the second, four if on the third, eight if on the fourth, and so on, so that on each additional throw the number of ducats he must pay is doubled.

The paradox in the problem is that the expected amount that Peter must pay Paul is infinite, but, intuitively, most people would not be willing to pay an infinite amount. In 1738, Daniel Bernoulli (Nicholas' cousin) provided an explanation for the St. Petersburg Paradox by introducing the concept of expected utility. The concept was later formalized by John von Neuman and Oskar Morgenstein in 1944, see [39].

The von Neumann-Morgenstern axioms of expected utility¹ are as follows

1) *Completeness*

For any two lotteries P and Q , either $Q > P$ or $P > Q$ or $P \sim Q$

2) *Transitivity*

If $Z \geq Q$ and $Q \geq P$, then $Z \geq P$

3) *Continuity*

If $Z \geq Q \geq P$, there exists some $\lambda \in [0, 1]$ such that $Q \sim \lambda Z + (1 - \lambda)P$

4) *Dominance*

Let P_1 be the compound lottery $\lambda_1 P + (1 - \lambda_1)Q$ and P_2 be the compound lottery $\lambda_2 P + (1 - \lambda_2)Q$.

If $P > Q$ then $P_1 > P_2$ iff $\lambda_1 > \lambda_2$

This axiom tells us that the investor prefers the lottery with highest expected utility. It is well known that this is only a partial explanation of investors behaviour, see [39]. We now look at risk aversion.

¹We omit the *Independence* axiom, as empirical data exists that shows that the axiom does not hold.

7.2.2 Risk Aversion

Let $U(X)$ denote some utility function. It was proposed by Daniel Bernoulli in 1738 that utility functions should be an increasing concave function of wealth. To show that this concavity represented the investors reservations to invest in a fair lottery, John W. Pratt(1964) used the concept of risk premium RP , where

$$U(X - RP) = \mathbb{E}[U(X + L)]$$

where L is the value of the lottery. Now setting $L = 0$, and taking the Taylor expansion about $L = 0$ and $X = 0$

$$RP = -\frac{1}{2}\sigma^2 \frac{U''(X)}{U'(X)}$$

where $\sigma^2 = \mathbb{E}[L^2]$. We can see that $RP > 0$ when $U(X)$ is a concave function and this led to the development of the Pratt(1964)-Arrow(1971) absolute risk aversion measure.

Definition 7.1. Absolute Risk Aversion

Let $U(X)$ in C^2 be a utility function for given wealth X then the absolute relative risk is given by

$$R(X) = -\frac{U''(X)}{U'(X)}$$

Definition 7.2. Relative Risk Aversion²

Let $U(X)$ in C^2 be a utility function for given wealth function X then the absolute relative risk is given by

$$R_r(X) = -X \frac{U''(X)}{U'(X)}$$

7.2.3 Filtering

Filtration is a heavily studied topic in fields such as engineering and economics. It is generally used for signal processing and estimation, however it is quickly becoming common in Finance, we direct the reader to Date and Ponomareva's review on linear and nonlinear filtering in math finance, see [16]. The main use of filtering is to estimate an unobservable parameter, β_t with an observable parameter, λ_t . We assume that λ_t and β_t follow mean-reverting processes, and apply the non-linear(in β_t) filtering technique to β_t and obtain a set of filtered equations, see [29] and [30]. We present the main theorem of non-linear filtering without proof below.

²The relative risk aversion is more frequently used in financial economics, this is because it allows us to account for different levels of wealth.

Theorem 7.3. Filtering^{3,4}

Let (β_t, λ_t) be random processes with differentials given by equations

$$\begin{aligned} d\beta_t &= [a_0(t, \lambda) + a_1(t, \lambda)\beta_t]dt + b_1(t, \lambda)dW_1(t) + b_2(t, \lambda)dW_2(t) \\ d\lambda_t &= [A_0(t, \lambda) + A_1(t, \lambda)\beta_t]dt + B(t, \lambda)dW_2(t) \end{aligned}$$

where $W_1(t)$ and $W_2(t)$ are independent Brownian motions and the functions $a_0, a_1, b_1, b_2, A_0, A_1, B$ meet all boundedness conditions in Liptser and Shiryaev, see [30],[29], and the conditional distribution $P(\beta_0 \leq a | \lambda_0) \sim N(m_0, v_0)$, then m_t and v_t are given by

$$\begin{aligned} dm_t &= [a_0 + a_1 m_t]dt + \frac{b_2 B + A_1 v_t}{B^2} [d\lambda_t - (A_0 + A_1 m_t)dt] \\ \frac{dv_t}{dt} &= 2a_1 v_t + b_1^2 + b_2^2 - \left(\frac{b_2 B + A_1 v_t}{B} \right)^2 \end{aligned}$$

subject to the conditions $m_0 = \mathbb{E}[\theta_0 | \lambda_0]$, $v_0 = \mathbb{E}[(\theta_0 - m_0)^2 | \lambda_0]$

Proof. see [29], Sections 8.1 and 12.1 □

We can see that v_t is the conditional variance, also known as the tracking error and m_t is the conditional mean also known to be the optimal value in the mean-square sense, see [29].

7.2.4 Overview of Stochastic Control

Fix $\mathcal{U} \subseteq \mathbb{R}^d$ as our domain, i.e. the subspace of admissible portfolios, see Definition 2.15, and let X_t be a stochastic process of the form

$$\frac{dX_t^u}{X_t^u} = \mu(u)dt + \sigma(u)dW_t$$

The predictable process, $u = u(t, \omega)$, is known as the control process, $\mu(u)$, is the mean as a function of the control process, and $\sigma(u)$ is the standard deviation as a function of the control process then we call $X_t = X_t^u$ our controlled process, see [37].

In stochastic control problems we consider the performance criterion $J(t, x, u) = J^u(x)$ with stopping time T as:

$$J^u(x) = \mathbb{E} \left[\int_t^T f(X_t, u)dt + U(X_T) \mathbb{1}_{T < \infty} | \mathcal{F}_t \right] \quad (7.1)$$

Where $f(\cdot)$ and $U(\cdot)$ are given continuous functions. We call $u \in \mathcal{U}$ admissible when there exists a unique strong solution for X_t for all $x \in \mathbb{R}$. The stochastic control problem is to find the value function, V_t , and, optimal control $u^* \in \mathcal{U}$ defined by

$$V_t(x, u) = \sup_{u \in \mathcal{U}} J^u(x) = J^{u^*}(x) \quad (7.2)$$

³For a comprehensive look at how to apply filtering to price dynamics we refer the reader to Gennotte, see [25]

⁴For boundedness conditions we refer the reader to Liptser and Shiryaev, see [29] and [30]

If $u = u(x)$ is a Markov control, i.e controls of the form $u(t, x) = u(X_{t-})$ then X_t has the infinitesimal generator given by $\mathcal{A}^u v(t, x)$, see [37]

7.2.5 Dynamic Programming

One way to solve the stochastic control problem is by using the Hamilton-Jacobi-Bellman equation. This approach was originally derived to solve control problems in the deterministic sense. Other ways include backward-stochastic differential equations, or determining the dual problem. It is known that there is an equivalence that exists between the three methods to solve control problems.

Theorem 7.4. Hamilton-Jacobi-Bellman Equation

We assume that $v(t, x) \in C^{1,2}$ and that it satisfy the following conditions

- (1) $\frac{\partial v}{\partial t} + \mathcal{A}^{u^*} v(t, x) + f(t, x, u) \leq 0$ for all $x \in \mathbb{R}$, $u^* \in \mathcal{U}$.
- (2) $X_T \in \mathcal{F}_t$ a.s. on $T < \infty$ and $\lim_{t \rightarrow T^-} U(X_T) \mathbb{1}_{T < \infty}$ a.s. for all $u \in \mathcal{U}$.
- (3)

$$\mathbb{E} \left[|v(T, X_T)| + \int_t^T \left(|\mathcal{A}v(t, X_t)| + \left| \sigma \frac{\partial v}{\partial x} \right|^2 \right) dt \middle| X_t = x \right] < \infty$$

for all $u \in \mathcal{U}$ and all \tilde{T} in the set of all stopping times.

- (4) $v^- \{X_{\tilde{T}}\}_{\tilde{T} < T}$ is uniformly integrable for all $u \in \mathcal{U}$ and $x \in \mathbb{R}$

then

$$v(t, x) \leq V_t \text{ for all } x \in \mathbb{R} \tag{7.3}$$

Next, suppose that for each $x \in \mathbb{R}$ there exists $u^* \in \mathcal{U}$ such that

- (1) $\sup_{u \in \mathcal{U}} \{ \mathcal{A}^u v(t, x) + f(t, x, u) \} = 0$
- (2) $v(t, X_{\tilde{T}}^{u^*})_{\tilde{T} < T}$ is uniformly integrable

Let $u^* \in \mathcal{U}$, then u^* is an optimal control and

$$V_t(x, u^*) = J^{u^*}(t, x) \text{ for all } x \in \mathbb{R} \tag{7.4}$$

Proof. see [37], Theorem 3.1 □

7.3 Problem Formulation

Let $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \in [0, T]}, \mathbb{P})$ be a complete probability space with a non-decreasing right continuous filtration. We assume the usual assumptions holds. We consider a market with a stock index, a zero-coupon bond and a money market account. We work in the perspective of an institutional investor with an infinite investment horizon. The money market account is given by

$$dA_t = r_t A_t dt \quad (7.5)$$

where r_t is the short term interest rate. The stock price, S_t is described by the stochastic differential equation

$$\frac{dS_t}{S_t} = \mu_{S,t} dt + \sigma_{S,t} dW_t^S \quad (7.6)$$

where $\mu_{S,t}$ is the expected stock return, $\sigma_{S,t} > 0$ is the stock return volatility and W_t^S is a Wiener process. The interest rate is given by a Ornstein-Uhlenbeck (OU) process

$$dr_t = \kappa_r (\bar{r} - r_t) dt - \sigma_r dW_t^P \quad (7.7)$$

where κ_r is the degree of mean reversion, \bar{r} is the long-run mean of the interest rate(r), $\sigma_r > 0$ is the interest rate volatility and W_t^P is a W_t^S correlated Wiener process such that $dW_t^S dW_t^P = \rho_{S,P} dt$.

We assume that stock risk premium, $\alpha_t = \frac{\mu_{S,t} - r_t}{\sigma_{S,t}}$, is an affine function of two stochastic parameters λ_t and β_t . λ_t is the parameter observable to the investor and β_t is the parameter unobservable to the investor. The parameter dynamics is given by

$$d\lambda_t = \kappa_\lambda (\bar{\lambda} - \lambda_t) + \sigma_\lambda dW_t^\lambda \quad (7.8)$$

$$d\beta_t = \kappa_\beta (\bar{\beta} - \beta_t) + \sigma_\beta dW_t^\beta \quad (7.9)$$

where κ_λ and κ_β are the degrees of mean reversion. $\bar{\lambda}$ and $\bar{\beta}$ are the long-run means. $\sigma_\lambda > 0$ and $\sigma_\beta > 0$ are the parameter volatilities of λ_t and β_t , respectively. W_t^λ and W_t^β are correlated Wiener processes such that $dW_t^S dW_t^\lambda = \rho_{S,\lambda} dt$, $dW_t^P dW_t^\lambda = \rho_{P,\lambda} dt$, $dW_t^S dW_t^\beta = \rho_{S,\beta} dt$ and $dW_t^P dW_t^\beta = \rho_{P,\beta} dt$.

It is assumed that the stock return volatility is dependent on the observable parameter λ_t , i.e. $\sigma_S(\lambda_t)$. This setup is the same setup presented in Escobar, et al, see [20]. They present a model where the stock risk premium depends on the observed and unobserved parameters while the stock return volatility depends on the parameter observed by the investor. Their setup further implies that the expected stock return depends on the two parameters λ_t and β_t , i.e

$\mu_{S,t} = \mu_{S,t}(\lambda_t, \beta_t)$ in a recursive relative risk utility framework. From this the investor assumes the risky asset with the following dynamics

$$\frac{dS_t}{S_t} = (r_t + \sigma_S(\lambda_t)(P + P_\lambda \lambda_t + P_\beta \beta_t))dt + \sigma_S(\lambda_t)dW_t^S \quad (7.10)$$

where P , P_λ and P_β are real constants. P_λ and P_β specify the predictive power of λ_t and β_t for the risk premium, respectively. For convenience we will simplify the notation and write $\sigma_S(\lambda_t)$ as σ_S . The price of the zero-coupon bond expiring at a time T is given by

$$\frac{dP_t}{P_t} = (r_t + q)dt + \sigma_{P,t}dW_t^P \quad (7.11)$$

where $q = q_r \sigma_{P,t}$ is the expected excess return on the bond, q_r is the constant market price of interest rate risk and $\sigma_{P,t} = \sigma_r \frac{1}{\kappa_r} (1 - e^{-\kappa_r(T-t)})$ is the bond price volatility. It is assumed that the investor follows a roll-over strategy and the maturity is kept constant i.e. $\sigma_{P,t} = \sigma_P$. The model of stock and bond dynamics follows the principle of no-arbitrage.

The optimal filtered equations from Proposition 1. in Escobar et al, see [20] is restated below.

Proposition 7.5. *Let $\mathcal{F}_t^{S,r,\lambda}$ be the σ -algebra generated by S_t , λ_t and r_t . The stock and bond prices of the investor is given by*

$$\begin{bmatrix} \frac{dS_t}{S_t} \\ \frac{dP_t}{P_t} \end{bmatrix} = \underbrace{\begin{bmatrix} r_t + \sigma_S(P + P_\lambda \lambda_t + P_\beta \hat{\beta}_t) \\ r_t + q \end{bmatrix}}_{:=\boldsymbol{\mu}} dt + \underbrace{\begin{bmatrix} \sigma_S & 0 \\ \rho_{SP}\sigma_P & \sigma_P\sqrt{1-\rho_{SP}^2} \end{bmatrix}}_{:=\boldsymbol{\Sigma}_X} \underbrace{\begin{bmatrix} dB_{t,1} \\ dB_{t,2} \end{bmatrix}}_{:=d\mathbf{B}} \quad (7.12)$$

and the dynamics of the state variables are given by

$$\underbrace{\begin{bmatrix} d\lambda_t \\ dr_t \\ d\hat{\beta}_t \end{bmatrix}}_{d\mathbf{Y}} = \underbrace{\begin{bmatrix} \kappa_\lambda(\bar{\lambda} - \lambda_t) \\ \kappa_r(\bar{r} - r_t) \\ \kappa_\beta(\bar{\beta} - \hat{\beta}_t) \end{bmatrix}}_{:=\boldsymbol{\mu}_Y = \boldsymbol{\kappa}(\bar{\mathbf{Y}} - \mathbf{Y})} dt + \underbrace{\begin{bmatrix} \sigma_\lambda \rho_{S,\lambda} & \sigma_\lambda \hat{\rho}_\lambda & 0 \\ -\sigma_r \rho_{S,P} & -\sigma_r \hat{\rho}_{\lambda,P} & -\sigma_r \hat{\rho}_P \\ A1 & A2 & A3 \end{bmatrix}}_{:=\boldsymbol{\Sigma}_Y} \underbrace{\begin{bmatrix} dB_{t,1}^Y \\ dB_{t,2}^Y \\ dB_{t,3}^Y \end{bmatrix}}_{:=d\mathbf{B}^Y} \quad (7.13)$$

where

$$\underbrace{\begin{bmatrix} \kappa_\lambda & 0 & 0 \\ 0 & \kappa_r & 0 \\ 0 & 0 & \kappa_\beta \end{bmatrix}}_{:=\boldsymbol{\kappa}}, \quad \underbrace{\begin{bmatrix} \bar{\lambda} \\ \bar{r} \\ \bar{\beta} \end{bmatrix}}_{:=\bar{\mathbf{Y}}} \quad \text{and} \quad \underbrace{\begin{bmatrix} \lambda_t \\ r_t \\ \hat{\beta}_t \end{bmatrix}}_{:=\mathbf{Y}}$$

The processes $B_{t,i}$ and $B_{t,j}^Y$ are $\mathcal{F}_t^{S,r,\lambda}$ -adapted correlated Wiener processes, for $i = 1, 2$ and $j = 1, 2, 3$, with the correlation matrix

$$\boldsymbol{\rho} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{\hat{\rho}_{\lambda,P}}{\sqrt{1-\rho_{SP}^2}} \\ 0 & \frac{\hat{\rho}_P}{\sqrt{1-\rho_{SP}^2}} \end{bmatrix} \quad (7.14)$$

where $\rho_{i,j} = d\hat{B}_{t,j}^Y dB_{t,i}$, $A1$, $A2$, $A3$, $\hat{\rho}_\lambda$, $\hat{\rho}_{\lambda,P}$ and $\hat{\rho}_P$ are found in Appendix A of Escobar et al., see [20]

Proof. see [20], Appendix A □

Let $\Phi_t = [\pi_t^S, \pi_t^P]^\top$ be the vector of the fraction of wealth invested in the stock and bond. Let $(1 - \pi_t^S - \pi_t^P)$ be the amount invested in the money market account. Then following Liu, [31], the wealth, X_t , of the investor is given by

$$dX_t = X_t(r_t + \Phi_t^\top(\boldsymbol{\mu} - \mathbf{r}) - C_t)dt + X_t\Phi_t^\top\Sigma_{\mathbf{X}}d\mathbf{B}_t \quad (7.15)$$

where $\Sigma_{\mathbf{X}}\Sigma_{\mathbf{X}}^\top$ is a positive semi-definite covariance matrix and C_t is consumption at time t . We assume that the investor has recursive preferences over consumption with the Duffie and Epstein continuous-time parametrization as shown in Liu, see [31]. In the absence of robustness the parametrization has the form

$$V_t = \int_t^\infty f(C_s, V_s)ds$$

where V_t is the value function and $f(C, V)$ is a normalized aggregator of current consumption and continuation utility given by

$$f(C, V) = \frac{\varphi}{1 - \frac{1}{\psi}}(1 - \gamma)V \left(\left(\frac{C}{(V(1 - \gamma))^{\frac{1}{1-\gamma}}} \right)^{1 - \frac{1}{\psi}} - 1 \right) \quad (7.16)$$

We can define a space of admissible portfolios, \mathcal{U} , such that $\{\pi_t\}_{t \in [0, \infty)} \subset \mathbb{R}^2$ satisfies

- (1) $\pi_t : [0, \infty) \times \Omega \mapsto \mathbb{R}^2$ is $\mathcal{F}_t^{S, \lambda, r}$ -progressively measurable;
- (2) in the subspace \mathcal{U} , there exists an optimal π_t^* for any $x \in (0, \infty)$ such that equation (7.15) admits a unique strong solution;
- (3) the utility function needs to satisfy the integrability condition, i.e. $\mathbb{E}^\mathbb{P} |f(C_t, V_t)| < \infty$ where $f(C_t, V_t)$ is given by equation (7.16);
- (4) $X_t \geq 0$ a.s, $t \in (0, \infty)$.

The investor is assumed to have a Epstein-Zin type preferences with the Value function for preferences given by

$$V = [G(t, Y)]^{-\frac{1-\gamma}{1-\psi}} \frac{X_t^{1-\gamma}}{1-\gamma} \quad (7.17)$$

We assume $\gamma > 1$ and $\psi \neq 1$ to avoid imaginary solutions. $\psi = 1$ case corresponds to the CRRA utility case.

$G(t, Y)$ is the ansatz given by

$$G(t, Y) = \exp(\mathbf{A}(t) + \mathbf{B}^\top(t)\mathbf{Y} + \frac{1}{2}\mathbf{Y}^\top\mathbf{Q}(t)\mathbf{Y})$$

where

$$\mathbf{A}(t) = \begin{bmatrix} a_0(t) \end{bmatrix}, \quad \mathbf{B}(t) = \begin{bmatrix} b_1(t) \\ b_2(t) \\ b_3(t) \end{bmatrix}, \quad \mathbf{Q}(t) = \begin{bmatrix} c_{11}(t) & c_{12}(t) & c_{13}(t) \\ c_{21}(t) & c_{22}(t) & c_{23}(t) \\ c_{31}(t) & c_{32}(t) & c_{33}(t) \end{bmatrix}$$

the partials of $G(t, Y)$ with respect to the state variables are given by

$$\begin{aligned} \mathbf{G}_Y &= G(t, Y) \left(\mathbf{B} + \frac{1}{2} (\mathbf{Q} + \mathbf{Q}^\top) \mathbf{Y} \right), \\ \mathbf{G}_{YY} &= G(t, Y) \left[\left(\mathbf{B} + \frac{1}{2} (\mathbf{Q} + \mathbf{Q}^\top) \mathbf{Y} \right) \left(\mathbf{B} + \frac{1}{2} (\mathbf{Q} + \mathbf{Q}^\top) \mathbf{Y} \right)^\top + \frac{1}{2} (\mathbf{Q} + \mathbf{Q}^\top) \right] \end{aligned}$$

To simplify notation we drop time in notation, i.e. $A_t := A$, and all subscripts denote derivatives. The partials of the recursive utility function, $V(X, Y)$, with respect to time, t , the Wealth process, X , and state vector, Y , is given by

$$\begin{aligned} V_X &= G^{-\frac{1-\gamma}{1-\psi}} X^{-\gamma} \\ V_{XX} &= -\gamma G^{-\frac{1-\gamma}{1-\psi}} X^{-\gamma-1} \\ \mathbf{V}_Y &= -\frac{1}{1-\psi} G^{-\frac{1-\gamma}{1-\psi}} X^{1-\gamma} \frac{\mathbf{G}_Y}{G} \\ \mathbf{V}_{YY} &= \frac{1}{1-\psi} \left(\frac{1-\gamma}{1-\psi} + 1 \right) G^{-\frac{1-\gamma}{1-\psi}} X^{1-\gamma} \frac{\mathbf{G}_Y \mathbf{G}_Y^\top}{G^2} - \frac{1}{1-\psi} G^{-\frac{1-\gamma}{1-\psi}} X^{1-\gamma} \frac{\mathbf{G}_{YY}}{G} \\ \mathbf{V}_{XY} &= -\frac{1-\gamma}{1-\psi} G^{-\frac{1-\gamma}{1-\psi}} X^{-\gamma} \frac{\mathbf{G}_Y}{G} \end{aligned}$$

A robust investor is one who deems the state dynamics of equations (7.7), (7.8), (7.9) and (7.15) to be approximates and possibly mis-specified. To ensure robustness, we introduce a change of measure \mathbb{P}_u such that, \mathbb{P}_u is absolutely continuous with respect to \mathbb{P} , denoted $\mathbb{P}_u \ll \mathbb{P}$. We denote $\mathbf{Y}_t = [\lambda_t, r_t, \hat{\beta}_t]^\top$ as the state vector where r_t , λ_t and $\hat{\beta}_t$ are the state variables such that each state variable follows the filtered dynamics shown in the paper by Escobar et al., see [20], Proposition 1.

Under the filtered measure \mathbb{P}_u , it follows from Girsanov's theorem that the Radon-Nikodym derivative is given by

$$\left(\frac{d\mathbb{P}_u}{d\mathbb{P}} \right) = \mathcal{E}(X, Y)$$

where

$$\frac{d\mathcal{E}(X, Y)}{\mathcal{E}(X, Y)} = \mathbf{u}(X, Y)^\top d\mathbf{B}^Y, \quad \mathcal{E}(X_0, Y_0) = 1$$

then the compensated Brownian motion is given by

$$\tilde{\mathbf{B}}^Y = \int_0^t \mathbf{u}_s ds + d\mathbf{B}^Y$$

where $\mathbf{u}_t := \mathbf{u}(X_t, Y_t)$ is a vector process that compensates the drift such that $\tilde{\mathbf{B}}_t^Y$ is a Wiener process with respect to the probability measure \mathbb{P}_u . Under the new measure \mathbb{P}_u the state dynamics of the model is given by

$$d\mathbf{Y} = [\kappa(\bar{\mathbf{Y}} - \mathbf{Y}) - \Sigma_{\mathbf{Y}}\mathbf{u}]dt + \Sigma_{\mathbf{Y}}d\tilde{\mathbf{B}}^Y$$

where $\mu_{\mathbf{Y}} - \Sigma_{\mathbf{Y}}\mathbf{u}$ is the perturbed drift vector, $\Sigma_{\mathbf{Y}}$ is the volatility matrix of \mathbf{Y} , and $\Sigma_{\mathbf{Y}}\Sigma_{\mathbf{Y}}^\top$ is the positive semi-definite covariance matrix. Under this new framework our wealth dynamics are given by

$$dX_t = [X_t(r_t + \Phi_t^\top(\boldsymbol{\mu} - \mathbf{r}) - \Phi_t^\top\tilde{\Sigma}_{\mathbf{X}}\mathbf{u}_t) - C_t]dt + X_t\Phi_t^\top\tilde{\Sigma}_{\mathbf{X}}d\tilde{\mathbf{B}}_t^Y$$

Then the optimization problem for a robust investor becomes

$$\sup_{C, \phi} \inf_u \mathbb{E}^{\mathbb{P}^u} \left[\int_0^\infty \left(f(C_s, V_s) - e^{-\varphi s} \frac{1}{2} \mathbf{u}_s^\top \boldsymbol{\eta}^{-1} \mathbf{u}_s \right) ds \right] \quad (7.18)$$

where the second term is a penalty term given by the discounted relative entropy. $\boldsymbol{\eta}$ is a matrix of preference parameters which measures the strength of preference for robustness in the three independent Brownian motions, $d\hat{B}_t^\lambda$, $d\hat{B}_t^r$ and $d\hat{B}_t^\beta$. With a slight abuse in notation; these three independent Brownian terms account for the ambiguity aversion in the stock model, bond model and the ambiguity in the unobserved parameter, β . When $\boldsymbol{\eta} = \mathbf{0}$, is the zero matrix, the investor believes the model is correct.

For an uncertainty averse investor the HJB equation in the infinite horizon is given by

$$0 = \sup_{C, \phi} \inf_u \{ f(C, V) + \mathcal{A}V - V_X X \Phi^\top \Sigma_{\mathbf{X}} \boldsymbol{\rho}^\top \mathbf{u} - \mathbf{u}^\top \Sigma_{\mathbf{Y}}^\top \mathbf{V}_{\mathbf{Y}} + \frac{1}{2} \mathbf{u}^\top \boldsymbol{\eta}^{-1} \mathbf{u} \} \quad (7.19)$$

We let $\tilde{\Sigma}_{\mathbf{X}} := \Sigma_{\mathbf{X}} \boldsymbol{\rho}^\top$, and $\mathcal{A}V$ is the infinitesimal generator of V given by

$$\begin{aligned} \mathcal{A}V &= V_X [X(r + \Phi^\top(\boldsymbol{\mu} - \mathbf{r})) - C] + (\kappa(\bar{\mathbf{Y}} - \mathbf{Y}))^\top \mathbf{V}_{\mathbf{Y}} \\ &+ \frac{1}{2} V_{XX} X^2 \Phi^\top \Sigma_{\mathbf{X}} \Sigma_{\mathbf{X}}^\top \Phi + X \Phi^\top \tilde{\Sigma}_{\mathbf{X}} \Sigma_{\mathbf{Y}}^\top \mathbf{V}_{\mathbf{X}\mathbf{Y}} + \frac{1}{2} \text{Tr}(\Sigma_{\mathbf{Y}} \Sigma_{\mathbf{Y}}^\top \mathbf{V}_{\mathbf{Y}\mathbf{Y}}) \end{aligned} \quad (7.20)$$

Substituting $\mathcal{A}V$ into the HJB equation we get

$$\begin{aligned} 0 &= \sup_{C, \phi} \inf_u \{ f(C, V) + V_X [X(r + \Phi^\top(\boldsymbol{\mu} - \mathbf{r})) - C] + (\kappa(\bar{\mathbf{Y}} - \mathbf{Y}))^\top \mathbf{V}_{\mathbf{Y}} \\ &+ \frac{1}{2} V_{XX} X^2 \Phi^\top \Sigma_{\mathbf{X}} \Sigma_{\mathbf{X}}^\top \Phi + X \Phi^\top \tilde{\Sigma}_{\mathbf{X}} \Sigma_{\mathbf{Y}}^\top \mathbf{V}_{\mathbf{X}\mathbf{Y}} + \frac{1}{2} \text{Tr}(\Sigma_{\mathbf{Y}} \Sigma_{\mathbf{Y}}^\top \mathbf{V}_{\mathbf{Y}\mathbf{Y}}) \\ &- V_X X \Phi^\top \tilde{\Sigma}_{\mathbf{X}} \mathbf{u} - \mathbf{u}^\top \Sigma_{\mathbf{Y}}^\top \mathbf{V}_{\mathbf{Y}} + \frac{1}{2} \mathbf{u}^\top \boldsymbol{\eta}^{-1} \mathbf{u} \} \end{aligned}$$

where

$$\mathbf{V}_{\mathbf{Y}} = \begin{bmatrix} V_\lambda \\ V_r \\ V_\beta \end{bmatrix}, \quad \mathbf{V}_{\mathbf{X}\mathbf{Y}} = \begin{bmatrix} V_{X\lambda} \\ V_{Xr} \\ V_{X\beta} \end{bmatrix}, \quad \mathbf{V}_{\mathbf{Y}\mathbf{Y}} = \begin{bmatrix} V_{\lambda r} & V_{\lambda\lambda} & V_{\lambda\beta} \\ V_{rr} & V_{r\lambda} & V_{r\beta} \\ V_{\beta r} & V_{\beta\lambda} & V_{\beta\beta} \end{bmatrix}$$

Given the space \mathcal{U} of admissible portfolios we can find the first order condition with respect to \mathbf{u} as

$$\mathbf{u}^* = V_X X \eta \tilde{\Sigma}_X^\top \Phi + \eta \Sigma_Y^\top V_Y \quad (7.21)$$

Substituting the first order condition back into the HJB equation and simplifying gives us

$$\begin{aligned} 0 = \sup_{C, \Phi} \{ & f(C, V) + V_X [X(r + \Phi^\top(\boldsymbol{\mu} - \mathbf{r})) - C] + \bar{\mathbf{Y}}^\top \boldsymbol{\kappa}^\top V_Y - \mathbf{Y}^\top \boldsymbol{\kappa}^\top V_Y \\ & + \frac{1}{2} V_{XX} X^2 \Phi^\top \Sigma_X \Sigma_X^\top \Phi + X \Phi^\top \tilde{\Sigma}_X \Sigma_Y^\top V_{XY} + \frac{1}{2} \text{Tr}(\Sigma_Y \Sigma_Y^\top V_{YY}) \\ & - \frac{1}{2} V_X^2 X^2 \Phi^\top \tilde{\Sigma}_X \eta \tilde{\Sigma}_X^\top \Phi \\ & - V_X X \Phi^\top \tilde{\Sigma}_X \eta \Sigma_Y^\top V_Y \\ & - \frac{1}{2} V_Y^\top \Sigma_Y \eta \Sigma_Y^\top V_Y \} \end{aligned} \quad (7.22)$$

Then the first order condition(FOC) for consumption becomes

$$C^* = V_X^{-\psi} [(1 - \gamma)V] \frac{1-\gamma\psi}{1-\gamma} \varphi^\psi \quad (7.23)$$

Substituting the value function we get that

$$C^* = \frac{X \varphi^\psi}{G(Y)}$$

to find the FOC for the portfolio vector Φ we take the derivative of the HJB equation with respect to Φ and set it to 0. Then, solving for Φ we get the expression

$$\Phi^* = \bar{\mathbf{A}}^{-1} \bar{\mathbf{B}}$$

Where

$$\begin{aligned} \bar{\mathbf{A}} &= V_X^2 X^2 \tilde{\Sigma}_X \eta \tilde{\Sigma}_X^\top - V_{XX} X^2 \Sigma_X \Sigma_X^\top \\ \bar{\mathbf{B}} &= V_X X(\boldsymbol{\mu} - \mathbf{r}) + X \tilde{\Sigma}_X \Sigma_Y^\top V_{XY} - V_X X \tilde{\Sigma}_X \eta \Sigma_Y^\top V_Y \end{aligned}$$

and $\bar{\mathbf{A}}$ is non-singular. Following Maenhout, see [32], we impose the homothetic robustness specification. We assume the preference parameter η takes the form

$$\eta(X, Y) = \frac{\boldsymbol{\theta}}{(1 - \gamma)V(X, Y)} > 0 \quad (7.24)$$

where $\boldsymbol{\theta} = \text{diag}([\theta_1, \theta_2, \theta_3]^\top)$ is a diagonal matrix of uncertainty aversion parameter about the stock model, bond model and learning estimation, respectively. Then substituting in V , V_X , V_{XY} , V_Y , V_{YY} , $G(Y)$ into the optimal portfolio we get

$$\Phi^* = \bar{\mathbf{A}}^{-1} \bar{\mathbf{B}} \quad (7.25)$$

where

$$\begin{aligned}\bar{A} &= \Sigma_X(\rho^\top \theta \rho + \gamma I_2) \Sigma_X^\top \\ \bar{B} &= (\mu - r) - \frac{1-\gamma}{1-\psi} \tilde{\Sigma}_X \Sigma_Y^\top \frac{G_Y}{G} + \frac{1}{1-\psi} \tilde{\Sigma}_X \theta \Sigma_Y^\top \frac{G_Y}{G} \\ &= (\mu - r) + \frac{1}{1-\psi} \tilde{\Sigma}_X (\theta - I_3 + \gamma I_3) \Sigma_Y^\top \frac{G_Y}{G}\end{aligned}$$

and

$$I_n := \text{diag}([1_1, 1_2, \dots, 1_n])$$

Then the optimal portfolio becomes

$$\begin{aligned}\Phi^* &= [\Sigma_X(\rho^\top \theta \rho + \gamma I_2) \Sigma_X^\top]^{-1} [(\mu - r) + \frac{1}{1-\psi} \tilde{\Sigma}_X (\theta - I_3 + \gamma I_3) \Sigma_Y^\top \frac{G_Y}{G}] \\ &= \underbrace{[\Sigma_X(\rho^\top \theta \rho + \gamma I_2) \Sigma_X^\top]^{-1}}_{:=\phi_1[2 \times 2]} (\mu - r) \\ &\quad + \frac{1}{1-\psi} [\Sigma_X(\rho^\top \theta \rho + \gamma I_2) \Sigma_X^\top]^{-1} \underbrace{[\tilde{\Sigma}_X (\theta - I_3 + \gamma I_3) \Sigma_Y^\top]}_{:=\phi_2[2 \times 3]} \frac{G_Y}{G} \\ &= \phi_1^{-1} (\mu - r) + \frac{1}{1-\psi} \phi_1^{-1} \phi_2 \frac{G_Y}{G}\end{aligned}$$

Substituting V , V_X , V_{XX} , V_{XY} , V_Y , V_{YY} , and G into the HJB equation we get

$$\begin{aligned}0 &= f(C^*, V) - \frac{\varphi^\psi}{G} V(1-\gamma) + V(1-\gamma)(r + \Phi^{*\top}(\mu - r)) \\ &\quad - \frac{1}{1-\psi} V(1-\gamma)(\bar{Y}^\top \kappa - Y^\top \kappa^\top) \frac{G_Y}{G} \\ &\quad - \frac{\gamma}{2} V(1-\gamma) \Phi^{*\top} \Sigma_X \Sigma_X^\top \Phi^* \\ &\quad + \frac{\gamma-1}{1-\psi} V(1-\gamma) \Phi^{*\top} \tilde{\Sigma}_X \Sigma_Y^\top \frac{G_Y}{G} \\ &\quad + \frac{1}{2} \frac{1}{1-\psi} V(1-\gamma) \left(\frac{1-\gamma}{1-\psi} + 1 \right) \frac{G_Y^\top}{G} \Sigma_Y \Sigma_Y^\top \frac{G_Y}{G} \\ &\quad - \frac{1}{2} \frac{1}{1-\psi} V(1-\gamma) \text{Tr}[\Sigma_Y \Sigma_Y^\top \frac{G_{YY}}{G}] \\ &\quad - \frac{1}{2} V(1-\gamma) \Phi^{*\top} \tilde{\Sigma}_X \theta \tilde{\Sigma}_X^\top \Phi^* \\ &\quad + \frac{1}{1-\psi} V(1-\gamma) \Phi^{*\top} \tilde{\Sigma}_X \theta \Sigma_Y^\top \frac{G_Y}{G} \\ &\quad - \frac{1}{2} V(1-\gamma) \frac{1}{1-\psi} \frac{G_Y^\top}{G} \Sigma_Y \theta \Sigma_Y^\top \frac{1}{1-\psi} \frac{G_Y}{G}\end{aligned} \tag{7.26}$$

where

$$f(C^*, V) = \frac{1}{1-\psi} G^{-\frac{1-\gamma}{1-\psi}} X^{1-\gamma} \left(\varphi \psi - \frac{\varphi^\psi \psi}{G} \right)$$

Dividing by $V(1-\gamma)$, multiplying $(1-\psi)$ and substituting Φ^* we get

$$\begin{aligned}
0 = & -\frac{\varphi^\psi}{G} + \varphi\psi + (1-\psi)r \\
& + (1-\psi)(\boldsymbol{\mu} - \mathbf{r})^\top \boldsymbol{\phi}_1^{-1}(\boldsymbol{\mu} - \mathbf{r}) + \frac{\mathbf{G}_Y^\top}{G} \boldsymbol{\phi}_2^\top \boldsymbol{\phi}_1^{-1}(\boldsymbol{\mu} - \mathbf{r}) \\
& - (\bar{\mathbf{Y}}^\top \boldsymbol{\kappa} - \mathbf{Y}^\top \boldsymbol{\kappa}^\top) \frac{\mathbf{G}_Y}{G} \\
& - \frac{1}{2}(1-\psi)(\boldsymbol{\mu} - \mathbf{r})^\top \boldsymbol{\phi}_1^{-1} \boldsymbol{\Sigma}_X (\gamma \mathbf{I}_2 + \boldsymbol{\theta}) \boldsymbol{\Sigma}_X^\top \boldsymbol{\phi}_1^{-1}(\boldsymbol{\mu} - \mathbf{r}) \\
& - (\boldsymbol{\mu} - \mathbf{r})^\top \boldsymbol{\phi}_1^{-1} \boldsymbol{\Sigma}_X (\gamma \mathbf{I}_2 + \boldsymbol{\theta}) \boldsymbol{\Sigma}_X^\top \boldsymbol{\phi}_1^{-1} \boldsymbol{\phi}_2 \frac{\mathbf{G}_Y}{G} \\
& - \frac{1}{2} \frac{1}{1-\psi} \frac{\mathbf{G}_Y^\top}{G} \boldsymbol{\phi}_2^\top \boldsymbol{\phi}_1^{-1} \boldsymbol{\Sigma}_X (\gamma \mathbf{I}_2 + \boldsymbol{\theta}) \boldsymbol{\Sigma}_X^\top \boldsymbol{\phi}_1^{-1} \boldsymbol{\phi}_2 \frac{\mathbf{G}_Y}{G} \\
& + (\boldsymbol{\mu} - \mathbf{r})^\top \boldsymbol{\phi}_1^{-1} \tilde{\boldsymbol{\Sigma}}_X (-\mathbf{I}_3 + \gamma \mathbf{I}_3 + \boldsymbol{\theta}) \boldsymbol{\Sigma}_Y^\top \frac{\mathbf{G}_Y}{G} \\
& + \frac{1}{1-\psi} \frac{\mathbf{G}_Y^\top}{G} \boldsymbol{\phi}_2^\top \boldsymbol{\phi}_1^{-1} \tilde{\boldsymbol{\Sigma}}_X (-\mathbf{I}_3 + \gamma \mathbf{I}_3 + \boldsymbol{\theta}) \boldsymbol{\Sigma}_Y^\top \frac{\mathbf{G}_Y}{G} \\
& + \frac{1}{2} \frac{\mathbf{G}_Y^\top}{G} \boldsymbol{\Sigma}_Y \underbrace{\left(\frac{1-\gamma}{1-\psi} \mathbf{I}_3 + \mathbf{I}_3 - \frac{1}{1-\psi} \boldsymbol{\theta} \right) \boldsymbol{\Sigma}_Y^\top}_{:=\phi_3} \frac{\mathbf{G}_Y}{G} \\
& - \frac{1}{2} \text{Tr}[\boldsymbol{\Sigma}_Y \boldsymbol{\Sigma}_Y^\top \frac{\mathbf{G}_Y \mathbf{Y}}{G}] \tag{7.27}
\end{aligned}$$

Collecting like terms and simplifying we get

$$\begin{aligned}
0 = & -\frac{\varphi^\psi}{G} + \varphi\psi - \frac{1}{2} \text{Tr}[\boldsymbol{\Sigma}_Y \boldsymbol{\Sigma}_Y^\top \frac{\mathbf{G}_Y \mathbf{Y}}{G}] \\
& + (1-\psi)r - (\boldsymbol{\kappa}(\bar{\mathbf{Y}} - \mathbf{Y}))^\top \frac{\mathbf{G}_Y}{G} \\
& + (\boldsymbol{\mu} - \mathbf{r})^\top \left[\frac{1}{2}(1-\psi) \boldsymbol{\phi}_1^{-1} \right] (\boldsymbol{\mu} - \mathbf{r}) \\
& + (\boldsymbol{\mu} - \mathbf{r})^\top \left[\boldsymbol{\phi}_1^{-1} \boldsymbol{\phi}_2 \right] \frac{\mathbf{G}_Y}{G} \\
& + \frac{1}{2} \frac{\mathbf{G}_Y^\top}{G} \left[\frac{1}{1-\psi} \boldsymbol{\phi}_2^\top \boldsymbol{\phi}_1^{-1} \boldsymbol{\phi}_2 + \phi_3 \right] \frac{\mathbf{G}_Y}{G} \tag{7.28}
\end{aligned}$$

Multiplying G and collecting terms to the HJB equation we get the following PDE

$$\begin{aligned}
0 = & -\varphi^\psi + \varphi\psi G - \frac{1}{2} \text{Tr}[\boldsymbol{\Sigma}_Y \boldsymbol{\Sigma}_Y^\top \mathbf{G}_Y \mathbf{Y}] \\
& + (1-\psi)rG - (\boldsymbol{\kappa}(\bar{\mathbf{Y}} - \mathbf{Y}))^\top \mathbf{G}_Y \\
& + \frac{1}{2}(1-\psi)(\boldsymbol{\mu} - \mathbf{r})^\top \boldsymbol{\phi}_1^{-1}(\boldsymbol{\mu} - \mathbf{r})G \\
& + (\boldsymbol{\mu} - \mathbf{r})^\top \boldsymbol{\phi}_1^{-1} \boldsymbol{\phi}_2 \mathbf{G}_Y \\
& + \frac{1}{2G} \mathbf{G}_Y^\top \underbrace{\left(\frac{1}{1-\psi} \boldsymbol{\phi}_2^\top \boldsymbol{\phi}_1^{-1} \boldsymbol{\phi}_2 + \phi_3 \right)}_{:=\phi_4} \mathbf{G}_Y \tag{7.29}
\end{aligned}$$

To separate the state variables from $(\boldsymbol{\mu} - \mathbf{r})$ we need to look more closely at the following two terms; $(\boldsymbol{\mu} - \mathbf{r})^\top \boldsymbol{\phi}_1^{-1}(\boldsymbol{\mu} - \mathbf{r})$ and $\boldsymbol{\phi}_2^\top \boldsymbol{\phi}_1^{-1}(\boldsymbol{\mu} - \mathbf{r})$. We first expand $(\boldsymbol{\mu} - \mathbf{r})^\top \boldsymbol{\phi}_1^{-1}(\boldsymbol{\mu} - \mathbf{r})$ to it's

components

Since

$$\phi_1^{-1} = \left[\begin{array}{c|c} \phi_1^{-1}(1,1) & \phi_1^{-1}(1,2) \\ \hline \phi_1^{-1}(2,1) & \phi_1^{-1}(2,2) \end{array} \right]$$

Where

$$\begin{aligned} \phi_1^{-1}(1,1) &= \frac{\underbrace{\hat{\rho}_\lambda^2(\theta_1 \rho_{SP}^2 - \gamma) + \theta_2(\hat{\rho}_{\lambda P}^2 - \hat{\rho}_{\lambda P}^2 \rho_{SP}^2) + \theta_3(\hat{\rho}_P^2 - \hat{\rho}_{\lambda P}^2 \rho_{SP}^2)}_{:=\vartheta_0}}{\sigma_S^2(\gamma + \theta_1) \underbrace{(\gamma(\hat{\rho}_\lambda^2 - \hat{\rho}_{\lambda P}^2 \rho_{SP}^2) + \theta_2(\hat{\rho}_{\lambda P}^2 - \hat{\rho}_{\lambda P}^2 \rho_{SP}^2) + \theta_3(\hat{\rho}_P^2 - \hat{\rho}_{\lambda P}^2 \rho_{SP}^2))}_{:=\vartheta_1}} \\ \phi_1^{-1}(1,2) &= \frac{-\rho_{SP} \hat{\rho}_\lambda^2}{\sigma_P \sigma_S \vartheta_1} = \phi_1^{-1}(2,1) \\ \phi_1^{-1}(2,2) &= \frac{\hat{\rho}_\lambda^2}{\sigma_P^2 \vartheta_1} \end{aligned}$$

Then

$$\begin{aligned} (\boldsymbol{\mu} - \mathbf{r})^\top \bar{\boldsymbol{\Phi}}_1^\top (\boldsymbol{\mu} - \mathbf{r}) &= \frac{(P + P_\lambda \lambda + P_\beta \hat{\beta})^2 \vartheta_0 - 2(\gamma + \theta_1) \rho_{SP} \hat{\rho}_\lambda^2 q_r (P + P_\lambda \lambda + P_\beta \hat{\beta}) + q_r^2 (\gamma + \theta_1)}{(\gamma + \theta_1) \vartheta_1} \\ &= J_0 + \mathbf{J}_1^\top \mathbf{Y} + \frac{1}{2} \mathbf{Y}^\top \mathbf{I}_3 \mathbf{J}_2 \mathbf{I}_3 \mathbf{Y} \end{aligned}$$

Where

$$\begin{aligned} J_0 &= \frac{P^2 \vartheta_0 - 2(\gamma + \theta_1) (\rho_{SP} \hat{\rho}_\lambda^2) q_r P + q_r^2 \rho_\lambda^2 (\gamma + \theta_1)}{(\gamma + \theta_1) \vartheta_1} \\ \mathbf{J}_1 &= -\frac{2((\gamma + \theta_1) (\rho_{SP} \hat{\rho}_\lambda^2) q_r - P \vartheta_0)}{(\gamma + \theta_1) \vartheta_1} \begin{bmatrix} P_\lambda \\ - \\ 0 \\ - \\ P_\beta \end{bmatrix} \\ \mathbf{J}_2 &= \frac{2\vartheta_0}{(\gamma + \theta_1) \vartheta_1} \left[\begin{array}{c|c|c} P_\lambda^2 & 0 & P_\lambda P_\beta \\ \hline 0 & 0 & 0 \\ \hline P_\lambda P_\beta & 0 & P_\beta^2 \end{array} \right] \end{aligned}$$

Next we look at $\phi_2^\top \phi_1^{-1}(\boldsymbol{\mu} - \mathbf{r})$. We let

$$\phi_2 \cdot \phi_1^{-1} = \mathbf{H}$$

where

$$\mathbf{F} = \sigma_S \mathbf{H}(\cdot, 1), \quad \mathbf{R} = q_r \sigma_P \mathbf{H}(\cdot, 2)$$

$$\mathbf{g}_0 = \mathbf{P} \mathbf{F} + \mathbf{R}$$

$$\mathbf{g}_1 = \begin{bmatrix} P_\lambda \mathbf{F}(1) & | & 0 & | & P_\beta \mathbf{F}(1) \\ \hline P_\lambda \mathbf{F}(2) & | & 0 & | & P_\beta \mathbf{F}(2) \\ \hline P_\lambda \mathbf{F}(3) & | & 0 & | & P_\beta \mathbf{F}(3) \end{bmatrix}$$

then

$$\phi_2^\top \phi_1^{-1}(\boldsymbol{\mu} - \mathbf{r}) = \mathbf{g}_0 + \mathbf{g}_1 \mathbf{Y} \quad (7.30)$$

Then the PDE becomes

$$\begin{aligned} 0 = & -\varphi^\psi + \varphi\psi G - \frac{1}{2} \text{Tr}[\boldsymbol{\Sigma}_Y \boldsymbol{\Sigma}_Y^\top \mathbf{G}_Y \mathbf{Y}] \\ & + ((1-\psi) \underbrace{r}_{:=\boldsymbol{\rho}^\top \mathbf{Y}} + \frac{1}{2}(1-\psi) \underbrace{(\boldsymbol{\mu} - \mathbf{r})^\top \phi_1^{-1}(\boldsymbol{\mu} - \mathbf{r})}_{:=J_0 + J_1^\top \mathbf{Y} + \frac{1}{2} \mathbf{Y}^\top \mathbf{I}_3 J_2 \mathbf{I}_3 \mathbf{Y}}) G \\ & + \underbrace{(\phi_2^\top \phi_1^{-1}(\boldsymbol{\mu} - \mathbf{r}) - (\boldsymbol{\kappa}(\bar{\mathbf{Y}} - \mathbf{Y}))^\top)}_{:=\mathbf{g}_0 + \mathbf{g}_1 \mathbf{Y}} \mathbf{G}_Y \\ & + \frac{1}{2G} \mathbf{G}_Y^\top \phi_4 \mathbf{G}_Y \end{aligned} \quad (7.31)$$

Where $\boldsymbol{\rho} := [0, 1, 0]^\top$.

We use a log-linear approximation to linearize the non-linear consumption to wealth ratio to reduce the system of ODEs to a system of equations, this same procedure has been used by Chan et al, see [8], Campbell et al, see [9], and Liu, see [31]. So we let

$$\frac{C}{X} = \frac{\varphi^\psi}{G} = \exp(c - x) \approx \exp(k_0 + k_1(\psi \log(\varphi) - \log(G)))$$

where

$$\begin{aligned} k_1 &= \exp(\mathbb{E}[c - z]) \\ c &= \log(C), \quad z = \log(X) \\ k_0 &= k_1(1 - \log(k_1)) \end{aligned}$$

Then we get the following system of equations

$$\begin{aligned} 0 &= -k_0 - k_1 \psi \log(\varphi) + k_1 \mathbf{A} - \frac{1}{2} \text{Tr}[\boldsymbol{\Sigma}_Y \boldsymbol{\Sigma}_Y^\top \tilde{\mathbf{Q}}] - \frac{1}{2} \mathbf{B}^\top (\boldsymbol{\Sigma}_Y \boldsymbol{\Sigma}_Y^\top - \phi_4) \mathbf{B} \\ &\quad - \frac{1}{2} (\psi - 1) J_0 + (\mathbf{g}_0 - \boldsymbol{\kappa} \bar{\mathbf{Y}})^\top \mathbf{B} \\ 0 &= k_1 \mathbf{B} - \tilde{\mathbf{Q}} (\boldsymbol{\Sigma}_Y \boldsymbol{\Sigma}_Y^\top - \phi_4) \mathbf{B} - \frac{1}{2} (\psi - 1) J_1 - (\psi - 1) \boldsymbol{\rho} + \tilde{\mathbf{Q}} (\mathbf{g}_0 - \boldsymbol{\kappa} \bar{\mathbf{Y}}) + (\mathbf{g}_1 + \boldsymbol{\kappa} \mathbf{Y})^\top \mathbf{B} \\ 0 &= k_1 \tilde{\mathbf{Q}} - \tilde{\mathbf{Q}} (\boldsymbol{\Sigma}_Y \boldsymbol{\Sigma}_Y^\top - \phi_4) \tilde{\mathbf{Q}} - \frac{1}{2} (\psi - 1) J_2 + (\mathbf{g}_1 + \boldsymbol{\kappa})^\top \tilde{\mathbf{Q}} + \tilde{\mathbf{Q}} (\mathbf{g}_1 + \boldsymbol{\kappa}) \end{aligned}$$

Where $\tilde{\mathbf{Q}} := \frac{1}{2}(\mathbf{Q} + \mathbf{Q}^\top)$. The original Ricatti equation gives us the condition that \mathbf{Q} is symmetric matrix such that

$$\hat{\mathbf{Q}} = \begin{bmatrix} c_{11} & | & c_{12} & | & c_{13} \\ \hline c_{12} & | & c_{22} & | & c_{23} \\ \hline c_{13} & | & c_{23} & | & c_{33} \end{bmatrix}$$

We then retrieve the equation

$$0 = -k_0 - k_1\psi \log(\varphi) + k_1\mathbf{A} - \frac{1}{2}\text{Tr}[\boldsymbol{\Sigma}_Y\boldsymbol{\Sigma}_Y^\top\hat{\mathbf{Q}}] - \frac{1}{2}\mathbf{B}^\top(\boldsymbol{\Sigma}_Y\boldsymbol{\Sigma}_Y^\top - \phi_4)\mathbf{B} - \frac{1}{2}(\psi - 1)\mathbf{J}_0 + (\mathbf{g}_0 - \kappa\bar{\mathbf{Y}})^\top\mathbf{B} \quad (7.32)$$

$$0 = k_1\mathbf{B} - \hat{\mathbf{Q}}(\boldsymbol{\Sigma}_Y\boldsymbol{\Sigma}_Y^\top - \phi_4)\mathbf{B} - \frac{1}{2}(\psi - 1)\mathbf{J}_1 - (\psi - 1)\boldsymbol{\rho} + \hat{\mathbf{Q}}(\mathbf{g}_0 - \kappa\bar{\mathbf{Y}}) + (\mathbf{g}_1 + \kappa\mathbf{Y})^\top\mathbf{B} \quad (7.33)$$

$$0 = k_1\hat{\mathbf{Q}} - \hat{\mathbf{Q}}(\boldsymbol{\Sigma}_Y\boldsymbol{\Sigma}_Y^\top - \phi_4)\hat{\mathbf{Q}} - \frac{1}{2}(\psi - 1)\mathbf{J}_2 + (\mathbf{g}_1 + \kappa)^\top\hat{\mathbf{Q}} + \hat{\mathbf{Q}}(\mathbf{g}_1 + \kappa) \quad (7.34)$$

We solve the system of equations in Equations (7.32)-(7.34) using the routine originally proposed by Campbell and Viceira, see [10]. This was implemented in Matlab where we used the `vpsolve` function to numerically solve Equations (7.32)-(7.34). The function `vpsolve` uses a Newton's method type approach to solve the symbolic equations numerically. The symbolic toolbox in Matlab was used to verify that, indeed, the optimal portfolio solves the system of equations given by minimizing the HJB equation with respect to portfolio Φ . A check was formed to verify that the derived optimal portfolio was correct, i.e. the linear equation $\bar{\mathbf{A}}\Phi^* - \bar{\mathbf{B}} := [0, 0]^\top$. Where $\bar{\mathbf{A}}$ is a 2×2 matrix with the coefficients of Φ^* and $\bar{\mathbf{B}}$ is a vector of constant terms.

7.4 Simulation and Results

Chan et al, see [8], show that the optimal strategy portfolio strategy does not depend on the EIS parameter ψ , instead it is dependent on ψ indirectly through the consumption-wealth ratio. We simulate the optimal portfolio allocations in stocks and bonds, as well as the myopic demand and hedging components. We use a mix of parameters taken from Escobar et al, [20], and Liu, see [31]. We take the uncertainty aversion parameter $\gamma = 4$, the time preference parameter $\varphi = 0.0153$, and the EIS parameter $\psi = [1/40, 1/20, 1/\gamma, 1/2, 1/0, 75]$. We do not simulate $\psi = 1$ to avoid numerical instability. $\psi = 1$ reduces the problem to the CRRA utility case, $\psi = 1/\gamma$ reduces to the power utility case and $\psi = 1, \gamma = 1$ case reduces to the log utility case as noted in Campbell and Viceira, see [10]. We refer the reader to Campbell and Viceira, see [11], or a comprehensive summary on portfolio choice and long-term investments. Campbell and Viceira, see [10], show that the expected utility is only maximized if the solution to Equation (7.34) is given by the positive square root of the discriminant.

We begin our analysis by looking at the optimal stock and bond allocation as a function of EIS and ambiguity. In Fig.7.1(a) shows us that in general, the change in EIS has a small decreasing affect on the optimal stock allocation. This is due to the decreasing effects of EIS on stock hedging demands for moderately risk averse investors. The investors preference for

robustness causes the optimal stock allocation to decrease. In Fig.7.1(b) the effects of EIS are quite profound. At low levels of EIS ($\psi < 1$) we observe that the investor that learns about stock returns, monotonically increases their short position in bonds as EIS increases, but the preference to short bonds decreases(absolute) as the investor becomes more robust. At moderate to high levels of EIS ($\psi > 1$) the investor takes on less risky positions and their willingness to substitute increases, as observed by Campbell and Viceira, see [10]. We observe that as the preference for robustness increases the investor decreases their investments in the risky assets. This is explained further by Maenhout's observations as follows.

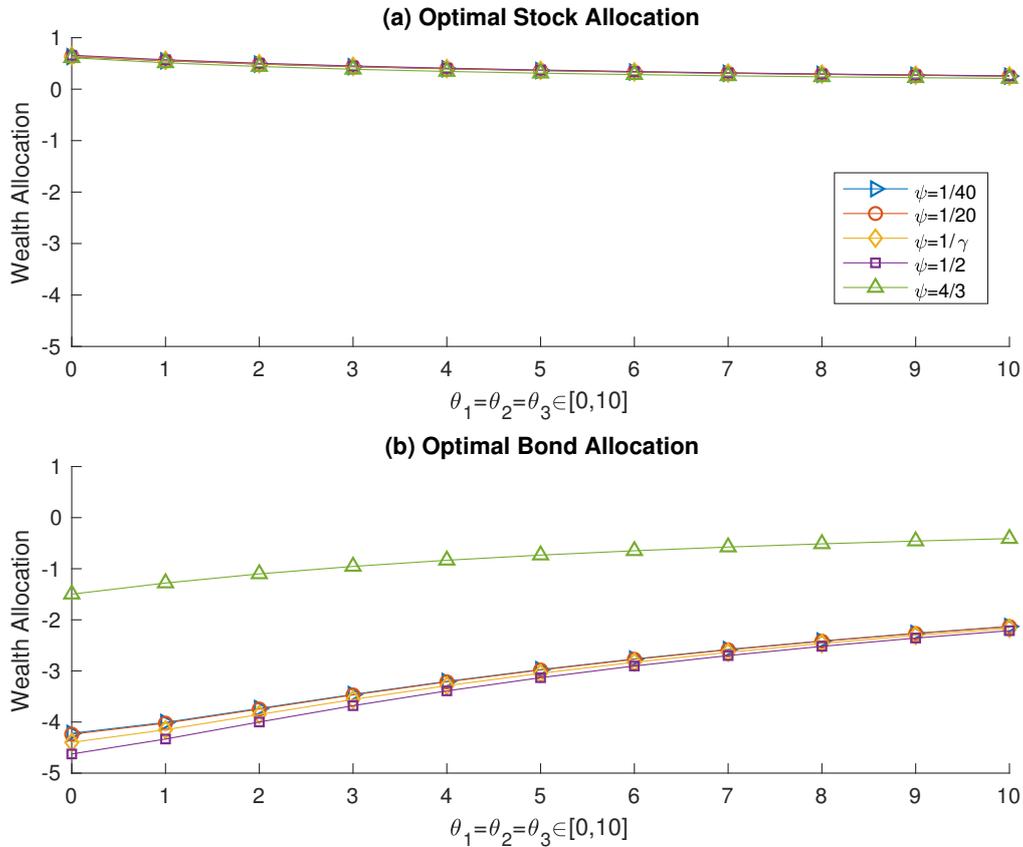


Figure 7.1: (a) optimal stock allocation, (b) optimal bond allocation

Maenhout, see [32], observed that as EIS increases the investors preference to consume decreases their positions in risky assets. This explains the sudden decrease(absolute) in the short position in the bond and stock demand and the sudden change in allocation was also observed in empirical data by Maenhout, he observed that as EIS increases, the risk aversion parameter becomes very large to keep the portfolio within 100%. Maenhout attributes this behaviour to 'effective' risk aversion. He notes that as EIS increases, the ambiguity aversion plus risk aversion effectively becomes the risk aversion and this explains the decrease in equity demand as robustness increases as shown in Figure 7.1. This relationship between EIS and 'effective' risk aversion also supports

the observation that for higher EIS, the investor aggressively decreases (absolute) short positions and slightly decrease their stock allocation as ambiguity aversion increases, as shown in Figure 7.1.

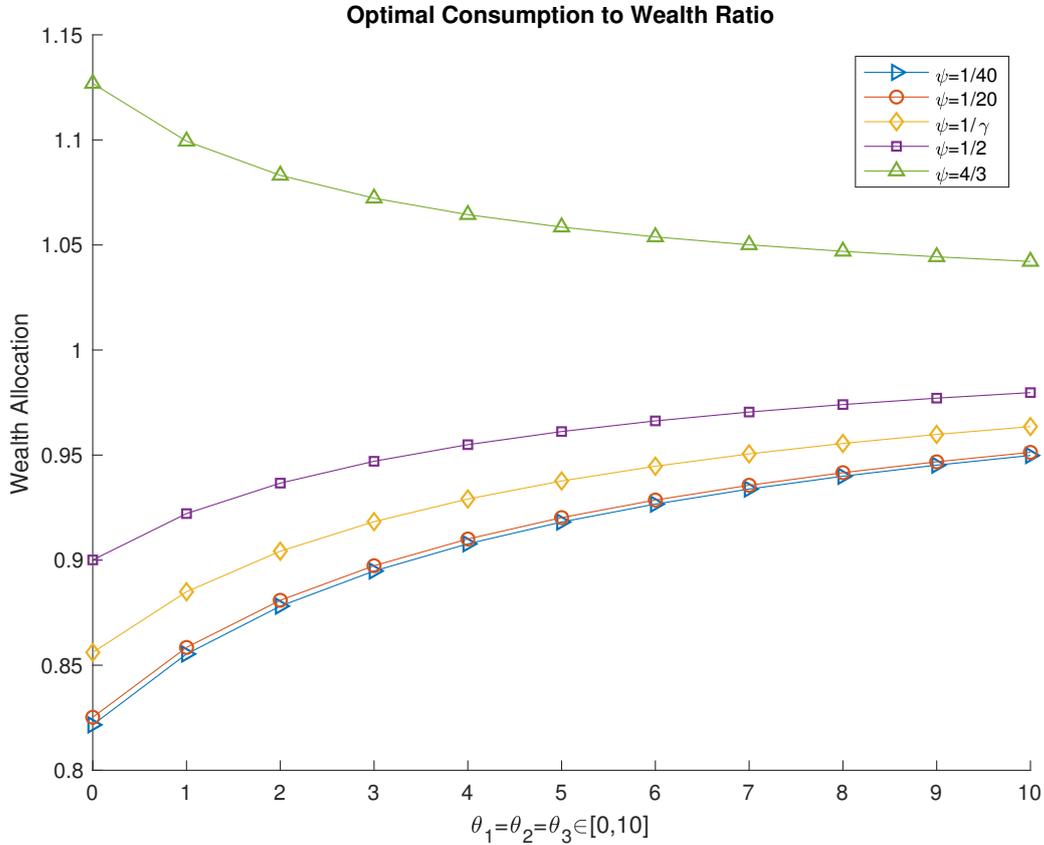


Figure 7.2: Relationship between consumption-wealth ratio, ambiguity and EIS

Campbell and Viceira, see [10], observed that at moderate and high levels of risk aversion, consumption increases as EIS increases, as shown in Figure 7.2. We observe that for $\psi < 1$, there is a concave behavior in the consumption-wealth ratio as ambiguity increases, conversely for $\psi > 1$, the consumption-wealth ratio shows convex behavior, this behavior was also observed by Ju and Miao, see [26]. This behavior is attributed to observation that for $\psi > 1$, the investor becomes more willing to substitute consumption with savings thus the investor at moderate levels of risk aversion seeks safer investments as shown in Figure 7.1. The effective increase in risk aversion causes the investor to invest in safer investments decreasing their consumption-wealth ratio, the opposite behavior is observed for $\psi < 1$.

The effects of learning at low levels of ambiguity aversion, increases the willingness of the investor to short the bond for all levels of ψ , this is observed in Fig.7.1(b). This is explained by the fact that learning encourages the investor to short the bond to increase their consumption-

wealth ratio, but as ambiguity aversion grows this willingness diminishes. The change in EIS, is negligible in the myopic demand. However, we observe that the myopic stock demand decreases with increased preference for robustness as shown in Figure 7.3. The same affect was observed by Liu, see [31], this affect goes back to Maenhout's observation that the preference for robustness effectively increases risk aversion.

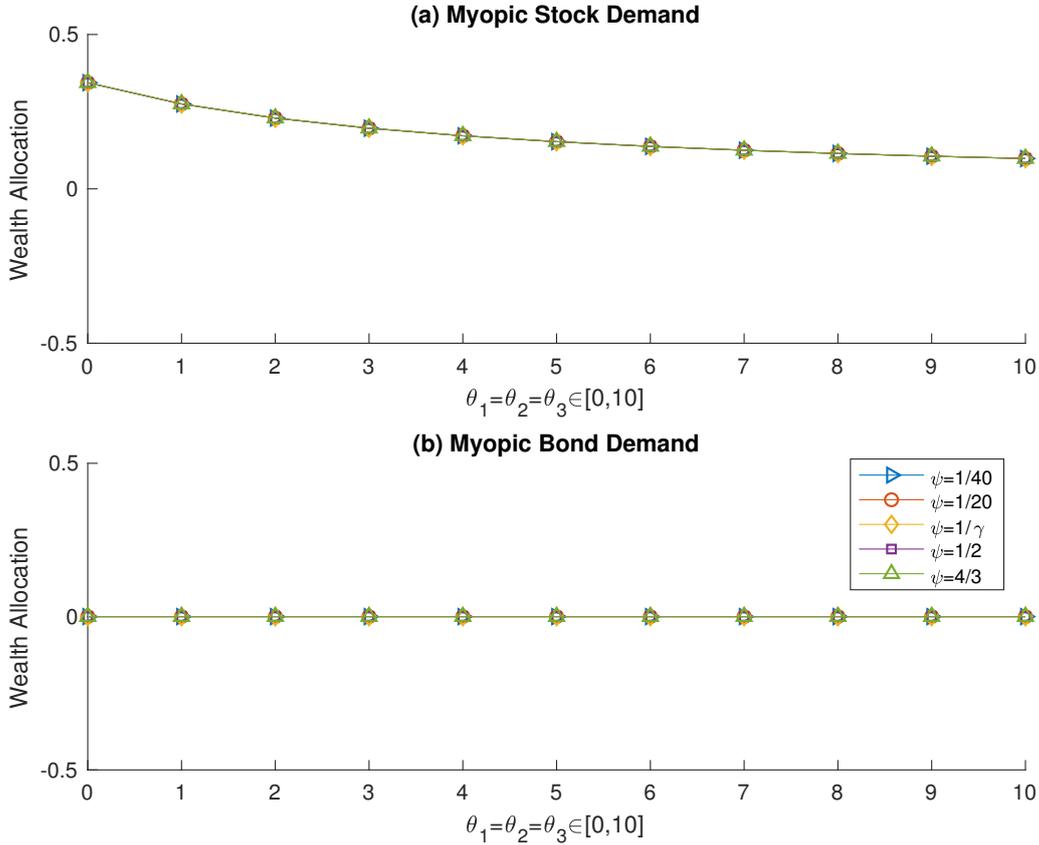


Figure 7.3: (a) Myopic stock allocation, (b) Myopic optimal bond allocation

We say that Myopic demand in the portfolio, as shown in Figure 7.3 is equivalent to the optimal portfolio derived from the mean-variance optimal portfolio problem solved by Merton, see [33]. At first glance it appears that Figures 7.1, 7.4 and 7.5, investors behave in a counter intuitive manner. However, our result is in line with observations from Campbell and Viceira, see [10], who also observe that for $\psi < 1$ the increase in EIS, increases the investors demand for higher returns. We can see this is exactly the case, for $\psi < 1$, we see that an investor who is less willing to substitute inter-temporally shorts the bond, but not at the level of the investors who are less more willing to substitute inter-temporally. This is also reflected in the optimal consumption to wealth ratio, which increases for moderate to high risk aversion as EIS increases, shown in Figure 7.2. Holding wealth constant, we observe from Figure 7.2 that the investors consumption is less at $\psi = [1/20, 1/40]$ compared to $\psi = [1/4, 1/2]$, this means that they are less willing to

substitute consumption for investments in risky or risk free asset. This explains why an investors with $\psi = [1/4, 1/2]$ is more willing to take riskier positions over investors with $\psi = [1/40, 1/20]$ as shown in Figure 7.1

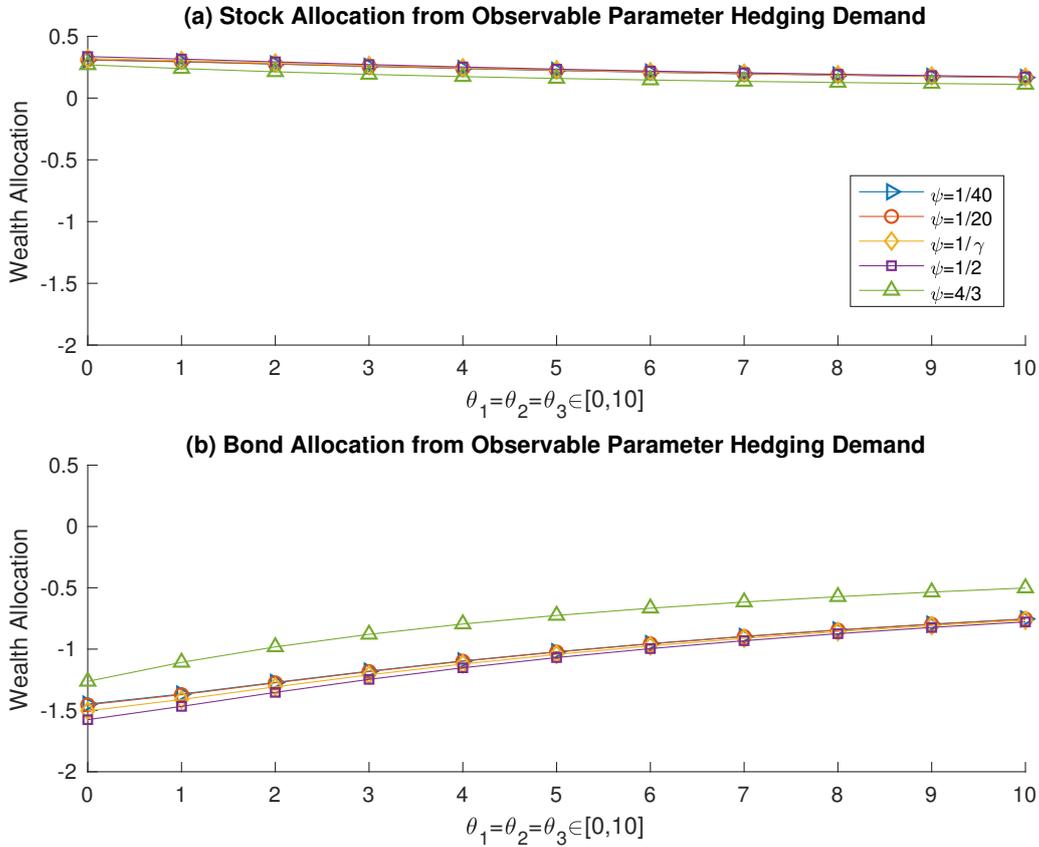


Figure 7.4: (a) Stock demand due to observable parameter hedging, (b) Bond demand due to observable parameter hedging

The equity demand coming from the observable parameter, Fig.7.4(a), is non zero. This is attributed to the assumption that this parameter is used to predict stock returns and stock volatility which implies the long run average of the observable parameter is not zero. For an investor with $\psi < 1$, that has no preference for robustness, the equity demand increases with EIS. For an investor with $\psi > 1$ the decrease in equity demand is due to the willingness to substitute as observed by Maenhout, see [32]. However, we observe that as preference for robustness increases any difference in equity demand vanishes rapidly and overall demand decreases. For the bond (Fig.7.4(b)), we observe that an investor who learns, shorts the bond, regardless of EIS, ψ . The observable parameter contributes about a third of the demand for shorting the bond. In Figure 7.5 we observe that the investor assumes more risk due to the unobservable parameter. Filtering about the unobservable parameter creates the largest demand to short the bond, learning about this parameter contributes the remaining two-thirds of the demand for

shorting the bond. The stock allocation from the unobservable parameter hedging demand is nearly non-existent, this is because in the long run average of the unobservable parameter is zero, this makes sense because the mean of the log process is assumed to go to zero in the long run. As observed by Escobar et al, see [20], the bond component of the hedge of the observed and unobserved components is much higher than the stock component and depends on the correlation values $\rho_\lambda P$ and $\rho_P \beta$. The investors preference to short the bond was explained by Escobar et al, see [20], through the following example; if $\rho_\lambda P > 0$ and $\rho_P \beta > 0$, then the bond tends to have higher returns when future stock risk premium is expected to worsen, and thus, the bond provides a hedge against this uncertainty in future investment opportunities. The investors value this hedge by investing more in bonds ($\phi_{P,obs} < 0$ and $\pi_{P,unobs} < 0$).

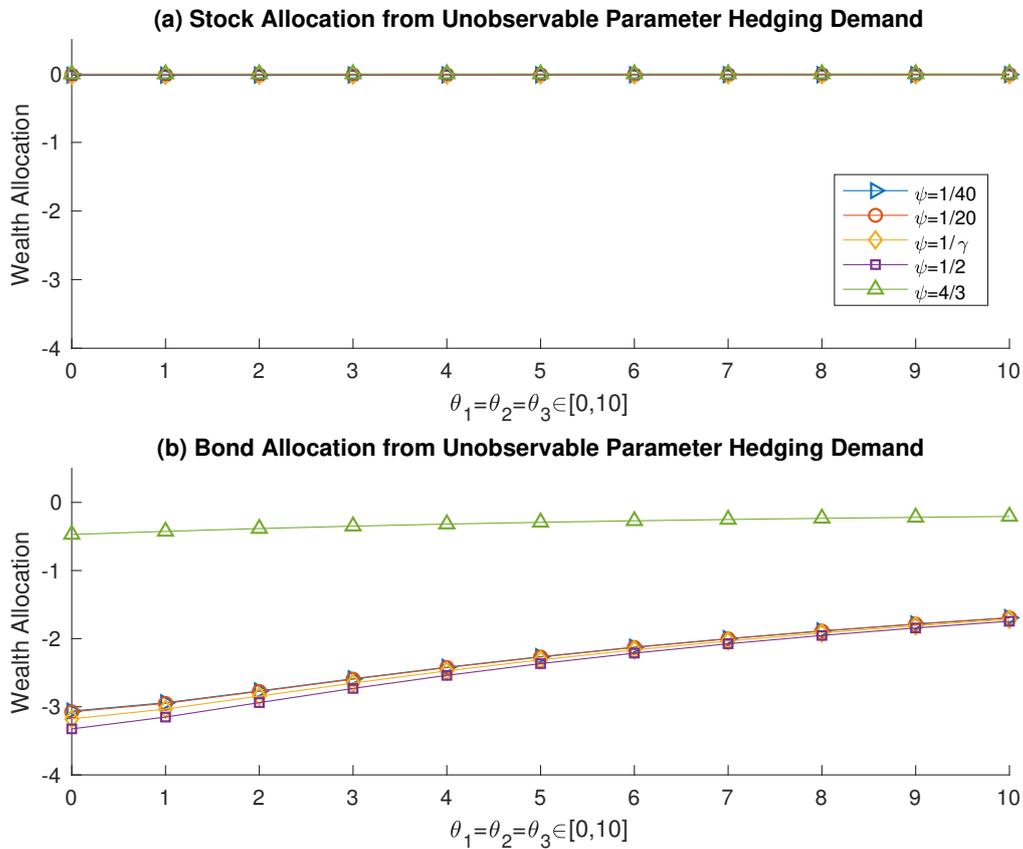


Figure 7.5: (a) Stock demand due to unobservable parameter hedging, (b) Bond demand due to unobservable parameter hedging

The initial convex behavior of the observable and unobservable hedging demand are due to the negative correlation they have with the stock return. This effect was a key result of Liu, see [31]. We can see this behaviour appearing for an investor that learns about stock return through the unobservable and observable hedging demands. Whereas, we don't see this behaviour in the interest rate hedging demand in Figure 7.6. However, as the investors preference for robustness

increases the effects of correlation vanishes. This is due to the diluting influence of the correlation coefficients as ambiguity aversion increases.

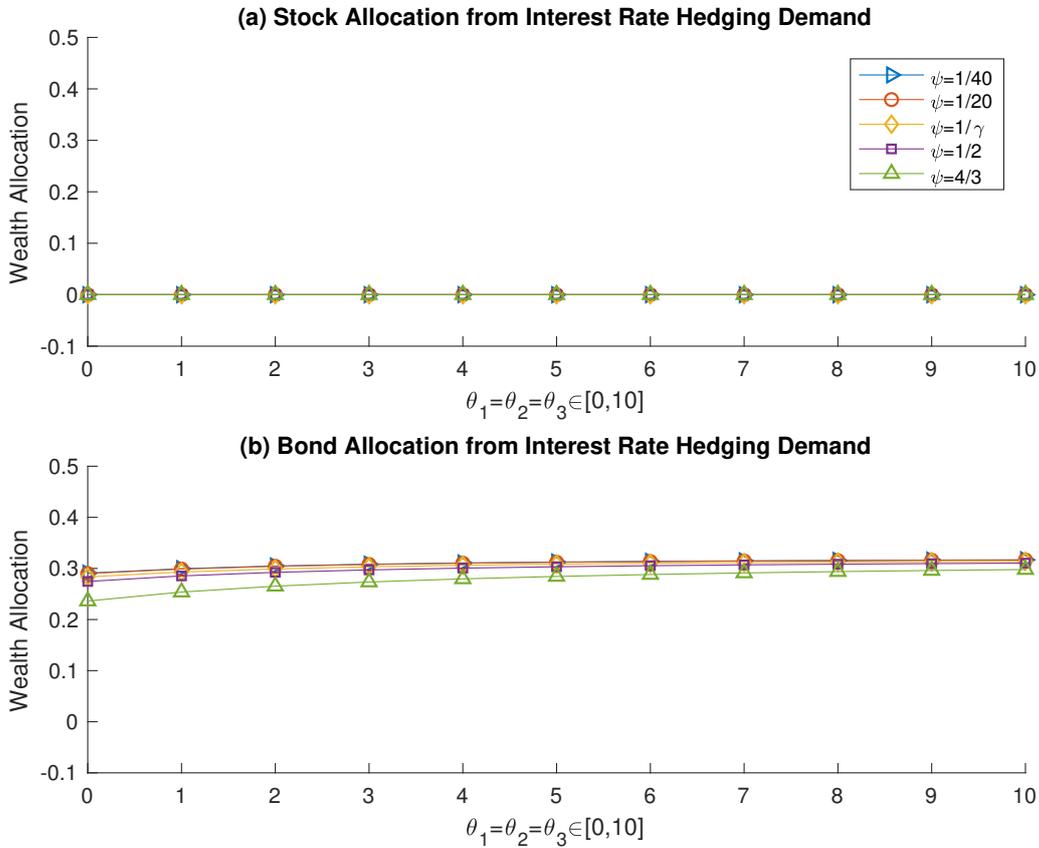


Figure 7.6: (a) Stock demand due to interest rate hedging, (b) Bond demand due to interest rate hedging

Maenhout, see [32], notes that as the preference for robustness increases the risk free rate decreases. We can see this behaviour reflected in Fig.7.6(b). The investor recognizes that the risk free rate changes randomly and hedges the risk purely using bonds. This is due to the perfectly negative correlation between the bond and interest rate, see [20]. We observe that as the EIS increases, the hedging demand for the bond to hedge interest rate risk decreases. This is explained by the increase in consumption of the investor at the moderate to high levels of risk aversion. However this is not enough to offset the investors willingness to short the bond. For $\psi > 1$ as ambiguity aversion increases, the effective risk aversion increases and drastically lowers the demand to short the bond, this supports the hypothesis proposed by Maenhout, see [32], which says that ambiguity aversion adds to the affect of risk aversion.

Chapter 8

Conclusion

Pricing and hedging has been a topic extensively studied in literature. There are many different viewpoints on how to model the movement of underlying assets. We propose a parametric model using the techniques of time-changed subordination, applied to Brownian Motion. This adds stochastic volatility in a tractable manner. Our model builds in inherent dependence between any number of assets in a portfolio. Intuitively this is equivalent to saying the underlying assets have a common pricing factor. The parametric nature of the model also let's us use heavy-tailed distributions to capture heavy tailedness in the data. For future work we propose incorporating learning about underlying returns into the model. This model is well suited to model incomplete information as it is a conditionally normal and fits well into the non-linear optimal filtering framework in Liptser and Shiryaev, see [30].

We show that our model can be priced using classical Monte Carlo method and the more efficient Fourier-Cosine method. We compare the two methods in pricing Arithmetic average Asian call options. We show that the Fourier-Cosine method is a very efficient method, but we noticed some instability of the method at a strike price of \$0, this is due to the solution being imaginary at that point. However, in practice nobody trades at a strike price at \$0, so it is not an issue for standard options. When used in calibration, this method does fails to reach the error tolerance of 1×10^{-6} . We suggest that this method be used with a grain of salt. It is a highly efficient method, but it is not as robust as Monte Carlo and it is sensitive to the integration bounds.

We use the Fourier-Cosine method to replicate option payoffs for quadratic risk, VaR risk and AVaR risk. We simulate the hedging strategies numerically using MC method as a benchmark and compare it to results based on the Fourier-Cosine method. The Fourier-Cosine method is tractable at determining cost functions under expectations. Thus it is quite successful at hedg-

ing quadratic and AVaR risk, however, it does not perform as well with VaR. This is because the Fourier-Cosine method was derived with the intention to compute expectations, and not a specific points in the distribution, as the sinusoidal nature makes it difficult to find a unique point. The Fourier-Cosine method does not explore the entire domain of the characteristic function and requires higher truncation bounds to perform more accurately for quadratic hedging problems.

In this work we show that hedging VaR is equivalent to creating a replicating portfolio for the smallest positive hedge that results in a positive hedge at a high probability. Table 5.3 shows that when hedging VaR the investor is only required to hold a small portion of wealth in the underlying and a large amount in cash. This implies that to hedge VaR risk, we can simply hedge the option by holding mostly cash. Counter intuitively, hedging AVaR requires a more risky portfolio as seen in Table 5.4. However, this is to be expected because to hedge a large error, simply holding cash cannot provide sufficient returns. We also look at the discounted expected hedging errors in Table 5.5. For an investor looking to hedge VaR they should expect their returns to reflect the payoff of the option as most of their hedging investments are allocated to the cash account and will at most lose a fraction of any negative movements in the underlying asset. The investor hedging AVaR is actually more exposed to risk as they need to generate higher returns to hedge risk in the entire tail. We also extend AVaR hedging to a robust framework using the Neyman-Pearson lemma as outlined in Föllmer et al, see [23]. For future work it would be interesting to extend the work to include more assets in the portfolio, consider stochastic interest rates and simulate the robust AVaR hedging portfolio.

The appearance of the volatility smile after the crash of 1987 created a new problem to explore. The cause the volatility smile is still not clear, but new models were proposed to capture the phenomenon. We show through numerical example that our model can replicate the volatility smile. We also propose a conditional calibration method to calibrate parameters. The calibration problem is reduced to two separate optimization problems. To solve this problem we use the pattern search algorithm. This direct method is known to have a global optimal point if the cost function is convex as shown by Torczon, see [46]. We show that with the RMSE cost function coupled with the pattern search algorithm effectively calibrates our model and matches market prices of WTI Asian options and Brent crude European options.

In the second part of this thesis, we explore the behavior of robust infinitely-lived investor

who maximizes their recursive utility. We also assume that the investor learns about the stock returns. We performed our simulation using the method found in Campand Viceira, see [10]. From our numerical study, we found that, generally, as the EIS increases the equity demand decreases, however the effect is small, and as preference for robustness increases the demand for equity decreases. This is due to the preference for robustness adding more risk aversion to the behaviour of the investor, effectively increasing risk aversion. It was observed that when $\psi > 1$, the investor decreases (absolute) short positions in riskless assets. We also observed that for any level of EIS, as ambiguity aversion decreases (absolute) short positions in the risk free asset. This is due to the investor becoming more risk averse. When $\psi < 1$, the investor increases short positions in riskless assets as EIS increases, in this case ambiguity aversion behaves the same as when $\psi > 1$.

We found that for $\psi > 1$ there is a convex behavior in consumption-wealth ratio as ambiguity increases. This is because the investor is less willing to seek risky investments and decreases their consumption-wealth ratio. Conversely for $\psi < 1$ there is a concave behavior, because when $\psi < 1$ the investor is more willing to seek higher returns, but the preference for robustness diminishes this willingness to take risk. For moderate levels of risk aversion, consumption-wealth ratio increases as EIS increases and like Liu, see [31], we observed that as ambiguity increases, the myopic demand for stock decreases. We also found that learning about stock returns drives the investor to short the riskless asset regardless of ψ . The majority of the demand for shorting comes from the unobservable parameter, which is about two-thirds of the demand, while the observable parameter contributes one-third. Interest rate risk, creates a hedging demand for the bond, but the hedging demand is not enough to overcome the willingness to short the bonds. For future work, it would be interesting to consider the effects of inflation, this is quite relevant in long-term investing, as inflation is no longer constant. It would also be interesting to look at the general utility proposed by Jun and Miao, see [26].

Appendix A: Matlab Code

Pricing Asian Options

MC Pricing

Time-changed Univariate Main Code

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %                               The is the main script                               %
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 clear ;
5 clc ;
6 tic
7 %%
8 % Initialize Variables
9 M = 250; % sampling frequency
10 T = 1; % maturity time in years
11 dT = T/M; % sampling period
12 N = 100000; % number of simulations
13 rate = 0.0367; % risk free rate of return in %*100
14 S0 = 57; % initial stock price of two assets
15 K = 1:100; % Strike
16 KSpread = 52; % Strike for Spread
17 time = transpose(linspace(0,T,T/dT));
18 k = 1; l = 1;
19 fprintf('Total number of simulations: %i\n',N)
20 %%
21 % Parameters from calibration
22 muY = 0; % drift parameter
23 sigmaY = 0.02; % volatitlity
24 kappa = 0.1;
25 c = 0;
26 %%
```

```

27 % IG Subordinated Process
28 % Generate dependent IG random time process
29 [R1,L0] = IGProc(kappa,c,dT,T,N,0);
30 % Generate NIG process
31 [S1,dY1,Y1] = NIGProc(R1,S0,muY,sigmaY);
32 fprintf('Done IG Sub\n')
33 %%
34 % Gamma Subordinated Process
35 % Generate dependent Gamama random time process
36 [R2,G0] = GammaProc(kappa,c,dT,T,N,0);
37 % Generate VG process
38 [S2,dY2,Y2] = VGProc(R2,S0,muY,sigmaY);
39 fprintf('Done Gamma Sub\n')
40 %%
41 % Geometric Brownian Motion Process
42 % Z = normrnd(0,1,[M,N]);
43 % dY3 = muY*dT+sqrt(dT)*sigmaY.*Z;
44 % Y3 = cumsum(dY3);
45 % S3 = S0*exp(Y3);
46 %%
47 % Asian option payoff for each strike price
48 Payoff1 = zeros(1,length(K)-1); Payoff2 = Payoff1; %BlS = Payoff1;
49 while k < length(K)+1
50     % Generating Asian option price(p) and payoff(h)
51     [Payoff1(k)] = AsianOption(S1,K(k),rate,dT);
52     [Payoff2(k)] = AsianOption(S2,K(k),rate,dT);
53     % Bls(k) = blsprice(S0,K(k),rate,dT,sigmaY);
54     % Generating Spread option price(p) and payoff(h)
55     [H(:,k)] = SpOpt(S1,S2,KSread(k),rate,dT,N);
56     k = k+1;
57 end
58 fprintf('Done Payoff calculations\n')
59 toc
60
61 %%
62 % Save results
63 % save('MCTimeChangedData.mat',...
64 %     'K','time','Payoff1','Payoff2')
65 % fprintf('Variables saved!\n')
66
67 % save('MCTimeChangedDataCI.mat','H1','S1','dY1','Y1','R1','H2','S2',...
68 %     'dY2','Y2','R2','H','K','KSread','N','time')

```

```

69 %fprintf('More variables saved!\n')
70 %%
71 % GPU Stuff
72
73 % if M <= 1000
74 %     N = M;
75 %     % Parameters to Generate IG random numbers
76 %     alpha0 = 6.188;
77 %     beta0 = -3.894;
78 %     delta0 = 0.1622;
79 %     alpha = alpha0;
80 %     beta = beta0;
81 %     delta = delta0;
82 %     a = [1 1 1];
83 %     b = [delta0*sqrt(alpha0^2-beta0^2), delta*sqrt(alpha^2-beta^2), delta*sqrt(
alpha^2-beta^2)];
84 %     c = [1 1];
85 %     % Generate dependent IG subordinators
86 %     [R1,L0] = IGProc(a,b,c(1),dT,T,N,0,1);
87 %     [R2] = IGProc(a,b,c(2),dT,T,N,L0,2);
88 %     R = [R1;R2];
89 %     % Generate NIG processes
90 %     covariance = [delta^2 delta*delta; delta*delta delta^2];
91 %     drift = [beta*delta^2, ...
92 %             beta*delta^2];
93 %     [S1] = NIGProc(R, initialStockPrices, drift(1), covariance(1,:), T, dT, N);
94 %     [S2] = NIGProc(R, initialStockPrices, drift(2), covariance(2,:), T, dT, N);
95 %     % Generating Asian option price(p) and payoff(h)
96 %     [P1] = AsianOption(S1, strikePrice, rate, T, dT, N);
97 %     [P2] = AsianOption(S2, strikePrice, rate, T, dT, N);
98 %     % Generating Asian Spread option price(p) and payoff(h)
99 %     [P] = AsianSpOpt(S1, S2, spreadStrikePrice, rate, T, dT, N);
100 %     s1 = gather(S1);
101 %     s2 = gather(S2);
102 %     p1 = gather(P1);
103 %     p2 = gather(P2);
104 %     p = gather(P);
105 %     reset(dev);
106 % end

```

Time-changed Bivariate Main Code

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %                               The is the main script                               %
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 clear;
5 clc;
6 tic
7 %%
8 % Initialize Variables
9 M = 12; % sampling frequency
10 T = 1/75; % maturity time in years
11 dT = T/M; % sampling period
12 N = 1000000; % number of simulations
13 rate = 0.02; % risk free rate of return in %*100
14 S0 = [68.58;74.88]; % initial stock price of two assets
15 K = 39:0.5:48; % Strike
16 time = transpose(linspace(0,T,T/dT));
17 k = 1; l = 1;
18 fprintf('Total number of simulations: %i\n',N)
19 %%
20 %% Gamma Subordinated Process
21 %% Generate dependent Gamama random time process
22 % [R2,G0] = GammaProc(kappa,c,dT,T,N);
23 %% Generate VG process
24 % [S2,dY2,Y2] = VGProc(R2,S0_1,muY,sigmaY);
25 % fprintf('Done Gamma Sub\n')
26 %%
27 % Geometric Brownian Motion Process
28 % Z = normrnd(0,1,[M,N]);
29 % dY3 = muY*dT+sqrt(dT)*sigmaY.*Z;
30 % Y3 = cumsum(dY3);
31 % S3 = S0*exp(Y3);
32 %%
33 load('BivarY1.mat')
34 load('BivarX1.mat')
35 % Parameters from calibration
36 muX = [y(4,13:31);x(3,1:19)];
37 sigmaX = [y(5,13:31);x(4,1:19)];
38 kappa = [y(1,13:31);y(2,13:31);x(1,1:19)];
39 c = [y(3,13:31);x(2,1:19)];
40 muY = dT.*muX+c.*muX*dT;
41 sigmaY = sqrt(dT.*((kappa(1,:) + c.^2.*[kappa(2,);kappa(3,)]).*muX.^2 + ...
42 ([1;1] + c).*sigmaX.^2));

```

```

43 % Asian option payoff for each strike price
44 Payoff1 = zeros(1, length(K) -1); %Payoff2 = Payoff1; %BlS = Payoff1;
45 while k < length(K)+1
46     % IG Subordinated Process
47     % Generate dependent IG random time process
48     [R1, L0] = IGProc(kappa(:, k), c(1, k), dT, T, N, 1);
49     [R2] = IGProc(kappa(:, k), c(2, k), dT, T, N, 2, L0);
50     % Generate NIG process
51     [S1] = NIGProc(R1, S0(1), muY(1, k), sigmaX(1, k));
52     [S2] = NIGProc(R2, S0(2), muY(2, k), sigmaX(2, k));
53     fprintf('Done IG Sub\n')
54     % Generating Asian option price(p) and payoff(h)
55     [Payoff1(k)] = AsianSprOption(S1, S2, K(k), rate, dT, 0.85);
56     k = k+1;
57 end
58 fprintf('Done Payoff calculations\n')
59 toc
60
61 %%
62 % Save results
63 filename = sprintf('MCTCSprSim%d.mat', T);
64 save(filename, ...
65     'K', 'time', 'Payoff1')
66 fprintf('Variables saved!\n')
67
68 % save('MCTimeChangedDataCI.mat', 'H1', 'S1', 'dY1', 'Y1', 'R1', 'H2', 'S2', ...
69 %     'dY2', 'Y2', 'R2', 'H', 'K', 'KSpread', 'N', 'time')
70 %fprintf('More variables saved!\n')
71 %%
72 % GPU Stuff
73
74 % if M <= 1000
75 %     N = M;
76 %     % Parameters to Generate IG random numbers
77 %     alpha0 = 6.188;
78 %     beta0 = -3.894;
79 %     delta0 = 0.1622;
80 %     alpha = alpha0;
81 %     beta = beta0;
82 %     delta = delta0;
83 %     a = [1 1 1];
84 %     b = [delta0*sqrt(alpha0^2-beta0^2), delta*sqrt(alpha^2-beta^2), delta*sqrt(

```

```

    alpha^2-beta^2)];
85 %     c = [1 1];
86 %     % Generate dependent IG subordinators
87 %     [R1,L0] = IGProc(a,b,c(1),dT,T,N,0,1);
88 %     [R2] = IGProc(a,b,c(2),dT,T,N,L0,2);
89 %     R = [R1;R2];
90 %     % Generate NIG processes
91 %     covariance = [delta^2 delta*delta;delta*delta delta^2];
92 %     drift = [beta*delta^2,...
93 %             beta*delta^2];
94 %     [S1] = NIGProc(R,initialStockPrices,drift(1),covariance(1,:),T,dT,N);
95 %     [S2] = NIGProc(R,initialStockPrices,drift(2),covariance(2,:),T,dT,N);
96 %     % Generating Asian option price(p) and payoff(h)
97 %     [P1] = AsianOption(S1,strikePrice,rate,T,dT,N);
98 %     [P2] = AsianOption(S2,strikePrice,rate,T,dT,N);
99 %     % Generating Asian Spread option price(p) and payoff(h)
100 %     [P] = AsianSpOpt(S1,S2,spreadStrikePrice,rate,T,dT,N);
101 %     s1 = gather(S1);
102 %     s2 = gather(S2);
103 %     p1 = gather(P1);
104 %     p2 = gather(P2);
105 %     p = gather(P);
106 %     reset(dev);
107 % end

```

IG Random Number Generator

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % This function generates IG random numbers. I returns the IG generated %
3 % numbers V returns the Chi squared generated numbers and Y returns the %
4 % uniformly generated numbers. a and b are the IG parameters, n is the %
5 % number of subordinators and m is the number of simulations. %
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 function [I] = IGRNG(a,b,n,m)
8     % Initialize variables
9     a_ones = ones(1,m); b_ones = ones(1,m);
10    a = (transpose(a)*a_ones); b = (transpose(b)*b_ones);
11    % Schucany's algorithm to generate IG random numbers
12    V = (normrnd(0,1,[n m])).^2; % generates a chi squared n by m array
13    W = a.*V;
14    C = a./(2.*b);
15    X = a + C.*(W - sqrt(W.*(4.*b + W)));

```

```

16     P = a./(a+ones(n,m)); Y = unifrnd(0,1,[n,m]);
17 %     Yg = gather(Y); Pg = gather(P); %Only required for gpu arrays
18     if Y >= P
19         I = (a.^2)./X;
20     else
21         I = X;
22     end
23 end

```

IG Process Generator

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %           This function will generate the IG dist random time           %
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 function [R,L0] = IGProc(kappa,c,dT,T,N,j,L0)
5     a0 = dT/sqrt(kappa(1)); b0 = 1/sqrt(kappa(1));
6     if j>0
7         aj = dT/sqrt(kappa(j+1)); bj = 1/sqrt(kappa(j+1));
8     else
9         aj=a0; bj=b0;
10    end
11    if nargin < 7
12 %    pd = makedist('InverseGaussian','mu',a,'lambda',b);
13    L0 = zeros(uint64(T/dT),N); L = L0;
14    n = length(a0); m = N; i = 1; t = 0;
15    while t < T-dT
16        t = t+dT;
17 %        X = random(pd,[n,m]); Y = random(pd,[n,m]);
18        X = IGRNG(a0,b0,n,m); Y = IGRNG(aj,bj,n,m);
19    % Independent IG Processes
20        L0(i+1,:) = L0(i,:)+X(1,:);
21        L(i+1,:) = L(i,:)+Y(1,:);
22        i = i+1;
23    end
24    % Dependent IG Processes
25        R = L0 + c*L;
26    else
27 %    pd = makedist('InverseGaussian','mu',a,'lambda',b);
28    L = zeros(uint64(T/dT),N);
29    n = length(a0); m = N; i = 1; t = 0;
30    while t < T-dT
31        t = t+dT;

```

```

32 %           X = random(pd,[n,m]); Y = random(pd,[n,m]);
33           X = IGRNG(a0,b0,n,m); Y = IGRNG(aj,bj,n,m);
34 % Independent IG Processes
35           L(i+1,:) = L(i,:)+Y(1,:);
36           i = i+1;
37           end
38 % Dependent IG Processes
39           R = L0 + c*L;
40           end
41 end

```

NIG Process Generator

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %           This generates a IG subordinated time changed process
3 %
4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5 function [S,dY,Y,dW]=NIGProc(R,S0,mu,sigma)
6     n = size(R,1); m = size(R,2);
7     Z = normrnd(0,1,[1,m]);
8     dR = [zeros(m,1),diff(R)'];
9     dR =dR';
10    dY = sigma*sqrt(dR).*Z+...
11         dR*mu;
12    dW = sigma*sqrt(dR).*Z-...
13         dR*sigma;
14    Y = cumsum(dY);
15    S = S0*exp(Y);
16 end

```

Gamma Random Number Generator

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % This function generates Gamma random numbers
3 %
4 % U and V return uniformly generated random numbers
5 %
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 function [G] = GammaRNG(a,n,m)
8     % Initialize variables
9     a_ones = ones(1,m);
10    a = (transpose(a)*a_ones);

```

```

9      X = ones(n,m); Y = ones(n,m);
10     % Jonk's algorithm to generate Gamma distributed random numbers
11     while X+Y > 1
12         U = unifrnd(0,1,[n,m]); V = unifrnd(0,1,[n,m]);
13         X = U.^(1./a); Y = V.^(1./(1-a));
14     end
15     Z = -log(unifrnd(0,1,[n,m]));
16     G = Z.*X./(X+Y);
17 end

```

Gamma Process Generator

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %           This function will generate the Gamma dist random time           %
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 function [R,G0] = GammaProc(kappa ,c ,dT,T,N,j)
5     a0 = dT/kappa(1);
6     if length(kappa)>1
7         aj=dT/kappa(j+1);
8     else
9         aj=a0;
10    end
11    % pd = makedist('Gamma','a',a);
12    G0 = zeros(T/dT,N); G = zeros(T/dT,N);
13    n = length(a0); m = N; i = 1; t = 0;
14    while t < T-dT
15        t = t+dT;
16    % X = random(pd,[n,m]); Y = random(pd,[n,m]);
17        X = GammaRNG(a0,n,m); Y = GammaRNG(aj,n,m);
18    % Independent IG Processes
19        G0(i+1,:) = G0(i,:)+X(1,:);
20        G(i+1,:) = G(i,:)+Y(1,:);
21        i = i+1;
22    end
23    % Dependent IG Processes
24        R = G0 + c*G;
25 end
26

```

VG Process Generator

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

2 %           This generates a Gamma subordinated time changed process
3           %
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 function [S,dY,Y,dW]=VGProc(R,S0,mu,sigma,L,j,l)
5     n = size(R,1); m = size(R,2); dY=zeros(n,m);
6     if nargin <= 6
7         Z = normrnd(0,1,[1,m]);
8         dR = [zeros(m,1),diff(R)'];
9         dR =dR';
10        dY = sigma*sqrt(dR).*Z+...
11            dR*mu;
12        dW = sigma*sqrt(dR).*Z-...
13            dR*sigma;
14    elseif nargin > 6
15        % j is the row index of covariance matrix
16        % l is the column index of jth row that corresponds to the other stock
17        sigma1 = sigma(j,j); sigma2 = sigma(j,l);
18        Z = normrnd(0,1,[1,m]);
19        dR = [zeros(m,1),diff(R)'];
20        dR =dR';
21        dL = [zeros(m,1),diff(L)'];
22        dL =dL';
23        dY = sigma1*sqrt(dR).*Z+...
24            sigma2*sqrt(dL).*Z+...
25            dR*mu(j);
26        dW = sigma*sqrt(dR).*Z-...
27            dR*sigma;
28    end
29    Y = cumsum(dY);
30    S = S0*exp(Y);
31 end

```

Asian Option Payoff

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %           This computes Asian Option Price           %
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 function [payoff] = AsianOption(S,K,r,dT)
5     % option payoff
6     h = mean(max(mean(S)-K,0));
7     % option discounted value
8     payoff = exp(-r*(dT))*h;

```

```
9 end
```

Asian Basket Spread Option Payoff

```
1 function [payoff] = AsianSprOption(S1,S2,K,r,dT,w)
2     % option payoff
3     h = mean(max((w*mean(S1)-(1-w)*mean(S2))-K,0));
4     % option discounted value
5     payoff = exp(-r*(dT))*h;
6 end
```

Hedging Asian Options

Quadratic Hedging

NIG Quadratic Hedging

```
1 clear;
2 clc;
3 format long
4
5 T=1;
6 dt=1/250;
7 M=T/dt;
8 S0=57;
9 r=0.0367;
10 N = 100000;
11 K = 52;
12 finalStrikePrice = K;
13
14 %% Hedging by MC
15 tic
16 k = 1; l = 1;
17 fprintf('Total number of simulations: %i\n',N)
18 % Parameters from calibration
19 muY = 0; % drift parameter
20 sigmaY = 0.02; % volatility
21 kappa = 0.1;
22 c = 0;
23 % IG Subordinated Process
24 % Generate dependent IG random time process
```

```

25 [R1,L0] = IGProc(kappa,c,dt,T,N);
26 % Generate NIG process
27 [S] = NIGProc(R1,S0,muY,sigmaY,T,dt);
28 V = max(mean(S)-K,0);
29 fprintf('Done IG Sub\n');
30
31 %% Hedging MC by replication
32 St = mean(S,2);
33 alpha = cov(S(end,:),V)/var(S(end,:));
34 beta = exp(-r*dt)*(mean(V)-alpha*St(end));
35 % wealth process
36 X = exp(-r*dt)*mean(mean(V))+exp(r*T).*beta+alpha.*St(end);
37 % determining optimal alpha and beta
38 J = mean((mean(V)-X).^2);
39 [L,I]=min(J(:));
40 [row, col] = ind2sub(size(J),I);
41 alpha = alpha(row,col);
42 beta = beta(row,col);
43 X = exp(-r*dt)*mean(mean(V))+exp(r*T).*beta+alpha.*St(end);
44 % fraction of wealth in stocks and bonds
45 pi_s = alpha./X;
46 pi_p = 1-pi_s;
47 % hedging error
48 RL = V - X;
49
50 toc
51
52 %% Hedging with COS Method
53 tic
54 fprintf('Start T=%i\n',T)
55 %Note: According to our error analysis, there is term N(error(q)), so when
56 %N goes up, N_I MUST BE INCREASED AT THE SAME TIME!
57 %Default values
58 N=256;
59 N_I=400;
60
61 % Initializing
62 tic;
63 i=complex(0,1);
64
65 q=0;
66

```

```

67 % Parameters from calibration
68 muX = 0; % drift parameter
69 sigmaX = 0.02; % volatitlity
70 kappa = 0.1;
71 c = 0;
72
73 a0 = 1/sqrt(kappa); b0 = 1/sqrt(kappa);
74 %% Cumulant and trunction range for NIG
75 c1 = dt*muX;
76 c2 = dt*(sigmaX^2+muX*kappa);
77 %
78 %% integration truncation
79
80 % L=10 converges well to the reference value. When increasing L, N, N.I
81 % should also be increased. And when L is very large , should use put-call
82 % parity .
83 L=10;
84 a=c1-L*sqrt(M*c2);
85 b=log(M)+M*c1+L*sqrt(M*c2);
86
87
88 k=0:N-1;
89 omega=k'*pi/(b-a);
90 v = i*muX*omega-0.5*omega.^2*sigmaX^2;
91 cf_Z=exp(-a0*dt*(sqrt(-2*i*(-i*v)+b0^2)-b0));
92 Int_Aasian=zeros(N, N);
93
94 % Only for Clenshaw-Curtis quadrature
95 Ni=0:2:N.I;
96 Ni=Ni';
97 d=2./(1-Ni.^2);
98 d(1)=1;
99 d(end)=0.5*d(end);
100 n=0:1:floor(N.I/2);
101 D=n'*n;
102 D=2/(N.I)*cos(D*2*pi/N.I);
103 D(:,1)=0.5*D(:,1);
104 D(:,end)=0.5*D(:,end);
105 w1=D'*d;
106
107 for k1=1:N-1
108     for k2=1:N-1

```

```

109         Int_Aasian (k1,k2) = asianmat(omega(k1), omega(k2), N-I, a, b, a, w1);
110     end
111 end
112
113 %% Only for Beta function
114 % for k1=1:N-1
115 %     for k2=1:N-1
116 %         Int_Aasian (k1,k2) = asianmat(k1, k2, a, b);
117 %     end
118 % end
119
120 cf_B=cf_Z;
121
122 for j=2:M
123     A=real(cf_B.*exp(-i*a*omega));
124     A(1)=0.5*A(1);
125     Phi_B=(2/(b-a))*Int_Aasian*A;
126     cf_B=cf_Z.*Phi_B;
127 end
128
129
130 % ep stand for 'exercise point'
131 ep=log(K*(M+1)/S0-1);
132
133 if ep>b
134     ep=b;
135 elseif ep<a
136     ep=a;
137 end
138
139 %% Hedging with stocks and bonds
140 U=(2./(b-a))*((S0./(M+1))*Chi(ep,b,N,a,b)-(K-(S0/(M+1)))*Psi(ep,b,N,a,b));
141 Re=real(cf_B.*exp(-i*a*omega));
142 % Asian option vlaue
143 H=U.*Re';
144 H(1)=0.5*H(1);
145 % Price process
146 Re=real(cf_Z.*exp(-i*a*omega));
147 % E[S0exp(Y)] = V.*Re'
148 V = (2./(b-a))*S0*Chi(a,b,N,a,b);
149 % stock price at t=T
150 S=V.*Re';

```

```

151 S(1)=0.5*S(1);
152 % FOC
153 alpha_cos = cov(H,S)/var(S);
154 beta_cos = exp(-r*dt)*(sum(H)-alpha_cos*sum(S));
155 % Wealth process
156 X_cos = exp(-r*dt)*sum(H)+exp(r*T).*beta_cos+alpha_cos.*sum(S);
157 % determing the minimum
158 J = (ones(2,2)*sum(H)-X_cos).^2;
159 [M,I]=min(J(:));
160 [row, col] = ind2sub(size(J),I);
161 % optimal stocks and bonds
162 alpha_cos = alpha_cos(row,col);
163 beta_cos = beta_cos(row,col);
164 X_cos = exp(-r*dt)*sum(H)+exp(r*T).*beta_cos+alpha_cos.*sum(S);
165 % fraction of wealth in stocks and bonds
166 pi_s_cos = alpha_cos/X_cos;
167 pi_p_cos = 1-pi_s_cos;
168 % hedging loss
169 RL_cos = sum(H)-(X_cos);
170 toc
171
172 fprintf('Done\n')
173 save('NIGQHedge.mat','RL','RL_cos','alpha',...
174     'beta','alpha_cos','beta_cos','pi_s','pi_p',...
175     'pi_s_cos','pi_p_cos','r','S0','dt','T','muX','K','kappa','sigmaX',...
176     'a','b')
177 fprintf('Variables saved!\n')

```

VG Quadratic Hedging

```

1 clear;
2 clc;
3 format long
4
5 T=1;
6 dt=1/250;
7 M=T/dt;
8 S0=57;
9 r=0.0367;
10 N = 100000;
11 K = 52;
12 finalStrikePrice = K;

```

```

13
14 %% Hedging by MC
15 tic
16 k = 1; l = 1;
17 fprintf('Total number of simulations: %i\n',N)
18 % Parameters from calibration
19 muY = 0; % drift parameter
20 sigmaY = 0.02; % volatitlity
21 kappa = 0.1;
22 c = 0;
23 % Gamma Subordinated Process
24 % Generate dependent Gamama random time process
25 [R,G0] = GammaProc(kappa , c , dt , T,N) ;
26 % Generate VG process
27 [S] = VGProc(R,S0 ,muY ,sigmaY ,T, dt) ;
28 V = max(mean(S)-K,0) ;
29 fprintf('Done Gamma Sub\n')
30
31 %% Hedging MC by replication
32 St = mean(S,2) ;
33 alpha = cov(S(end,:),V)/var(S(end,:)) ;
34 beta = exp(-r*dt)*(mean(V)-alpha*St(end)) ;
35 % wealth process
36 X = exp(-r*dt)*mean(mean(V))+exp(r*T).*beta+alpha.*St(end) ;
37 % determining optimal alpha and beta
38 J = mean((mean(V)-X).^2) ;
39 [L,I]=min(J(:)) ;
40 [row, col] = ind2sub(size(J),I) ;
41 alpha = alpha(row, col) ;
42 beta = beta(row, col) ;
43 X = exp(-r*dt)*mean(mean(V))+exp(r*T).*beta+alpha.*St(end) ;
44 % fraction of wealth in stocks and bonds
45 pi_s = alpha./X;
46 pi_p = 1-pi_s;
47 % hedging error
48 RL = V - X;
49
50 toc
51 %% Hedging with COS Method
52 tic
53 fprintf('Start T=%i\n',T)
54 %Note: According to our error analysis , there is term N(error(q)), so when

```

```

55 %N goes up, N_I MUST BE INCREASED AT THE SAME TIME!
56 %Default values
57 N=256;
58 N_I=400;
59
60 % Initializing
61 tic;
62 i=complex(0,1);
63 q=0;
64
65 % Parameters from calibration
66 muX = 0; % drift parameter
67 sigmaX = 0.02; % volatitlity
68 kappa = 0.1;
69 c = 0;
70
71 a0 = 1/kappa; b0 = 1/kappa;
72 %% Cumulant and trunction range for VG
73 c1 = dt*muX;
74 c2 = dt*(sigmaX^2+muX*kappa);
75 %
76 %% integration truncation
77
78 % L=10 converges well to the reference value. When increasing L, N, N_I
79 % should also be increased. And when L is very large, should use put-call
80 % parity.
81 L=10;
82 a=c1-L*sqrt(M*c2);
83 b=log(M)+M*c1+L*sqrt(M*c2);
84
85
86 k=0:N-1;
87 omega=k'*pi/(b-a);
88 v = i*muX*omega-0.5*omega.^2*sigmaX^2;
89 cf_Z=(1-i.*(-i*v)./b0).^(a0*dt);
90 Int_Asian=zeros(N, N);
91
92 % Only for Clenshaw-Curtis quadrature
93 Ni=0:2:N_I;
94 Ni=Ni';
95 d=2./(1-Ni.^2);
96 d(1)=1;

```

```

97 d(end)=0.5*d(end);
98 n=0:1:floor(N_I/2);
99 D=n'*n;
100 D=2/(N_I)*cos(D*2*pi/N_I);
101 D(:,1)=0.5*D(:,1);
102 D(:,end)=0.5*D(:,end);
103 w1=D'*d;
104
105 for k1=1:N-1
106     for k2=1:N-1
107         Int_Asian(k1,k2) = asianmat(omega(k1), omega(k2), N_I, a, b, a, w1);
108     end
109 end
110
111 %% Only for Beta function
112 % for k1=1:N-1
113 %     for k2=1:N-1
114 %         Int_Asian(k1,k2) = asianmat(k1, k2, a, b);
115 %     end
116 % end
117
118 cf_B=cf_Z;
119
120 for j=2:M
121     A=real(cf_B.*exp(-i*a*omega));
122     A(1)=0.5*A(1);
123     Phi_B=(2/(b-a))*Int_Asian*A;
124     cf_B=cf_Z.*Phi_B;
125 end
126
127
128 % ep stand for 'exercise point'
129 ep=log(K*(M+1)/S0-1);
130
131 if ep>b
132     ep=b;
133 elseif ep<a
134     ep=a;
135 end
136 %% Hedging with stocks and bonds
137 U=(2./(b-a))*((S0./(M+1))*Chi(ep,b,N,a,b)-(K-(S0/(M+1)))*Psi(ep,b,N,a,b));
138 Re=real(cf_B.*exp(-i*a*omega));

```

```

139 % Asian option vlaue
140 H=U.*Re';
141 H(1)=0.5*H(1);
142 % Price process
143 Re=real(cf_Z.*exp(-i*a*omega));
144 % E[S0exp(Y)] = V.*Re'
145 V = (2./(b-a))*S0*Chi(a,b,N,a,b);
146 % stock price at t=T
147 S=V.*Re';
148 S(1)=0.5*S(1);
149 % FOC
150 alpha_cos = cov(H,S)/var(S);
151 beta_cos = exp(-r*dt)*(sum(H)-alpha_cos*sum(S));
152 % Wealth process
153 X_cos = exp(-r*dt)*sum(H)+exp(r*T).*beta_cos+alpha_cos.*sum(S);
154 % determing the minimum
155 J = (ones(2,2)*sum(H)-X_cos).^2;
156 [M,I]=min(J(:));
157 [row, col] = ind2sub(size(J),I);
158 % optimal stocks and bonds
159 alpha_cos = alpha_cos(row,col);
160 beta_cos = beta_cos(row,col);
161 X_cos = exp(-r*dt)*sum(H)+exp(r*T).*beta_cos+alpha_cos.*sum(S);
162 % fraction of wealth in stocks and bonds
163 pi_s_cos = alpha_cos/X_cos;
164 pi_p_cos = 1-pi_s_cos;
165 % hedging loss
166 RL_cos = sum(H)-(X_cos);
167 toc
168
169 save('VGQHedge.mat','RL','RL_cos','alpha',...
170     'beta','alpha_cos','beta_cos','pi_s','pi_p',...
171     'pi_s_cos','pi_p_cos','r','S0','dt','T','muX','K','kappa','sigmaX',...
172     'a','b')
173 fprintf('Variables saved!\n')

```

Quantile Hedging

NIG Quantile Hedging

```
1 clear;
```

```

2  clc;
3  format long
4  %% Initial Parameters
5  T=1;
6  dt=1/250;
7  M=T/dt;
8  S0=57;
9  r=0.0367;
10 N = 100000;
11 K = 52;
12 eps = 0.99;
13
14 %% Hedging by MC
15 tic
16 fprintf('Total number of simulations: %i\n',N)
17 % Parameters from calibration
18 muY = 0; % drift parameter
19 sigmaY = 0.02; % volatitlity
20 kappa = 0.1;
21 c = 0;
22 % IG Subordinated Process
23 % Generate dependent IG random time process
24 [R1,L0] = IGProc(kappa,c,dt,T,N);
25 % Generate NIG process
26 [S] = NIGProc(R1,S0,muY,sigmaY,T,dt);
27 V = max(S-K,0);
28 ST = S(end,:);
29 VT = V(end,:);
30 V0 = exp(-r*T)*mean(mean(V));
31 fprintf('Done IG Sub\n')
32 %% VaR Hedging with MC by replication
33 % hedging portfolio
34 % initial portfolio given by C0 == 0
35 syms a
36 b = V0-a*S0;
37 x = b+a*S0;
38 c = x-V0;
39 system = c == 0;
40 var = a;
41 alpha = double(solve(system,var));
42 beta = V0-alpha*S0;
43 % wealth process

```

```

44 X = V0+exp(r*T).*beta+alpha.*ST;
45 % determining quantile index
46 C = exp(-r*T)*X-exp(-r*T)*VT;
47 [row, col] = find(C>=0);
48 A = VT(col);
49 [J, I] = sort(C(col), 'ascend');
50 A = A(I);
51 idx = floor(length(A)*(eps))+1;
52 dN = length(A(1:idx));
53 % alpha at C == 0
54 syms a
55 V1 = exp(-r*T)*A(idx);
56 b = V0-a*S0;
57 x = V1+exp(r*T)*b+a*mean(ST);
58 c = exp(-r*T)*x - V0;
59 system = c == 0;
60 var = a;
61 % optimal portfolio for VaR
62 alpha = double(solve(system, var));
63 beta = V0-alpha*S0;
64 X_var = V0+exp(r*T)*beta+alpha*mean(ST);
65 pi_s_var = alpha/X_var;
66 pi_p_var = 1-pi_s_var;
67 % Initial investments
68 H0_var = beta+alpha*S0;
69 % hedging error
70 RL_var = exp(-r*T)*(X_var-VT);
71
72 %% ES Hedging with MC by replication
73 % optimal portfolio for ES
74 % alpha at C>=0 from 1:idx-1
75 syms a
76 V2 = exp(-r*T)*(1/(dN*eps))*sum(A(1:idx-1))+V1;
77 b = V0-a*S0;
78 x = V2+exp(r*T)*b+a*mean(ST);
79 c = exp(-r*T)*x - V0;
80 system = c == 0;
81 var = a;
82 alpha_a = double(solve(system, var));
83 % optimal portfolio
84 alpha_es = alpha_a;
85 beta_es = V0-alpha_es*S0;

```

```

86 X_es = V0+exp(r*T)*beta_es+alpha_es*mean(ST);
87 pi_s_es = alpha_es/X_es;
88 pi_p_es = 1-pi_s_es;
89 % Initial investments
90 H0_es = beta_es+alpha_es*S0;
91 % hedging error
92 RL_es = exp(-r*T)*(X_es-VT);
93 toc
94 %% Hedging with COS Method
95 format long
96
97 T=1;
98 dt=1/250;
99 M=T/dt;
100 S0=57;
101 r=0.0367;
102 K = 52;
103 finalStrikePrice = K;
104 eps = 0.99;
105
106 fprintf('Start T=%i\n',T)
107 %Note: According to our error analysis, there is term N(error(q)), so when
108 %N goes up, N_I MUST BE INCREASED AT THE SAME TIME!
109 %Default values
110 N=256;
111 N_I=400;
112
113 % Initializing
114 tic;
115 i=complex(0,1);
116
117 q=0;
118
119 a0 = 1/sqrt(kappa); b0 = 1/sqrt(kappa);
120 %% Cumulant and truncation range for NIG
121 c1 = dt*muY;
122 c2 = dt*(sigmaY^2+muY*kappa);
123 %
124 %% integration truncation
125
126 % L=10 converges well to the reference value. When increasing L, N, N_I
127 % should also be increased. And when L is very large, should use put-call

```

```

128 % parity .
129 J=10;
130 a=c1-J*sqrt(M*c2);
131 b=log(M)+M*c1+J*sqrt(M*c2);
132
133
134 k=0:N-1;
135 omega=k'*pi/(b-a);
136 v = i*muY*omega-0.5*omega.^2*sigmaY^2;
137 cf_Z=exp(-a*dt*(sqrt(-2*i*(-i*v)+b0^2)-b0));
138 Int_Aasian=zeros(N, N);
139
140 % Only for Clenshaw-Curtis quadrature
141 Ni=0:2:N_I;
142 Ni=Ni';
143 d=2./(1-Ni.^2);
144 d(1)=1;
145 d(end)=0.5*d(end);
146 n=0:1:floor(N_I/2);
147 D=n'*n;
148 D=2/(N_I)*cos(D*2*pi/N_I);
149 D(:,1)=0.5*D(:,1);
150 D(:,end)=0.5*D(:,end);
151 w1=D'*d;
152
153 for k1=1:N-1
154     for k2=1:N-1
155         Int_Aasian(k1,k2) = asianmat(omega(k1), omega(k2), N_I, a, b, a, w1);
156     end
157 end
158
159 %% Only for Beta function
160 % for k1=1:N-1
161 %     for k2=1:N-1
162 %         Int_Aasian(k1,k2) = asianmat(k1, k2, a, b);
163 %     end
164 % end
165
166 cf_B=cf_Z;
167
168 for j=2:M
169     B=real(cf_B.*exp(-i*a*omega));

```

```

170     B(1)=0.5*B(1);
171     Phi_B=(2/(b-a))*Int_Asian*B;
172     cf_B=cf_Z.*Phi_B;
173 end
174
175
176 % ep stand for 'exercise point'
177 ep=log(K*(M+1)/S0-1);
178
179 if ep>b
180     ep=b;
181 elseif ep<a
182     ep=a;
183 end
184
185 %% VaR and ES Hedging with COS Method
186
187 U=(2./(b-a))*((S0./(M+1))*Chi(ep,b,N,a,b)-(K-(S0/(M+1)))*Psi(ep,b,N,a,b));
188 Re=real(cf_B.*exp(-i*a*omega));
189 % Asian option vlaue
190 H=U.*Re';
191 H(1)=0.5*H(1);
192 H0 = exp(-r*T)*sum(H);
193 % Price process
194 Re=real(cf_Z.*exp(-i*a*omega));
195 % E[S0exp(Y)] = V.*Re'
196 V = (2./(b-a))*S0*Chi(a,b,N,a,b);
197 % stock price at t=T
198 S=V.*Re';
199 S(1)=0.5*S(1);
200 % Fourier-Cosine of 1
201 W = (2./(b-a))*Psi(a,b,N,a,b);
202 F = real(exp(-i*a*omega)).*W;
203 F(1)=0.5*F(1);
204 h = sum(H)+F;
205 s = sum(S)+F;
206 % The CDF of cf B
207 W = (2./(b-a))*Chi(a,b,N,a,b);
208 CDF = real(cf_Z.*exp(-i*a*omega)).*W;
209 CDF(1)=0.5*CDF(1);
210 % initial portfolio given by C0 == 0
211 syms a

```

```

212 b = H0-a*S0;
213 x = b+a*S0;
214 c = x - H0;
215 system = c == 0;
216 var = a;
217 alpha_cos = double(solve(system, var));
218 beta_cos = H0-alpha_cos*S0;
219 % Wealth proces
220 X_cos = exp(-r*T).*H+exp(r*T).*beta_cos.*F+alpha_cos.*S;
221 pi_s_cos = alpha_cos/sum(X_cos);
222 pi_p_cos = 1-pi_s_cos;
223 % determining quantile index
224 n = 1;
225 while n < length(H)+1
226     J(n) = sum(CDF(1:n));
227     if J(n) >= eps && J(n) < 1
228         break;
229     else
230         n = n+1;
231     end
232 end
233 idx = n-1;
234 C = exp(-r*T).*sum(X_cos(idx:end)).*F-exp(-r*T).*sum(H(idx:end)).*F;
235 [row, col] = find(C>=0);
236 A_cos = sum(H(col))+F;
237 dN = length(A_cos(idx:end));
238 % alpha at C>=0 at idx
239 syms a
240 h1 = exp(-r*T)*A_cos(idx);
241 b = H0-a*S0;
242 x = h1+exp(r*T)*b+a*sum(S);
243 c = exp(-r*T)*x-H0;
244 system = c == 0;
245 var = a;
246 % optimal portfolio for VaR
247 alpha_cos = double(solve(system, var));
248 beta_cos = H0-alpha_cos*S0;
249 X_var_cos = H0+exp(r*T)*beta_cos+alpha_cos.*sum(S);
250 pi_s_var_cos = alpha_cos/X_var_cos;
251 pi_p_var_cos = 1-pi_s_var_cos;
252 % Initial investments
253 H0_var_cos = beta_cos+alpha_cos*S0;

```

```

254 % hedging error
255 RL_var_cos = exp(-r*T)*X_var_cos-H0;
256 % alpha at C>=0 from idx
257 syms a
258 h2 = exp(-r*T)*(1/(dN*eps))*sum(A_cos(idx+1:end))+h1;
259 b = H0-a*S0;
260 x = h2+exp(r*T)*b+a*sum(S);
261 c = exp(-r*T)*x - H0;
262 system = c == 0;
263 var = a;
264 alpha_a_cos = double(solve(system, var));
265 % optimal portfolio for ES
266 alpha_es_cos = alpha_a_cos;
267 beta_es_cos = H0-alpha_es_cos*S0;
268 X_es_cos = H0+exp(r*T)*beta_es_cos+alpha_es_cos*sum(S);
269 pi_s_es_cos = alpha_es_cos/X_es_cos;
270 pi_p_es_cos = 1-pi_s_es_cos;
271 % Initial investments
272 H0_es_cos = beta_es_cos+alpha_es_cos*S0;
273 % hedging error
274 RL_es_cos = exp(-r*T)*(X_es_cos-H);
275 toc
276
277 fprintf('Done\n')
278 save('NIGVarHedge.mat', 'RL_var', 'RL_es', 'RL_var_cos', 'RL_es_cos', ...
279     'alpha_a', 'alpha', 'beta', 'alpha_es', 'beta_es', 'alpha_a_cos', 'alpha_cos', '
280     beta_cos', 'alpha_es_cos', 'beta_es_cos', ...
281     'a', 'b', 'pi_p_es_cos', 'pi_p_var_cos', 'pi_s_es_cos', 'pi_s_var_cos', 'pi_p_es'
282     , ...
283     'pi_p_var', 'pi_s_es', 'pi_s_var', 'H0_var', 'H0_es', 'H0_var_cos', 'H0_es_cos')
284 fprintf('Variables saved!\n')

```

VG Quantile Hedging

```

1 clear;
2 clc;
3 format long
4 %% Initial Parameters
5 T=1;
6 dt=1/250;
7 M=T/dt;
8 S0=57;

```

```

9  r=0.0367;
10 N = 100000;
11 K = 52;
12 finalStrikePrice = K;
13 eps = 0.99;
14
15 %% Hedging by MC
16 tic
17 k = 1; l = 1;
18 fprintf('Total number of simulations: %i\n',N)
19 % Parameters from calibration
20 muY = 0; % drift parameter
21 sigmaY = 0.02; % volatitlity
22 kappa = 0.1;
23 c = 0;
24 % Gamma Subordinated Process
25 % Generate dependent Gamama random time process
26 [R,G0] = GammaProc(kappa , c , dt , T,N) ;
27 % Generate VG process
28 [S] = VGProc(R, S0 ,muY, sigmaY ,T, dt) ;
29 V = max(S-K,0) ;
30 ST = S(end ,:);
31 VT = V(end ,:);
32 V0 = exp(-r*T)*mean(mean(V));
33 fprintf('Done Gamma Sub\n')
34
35 %% VaR Hedging with MC by replication
36 % hedging portfolio
37 % initial portfolio given by  $C_0 = 0$ 
38 syms a
39 b = V0-a*S0;
40 x = b+a*S0;
41 c = x-V0;
42 system = c == 0;
43 var = a;
44 alpha = double(solve(system , var));
45 beta = V0-alpha*S0;
46 % wealth process
47 X = V0+exp(r*T).*beta+alpha.*ST;
48 % determining quantile index
49 C = exp(-r*T)*X-exp(-r*T)*VT;
50 [row , col] = find(C>=0);

```

```

51 A = VT(col);
52 [J,I] = sort(C(col), 'ascend');
53 A = A(I);
54 idx = floor(length(A)*(eps))+1;
55 dN = length(A(1:idx));
56 % alpha at C == 0
57 syms a
58 V1 = exp(-r*T)*A(idx);
59 b = V0-a*S0;
60 x = V1+exp(r*T)*b+a*mean(ST);
61 c = exp(-r*T)*x - V0;
62 system = c == 0;
63 var = a;
64 % optimal portfolio for VaR
65 alpha = double(solve(system, var));
66 beta = V0-alpha*S0;
67 X_var = V0+exp(r*T)*beta+alpha*mean(ST);
68 pi_s_var = alpha/X_var;
69 pi_p_var = 1-pi_s_var;
70 % Initial investments
71 H0_var = beta+alpha*S0;
72 % hedging error
73 RL_var = exp(-r*T)*(X_var-VT);
74
75 %% ES Hedging with MC by replication
76 % optimal portfolio for ES
77 % alpha at C>=0 from 1:idx-1
78 syms a
79 V2 = exp(-r*T)*(1/(dN*eps))*sum(A(1:idx-1))+V1;
80 b = V0-a*S0;
81 x = V2+exp(r*T)*b+a*mean(ST);
82 c = exp(-r*T)*x - V0;
83 system = c == 0;
84 var = a;
85 alpha_a = double(solve(system, var));
86 % optimal portfolio
87 alpha_es = alpha_a;
88 beta_es = V0-alpha_es*S0;
89 X_es = V0+exp(r*T)*beta_es+alpha_es*mean(ST);
90 pi_s_es = alpha_es/X_es;
91 pi_p_es = 1-pi_s_es;
92 % Initial investments

```

```

93 H0_es = beta_es+alpha_es*S0;
94 % hedging error
95 RL_es = exp(-r*T)*(X_es-VT);
96 toc
97 %% Hedging with COS Method
98 tic
99 fprintf('Start T=%i\n',T)
100 %Note: According to our error analysis , there is term N(error(q)), so when
101 %N goes up, N_I MUST BE INCREASED AT THE SAME TIME!
102 %Default values
103 N=256;
104 N_I=400;
105
106 % Initializing
107 tic;
108 i=complex(0,1);
109 q=0;
110
111 % Parameters from calibration
112 muX = 0; % drift parameter
113 sigmaX = 0.02; % volatitlity
114 kappa = 0.1;
115 c = 0;
116
117 a0 = 1/kappa; b0 = 1/kappa;
118 %% Cumulant and trunction range for VG
119 c1 = dt*muX;
120 c2 = dt*(sigmaX^2+muX*kappa);
121 %
122 %% integration truncation
123
124 % L=10 converges well to the reference value. When increasing L, N, N_I
125 % should also be increased. And when L is very large , should use put-call
126 % parity.
127 L=10;
128 a=c1-L*sqrt(M*c2);
129 b=log(M)+M*c1+L*sqrt(M*c2);
130
131
132 k=0:N-1;
133 omega=k'*pi/(b-a);
134 v = i*muX*omega-0.5*omega.^2*sigmaX^2;

```

```

135 cf_Z=(1-i.*(-i*v)./b0).^ (a0*dt);
136 % cf_Z=exp(i*omega*(r-q+w)*dt-0.5*(omega.^2)*(sigma^2)*dt+dt*delta*(sqrt(alpha^2-
      beta^2)-sqrt(alpha^2-(beta+i*omega).^2)));
137 Int_Aasian=zeros(N, N);
138
139 % Only for Clenshaw-Curtis quadrature
140 Ni=0:2:N_I;
141 Ni=Ni';
142 d=2./(1-Ni.^2);
143 d(1)=1;
144 d(end)=0.5*d(end);
145 n=0:1:floor(N_I/2);
146 D=n'*n;
147 D=2/(N_I)*cos(D*2*pi/N_I);
148 D(:,1)=0.5*D(:,1);
149 D(:,end)=0.5*D(:,end);
150 w1=D'*d;
151
152 for k1=1:N-1
153     for k2=1:N-1
154         Int_Aasian(k1,k2) = asianmat(omega(k1), omega(k2), N_I, a, b, a, w1);
155     end
156 end
157
158 %% Only for Beta function
159 % for k1=1:N-1
160 %     for k2=1:N-1
161 %         Int_Aasian(k1,k2) = asianmat(k1, k2, a, b);
162 %     end
163 % end
164
165 cf_B=cf_Z;
166
167 for j=2:M
168     B=real(cf_B.*exp(-i*a*omega));
169     B(1)=0.5*B(1);
170     Phi_B=(2/(b-a))*Int_Aasian*B;
171     cf_B=cf_Z.*Phi_B;
172 end
173
174
175 % ep stand for 'exercise point'

```

```

176 ep=log(K*(M+1)/S0-1);
177
178 if ep>b
179     ep=b;
180 elseif ep<a
181     ep=a;
182 end
183 %% VaR and ES Hedging with COS Method
184
185 U=(2./(b-a))*((S0./(M+1))*Chi(ep,b,N,a,b)-(K-(S0/(M+1)))*Psi(ep,b,N,a,b));
186 Re=real(cf_B.*exp(-i*a*omega));
187 % Asian option vlaue
188 H=U.*Re';
189 H(1)=0.5*H(1);
190 H0 = exp(-r*T)*sum(H);
191 % Price process
192 Re=real(cf_Z.*exp(-i*a*omega));
193 % E[S0exp(Y)] = V.*Re'
194 V = (2./(b-a))*S0*Chi(a,b,N,a,b);
195 % stock price at t=T
196 S=V.*Re';
197 S(1)=0.5*S(1);
198 % Fourier-Cosine of 1
199 W = (2./(b-a))*Psi(a,b,N,a,b);
200 F = real(exp(-i*a*omega))'.*W;
201 F(1)=0.5*F(1);
202 h = sum(H)+F;
203 s = sum(S)+F;
204 % The CDF of cf B
205 W = (2./(b-a))*Chi(a,b,N,a,b);
206 CDF = real(cf_Z.*exp(-i*a*omega))'.*W;
207 CDF(1)=0.5*CDF(1);
208 % initial portfolio given by C0 == 0
209 syms a
210 b = H0-a*S0;
211 x = b+a*S0;
212 c = x - H0;
213 system = c == 0;
214 var = a;
215 alpha_cos = double(solve(system,var));
216 beta_cos = H0-alpha_cos*S0;
217 % Wealth proces

```

```

218 X_cos = exp(-r*T).*H+exp(r*T).*beta_cos.*F+alpha_cos.*S;
219 pi_s_cos = alpha_cos/sum(X_cos);
220 pi_p_cos = 1-pi_s_cos;
221 % determining quantile index
222 n = 1;
223 while n < length(H)+1
224     J(n) = sum(CDF(1:n));
225     if J(n) >= eps && J(n) < 1
226         break;
227     else
228         n = n+1;
229     end
230 end
231 idx = n-1;
232 C = exp(-r*T).*sum(X_cos(idx:end)).*F-exp(-r*T).*sum(H(idx:end)).*F;
233 [row, col] = find(C>=0);
234 A_cos = sum(H(col))+F;
235 dN = length(A_cos(idx:end));
236 % alpha at C>=0 at idx
237 syms a
238 h1 = exp(-r*T)*A_cos(idx);
239 b = H0-a*S0;
240 x = h1+exp(r*T)*b+a*sum(S);
241 c = exp(-r*T)*x-H0;
242 system = c == 0;
243 var = a;
244 % optimal portfolio for VaR
245 alpha_cos = double(solve(system, var));
246 beta_cos = H0-alpha_cos*S0;
247 X_var_cos = H0+exp(r*T)*beta_cos+alpha_cos.*sum(S);
248 pi_s_var_cos = alpha_cos/X_var_cos;
249 pi_p_var_cos = 1-pi_s_var_cos;
250 % Initial investments
251 H0_var_cos = beta_cos+alpha_cos*S0;
252 % hedging error
253 RL_var_cos = exp(-r*T)*X_var_cos-H0;
254 % alpha at C>=0 from idx
255 syms a
256 h2 = exp(-r*T)*(1/(dN*eps)).*sum(A_cos(idx+1:end))+h1;
257 b = H0-a*S0;
258 x = h2+exp(r*T)*b+a*sum(S);
259 c = exp(-r*T)*x - H0;

```

```

260 system = c == 0;
261 var = a;
262 alpha_a_cos = double(solve(system, var));
263 % optimal portfolio for ES
264 alpha_es_cos = alpha_a_cos;
265 beta_es_cos = H0-alpha_es_cos*S0;
266 X_es_cos = H0+exp(r*T)*beta_es_cos+alpha_es_cos*sum(S);
267 pi_s_es_cos = alpha_es_cos/X_es_cos;
268 pi_p_es_cos = 1-pi_s_es_cos;
269 % Initial investments
270 H0_es_cos = beta_es_cos+alpha_es_cos*S0;
271 % hedging error
272 RL_es_cos = exp(-r*T)*(X_es_cos-H0);
273 toc
274
275 fprintf('Done\n')
276 save('VGVarHedge.mat', 'RL_var', 'RL_es', 'RL_var_cos', 'RL_es_cos', ...
277     'alpha_a', 'alpha', 'beta', 'alpha_es', 'beta_es', 'alpha_a_cos', 'alpha_cos', '
278     beta_cos', 'alpha_es_cos', 'beta_es_cos', ...
279     'a', 'b', 'pi_p_es_cos', 'pi_p_var_cos', 'pi_s_es_cos', 'pi_s_var_cos', 'pi_p_es'
280     , ...
281     'pi_p_var', 'pi_s_es', 'pi_s_var', 'H0_var', 'H0_es', 'H0_var_cos', 'H0_es_cos')
282 fprintf('Variables saved!\n')

```

Calibration

Implied Volatility

```

1 %% MC Implied Volatility
2 clear;
3 clc;
4 format long
5 tic
6 %%
7 % Initialize Variables
8 load('AsianOptionMarketData.mat');
9 load('AsianOptionMarketDataPut.mat');
10 load('UnivarY1.mat');
11 M = 12; % sampling frequency
12 T = 1;
13 dT = T/M;
14 N = 1000000; % number of simulations

```

```

15 rate = 0.02; % risk free rate of return in %*100
16 S0Call = 68.58; % initial commodity price
17 S0Put = S0Call;
18 % KSpread = 52; % Strike for Spread
19 fprintf('Total number of simulations: %i\n',N)
20 %%
21 % Asian option payoff for each strike price
22 s = size(WTIAsnOpt);
23 sigma_impl_initialCall = zeros(s(1),1);
24 sigma_impl_initialPut = zeros(s(1),1);
25 k=1;
26 tic
27 fprintf('Getting initial points\n')
28 while k < s(1)+1
29     % Parameters from calibration
30     muX = y(2,k);
31     sigmaX = y(3,k);
32     kappa = y(1,k);
33     c = 0;
34     muY = dT.*muX;
35     sigmaY = sqrt(dT.*(kappa.*muX.^2+sigmaX.^2));
36     % initial point
37     [R_C] = IGProc(kappa,c,dT,T,N);
38     [S_C] = NIGProc(R_C,S0Call,muY,sigmaY);
39     sigma_impl_initialCall(k) = sqrt(2/M*(exp(-rate*dT)*abs(mean(mean(S_C)))/K(k)
40         ));
41     [R_P] = IGProc(kappa,c,dT,T,N);
42     [S_P] = NIGProc(R_P,S0Put,muY,sigmaY);
43     sigma_impl_initialPut(k) = sqrt(2/M*(exp(-rate*dT)*abs(K(k)/mean(mean(S_P))))
44         );
45     k = k+1
46 end
47 save('initialPoint.mat','sigma_impl_initialCall','sigma_impl_initialPut')
48 fprintf('Variables saved!\n')
49 toc
50 %%
51 load('initialPoint.mat')
52 load('UnivarY1.mat')
53 % Asian option observed payoff
54 Payoff_WTI_mkt = WTIAsnOpt+WTIAsnOptPut;
55 % Implied Volatility WTI
56 fprintf('Starting implied vol calc\n')

```

```

55 tic
56 k=1;
57 sigma_impl_NIG = zeros(1,101);
58 while k < length(K)+1
59     % Parameters from calibration
60     muX = y(2,k);
61     sigmaX = y(3,k);
62     kappa = y(1,k);
63     c = 0;
64     muY = dT.*muX;
65     sigmaY = sqrt(dT.*(kappa.*muX.^2+sigmaX.^2));
66     fprintf('Elements Remaining: %d\n',length(K)-k);
67     if S0Call > K(k)
68         sigma_impl_NIG(k) = sigma_impl_initialCall(k);
69         error = 10; n = 2;
70         while error > 0.5*(10^(-4))
71             sigma_impl_NIG_old(k) = sigma_impl_NIG(k);
72             [R1,L0] = IGProc(kappa,c,dT,T,N);
73             [S1,dY1,Y1,W1] = NIGProc(R1,S0Call,muY,sigma_impl_NIG_old(k));
74             [Payoff1(k)] = AsianOption(S1,K(k),rate,dT);
75             vega_NIG = mean(sum(mean(S1)>K(k)).*mean(S1.*W1))
76             sigma_impl_NIG(k) = sigma_impl_NIG_old(k) - ((Payoff1(k) ...
77                 -Payoff_WTI_mkt(k,end))/vega_NIG)
78             error = abs(sigma_impl_NIG(k)-sigma_impl_NIG_old(k))
79             n = n+1;
80             if n > 1000000
81                 fprintf('Too many iterations\n');
82                 break;
83             end
84             k = k+1;
85         end
86     elseif S0Call <= K(k)
87         sigma_impl_NIG(k) = sigma_impl_initialPut(k);
88         error = 10; n = 1;
89         while error > 0.5*(10^(-4))
90             sigma_impl_NIG_old(k) = sigma_impl_NIG(k);
91             [R1,L0] = IGProc(kappa,c,dT,T,N);
92             [S1,dY1,Y1,W1] = NIGProc(R1,S0Put,muY,sigma_impl_NIG_old(k));
93             [Payoff1(k)] = AsianOption(S1,K(k),rate,dT);
94             vega_NIG = mean(sum(mean(S1)<K(k)).*mean(S1.*W1))
95             sigma_impl_NIG(k) = sigma_impl_NIG_old(k) - ((Payoff1(k) ...
96                 -Payoff_WTI_mkt(k,end))/vega_NIG)

```

```

97         error = abs(sigma_impl_NIG(k)-sigma_impl_NIG_old(k))
98         n = n+1;
99         if n > 1000000
100             fprintf('Too many iterations\n');
101             break;
102         end
103         k = k+1;
104     end
105 end
106 end
107 %% Correction for sigma < 0 using weighted interpolation
108 fprintf('Applying correction\n')
109 if min(size(find(sigma_impl_NIG < 0))) > 0
110     a = find(sigma_impl_NIG < 0);
111     a = sort(a);
112     a = [a(1)-1,a,a(end)+1];
113     j=2;
114     while j < length(a)
115         sigma_impl_NIG(a(j)) = (abs(sigma_impl_NIG(a(j))-sigma_impl_NIG(a(1)))/...
116             (sigma_impl_NIG(a(1))-sigma_impl_NIG(a(end))))*sigma_impl_NIG(a(1))...
117             +abs(sigma_impl_NIG(a(j))-sigma_impl_NIG(a(end)))/(sigma_impl_NIG(a(1))...
118             -sigma_impl_NIG(a(end))))*sigma_impl_NIG(a(end))/length(a-2);
119         j=j+1;
120     end
121 end
122 save('implVol.mat','sigma_impl_NIG')
123 fprintf('Variables saved!\n')
124 %%
125 load('implVol.mat')
126 figure
127 plot(K,sigma_impl_NIG,'-o')
128 ylim([0.3,0.9])
129 xlabel('Strike Price ($)')
130 ylabel('Implied Volatility')
131 title('Sensitivity of Implied Volatility to Strike')
132 toc

```

Parameter Calibration

```

1 %% MC Implied Volatility
2 clear;
3 clc;

```

```
4 format long
5 tic
6 %%
7 dT = 1/12;
8 N = 100000; % number of simulations
9 rate = 0.02; % risk free rate of return in %*100
10 S0 = 68.58; % initial commodity price
11 S0_Call2 = 74.88;
12 % KSpread = 52; % Strike for Spread
13 fprintf('Total number of simulations: %i\n',N)
14 %%
15 var = 2; % 1 is univar 2 is bivar
16 put = 0; % if calibrating puts set to 1
17 % Initialize Variables
18 if var == 2
19     load('AsianOptionMarketData.mat');
20     K_Asn = K;
21     Payoff_WTI_mkt = WTIAsopt;
22     load('EuroOptionMarketData.mat');
23     K_Eur= K;
24     Payoff_BRNT_euro_mkt = BRNTEuroOpt;
25 else
26     if put == 1
27         load('AsianOptionMarketDataPut.mat');
28         K_Asn = KPut;
29         K = KPut;
30         Payoff_WTI_mkt_Put = WTIAsoptPut;
31     else
32         load('AsianOptionMarketData.mat');
33         K_Asn = K;
34         Payoff_WTI_mkt = WTIAsopt;
35     end
36 end
37 % Parameters for calibration
38 if var == 1
39     muX_WTI_Call = 0.3;
40     kappa0 = 0.25;
41     sigmaX_WTI_Call = 0;
42 else
43     muX_WTI_Call = 0.3;
44     muX_WTI_Call2 = 0;
45     kappa0 = 0.25;
```

```

46     kappa1 = 0.25;
47     kappa2 = 0.25;
48     c1 = 0.2;
49     c2 = 1;
50     sigmaX_WTI_Call = 0;
51     sigmaX_WTI_Call2 = 0;
52 end
53 %%
54 if var == 2
55     k = 1
56     while k < length(K_Asn)+1
57         fprintf('Starting bivariate calibration for y\n')
58         options = optimoptions('patternsearch','Display','iter','MeshTolerance',1
59             e-10,'FunctionTolerance',1e-6,'StepTolerance',1e-6);
60         y0 = [kappa0;kappa1;c1;muX_WTI_Call;sigmaX_WTI_Call];
61         lb = [0;0;0;-inf;0]; ub = [inf;inf;inf;inf;inf];
62         fun_WTI = @(y) Payoff_WTI_NIG(y(1),y(2),y(3),K_Asn(k),y(4),y(5),N,rate,S0,
63             Payoff_WTI_mkt(k,end));
64         [y(:,k),fval_y,exitflag,output_y] = patternsearch(fun_WTI,y0,[],[],[],[],
65             lb,ub,[],options)
66         muY = dT*y(4,k)+y(3,k)*y(4,k)*dT;
67         sigmaY = sqrt(dT*((y(1,k)+(y(3,k)^2)*y(2,k))*y(4,k)^2+...
68             (1+y(3,k))*y(5,k)^2));
69         R1 = IGProc([y(1,k),y(2,k)],y(3,k),dT,1,N,1);
70         S1 = NIGProc(R1,S0,muY,sigmaY);
71         Payoff_WTI_cal(k) = mean(AsianOption(S1,K_Asn(k),rate,dT,Payoff_WTI_mkt(k
72             ,end)));
73         k = k+1
74     end
75     save('BivarY1.mat','y0','y','Payoff_WTI_cal')
76     fprintf('y Variables saved!\n')
77     %% x
78     fprintf('Starting calibration for x\n')
79     load('BivarY1.mat')
80     load('EuroOptionMarketData.mat');
81     K_Eur= K;
82     Payoff_BRNT_euro_mkt = BRNTEuroOpt;
83     l = 1
84     [a,b] = intersect(K_Asn,K_Eur);
85     y = y(:,b);
86     while l < length(K_Eur)+1
87         fprintf('Starting the search for x\n')

```

```

84     options = optimoptions('patternsearch','Display','iter','MeshTolerance',1
      e-10,'FunctionTolerance',1e-6,'StepTolerance',1e-6);
85     x0 = [kappa2;c2;muX_WTI_Call2;sigmaX_WTI_Call2];
86     lb = [0;0;-inf;0]; ub = [inf;inf;inf;inf];
87     fun_BRNT = @(x) Payoff_NIG_Call2(x(1),x(1),x(2),K_Eur(1),x(3),x(4),N,rate,
      S0_Call2,Payoff_BRNT_euro_mkt(1));
88     [x(:,l),fval_x,exitflag,output_x] = patternsearch(fun_BRNT,x0
     ,[],[],[],[],lb,ub,[],options)
89     [R1,L0,L1] = IGProc([y(1,l),x(1,l)],x(2,l),1/12,1,N,1);
90     muY = dT*x(3,l)+x(2,l)*x(3,l)*dT;
91     sigmaY = sqrt(dT*((y(1,l)+(x(2,l)^2)*x(1,l))*x(3,l)^2+...
92     (1+x(2,l))*x(4,l)^2));
93     S1 = NIGProc(x(2,l)*L1,S0_Call2,muY,sigmaY);
94     Payoff_BRNT_euro_cal(1) = mean(EuroOption(S1,K_Eur(1),rate,dT,
      Payoff_BRNT_euro_mkt(1,end)));
95     l = l+1
96     end
97     save('BivarX1.mat','x0','x','Payoff_BRNT_euro_cal')
98     fprintf('x Variables saved!\n')
99 elseif var == 1
100     if put == 1
101         k = 1;
102         while k < length(K)+1
103             fprintf('Starting univariate put calibration for y\n')
104             options = optimoptions('patternsearch','Display','iter','
      MeshTolerance',1e-10,'FunctionTolerance',1e-6,'StepTolerance',1e
      -6);
105             y0 = [kappa0;muX_WTI_Call;sigmaX_WTI_Call];
106             lb = [0;-inf;0];
107             ub = [inf;inf;inf];
108             fun_WTI = @(y) Payoff_WTI_NIG_Put(y(1),0,0,K_Asn(k),y(2),y(3),N,rate,
      S0,Payoff_WTI_mkt_Put(k,end));
109             [y(:,k),fval_y,exitflag,output_y] = patternsearch(fun_WTI,y0
     ,[],[],[],[],lb,ub,[],options)
110             R1 = IGProc(y(1,k),0,1/12,1,N,0);
111             muY = dT*y(2,k);
112             sigmaY = sqrt(dT*(y(1,k)*y(2,k)^2+y(3,k)^2));
113             S1 = NIGProc(R1,S0,muY,sigmaY);
114             Payoff_WTI_cal_Put(k) = mean(AsianOptionPut(S1,K_Asn(k),rate,dT,
      Payoff_WTI_mkt_Put(k,end)));
115             k = k+1
116         end

```

```

117     save('UnivarY1Put.mat','y0','y','Payoff_WTI_cal_Put')
118     fprintf('Variables saved!\n')
119     else
120     k = 1;
121     while k < length(K)+1
122         fprintf('Starting univariate calibration for y\n')
123         options = optimoptions('patternsearch','Display','iter','
            MeshTolerance',1e-10,'FunctionTolerance',1e-6,'StepTolerance',1e
            -6);
124         y0 = [kappa0;muX_WTI_Call;sigmaX_WTI_Call];
125         lb = [0;-inf;0];
126         ub = [inf;inf;inf];
127         fun_WTI = @(y)Payoff_WTI_NIG(y(1),0,0,K_Asn(k),y(2),y(3),N,rate,S0,
            Payoff_WTI_mkt(k,end));
128         [y(:,k),fval_y,exitflag,output_y] = patternsearch(fun_WTI,y0
           ,[],[],[],[],lb,ub,[],options)
129         R1 = IGProc(y(1,k),0,1/12,1,N,0);
130         muY = dT*y(2,k);
131         sigmaY = sqrt(dT*(y(1,k)*y(2,k)^2+y(3,k)^2));
132         S1 = NIGProc(R1,S0,muY,sigmaY);
133         Payoff_WTI_cal(k) = mean(AsianOption(S1,K_Asn(k),rate,dT,
            Payoff_WTI_mkt(k,end)));
134         k = k+1
135     end
136     save('UnivarY1.mat','y0','y','Payoff_WTI_cal')
137     fprintf('Variables saved!\n')
138 end
139 end
140
141 toc
142
143 %% Plots
144
145 load('UnivarY1.mat')
146 load('AsianOptionMarketData.mat')
147 Payoff_WTI_mkt = WTIASnOpt;
148 figure
149 subplot(4,1,1)
150 hold on
151 plot(Payoff_WTI_cal,'-+','LineWidth',0.5)
152 plot(Payoff_WTI_mkt(:,end),'-o','LineWidth',0.5)
153 xlabel('Strike Prices ($)')

```

```
154 ylabel('Price ($)')
155 title('(a) Calibration of Univariate Model Call')
156 % legend('Arithmetic Asian Call Calibrated Price','Asian Call Market Price')
157
158 load('AsianOptionMarketDataPut.mat');
159 load('UnivarY1Put.mat')
160 Payoff_WTI_mkt_Put = WTIASnOptPut;
161 subplot(4,1,2)
162 hold on
163 plot(Payoff_WTI_cal_Put, '-+', 'LineWidth', 0.5)
164 plot(Payoff_WTI_mkt_Put(:,end), '-o', 'LineWidth', 0.5)
165 xlabel('Strike Prices ($)')
166 ylabel('Price ($)')
167 title('(b) Calibration of Univariate Model Put')
168 % legend('Arithmetic Asian Put Calibrated Price','Asian Put Market Price')
169
170 load('BivarY1.mat')
171 load('AsianOptionMarketData.mat')
172 Payoff_WTI_mkt = WTIASnOpt;
173 subplot(4,1,3)
174 hold on
175 plot(Payoff_WTI_cal, '-+', 'LineWidth', 0.5)
176 plot(Payoff_WTI_mkt(:,end), '-o', 'LineWidth', 0.5)
177 xlabel('Strike Prices ($)')
178 ylabel('Price ($)')
179 title('(c) Calibration of Asian Payoff')
180 legend('Calibrated Price','Market Price')
181
182 load('BivarX1.mat')
183 load('EuroOptionMarketData.mat');
184 Payoff_BRNT_euro_mkt = BRNTEuroOpt;
185 subplot(4,1,4)
186 hold on
187 plot(Payoff_BRNT_euro_cal, '-+', 'LineWidth', 0.5)
188 plot(Payoff_BRNT_euro_mkt, '-o', 'LineWidth', 0.5)
189 xlabel('Strike Prices ($)')
190 ylabel('Price ($)')
191 title('(d) Conditional Calibration of European Payoff')
192 % legend('European Call Calibrated Price','European Call Market Price')
```

Portfolio Choice

Simulation

```

1 clear all
2 clc
3 %% loads the simplifeid symbolic portfolios from Check2.m
4 % load('portfolioEZ.mat');
5 %%
6 % time counter start
7 tic
8 %% Parameters
9 % Initial time
10 t0=0;
11 % Terminal time
12 tT=20;
13 % Initial variables
14 qr=0;
15 sigma_r=0.03;sigma_l=0.1467;sigma_b=0.2620;sigma_s = 0.2;
16 rbar=0.02; lbar=-2.1493; bbar=0;
17 k_r=0.5;k_l=0.2935;k_b=4.0942;
18 P=3.7816;P_l=1.6315;P_b=13.4035;
19 rho_sp=0; rho_sl=-0.2186; rho_sb=-0.2164;
20 rho_Pb=0.3;rho_Lp=0.3;rho_Lb=-0.4913;
21
22 %% mean reverting sigma_s(Long Run)
23 % epsilon = 0.025;
24 % sigma_s=epsilon+(sigma_l^2/(2*k_l));
25
26 % filtered correlation parameter
27 rho_l=sqrt(1-rho_sl^2);
28 rho_lp=(rho_Lp-rho_sl*rho_sp)/rho_l;
29 rho_p=sqrt(1-rho_sp^2-rho_lp^2);
30 rho_lb=(rho_Lb-rho_sl*rho_sb)/rho_l;
31 rho_pb=(rho_Pb-rho_sp*rho_sb-rho_lp*rho_lb)/rho_p;
32 rho_b=sqrt(1-rho_sb^2-rho_lb^2-rho_pb^2);
33
34 % solving for m
35 syms m
36 eqns = -2*k_b*m+sigma_b^2-(sigma_b*rho_sb+(m*P_b))^2-...
37         (sigma_b*rho_lb -((m*P_b*rho_sl)/rho_l))^2-...
38         (sigma_b*rho_pb+(m*P_b*(rho_sl*rho_lp-rho_l*rho_sp))/(rho_l*rho_p))^2;
39 vars = m;

```

```

40 [m]=solve (eqns==0,vars);
41 m = vpa(m);
42 m = double(m(2));
43 a1=sigma_b*rho_sb+(m*Pb);
44 a2=sigma_b*rho_lb -((m*Pb*rho_sl)/rho_l);
45 a3=sigma_b*rho_pb+(m*Pb*(rho_sl*rho_lp-rho_l*rho_sp))/(rho_l*rho_p);
46
47 % testing parameters , parameters for behavior
48 gamma=4;psi=1/0.75;vphi=0.0153;% vphi is time preference parameter>0
49 %%
50 % long run average of values
51 r=rbar; l=lbar; b=bbar;
52 sigma_p = (sigma_r/k_r)*(1-exp(-k_r*(tT-t0))); q = qr*sigma_p;
53 % variables
54 vrho = [0;1;0];
55 I3 = diag([1,1,1]);
56 I2 = diag([1,1]);
57 % state variable means
58 kbar = [k_l*lbar;k_r*rbar;k_b*bbar];
59 k =[k_l,0,0;0,k_r,0;0,0,k_b];
60 % state variable volatility matrix
61 Sy = [rho_sl*sigma_l,rho_l*sigma_l,0;...
62       -rho_sp*sigma_r,-rho_lp*sigma_r,-rho_p*sigma_r;...
63       a1,a2,a3];
64 % covariance of bonds and stock
65 S = [sigma_s,0,; sigma_p*rho_sp,sigma_p*sqrt(1-rho_sp^2)];
66 rho = [1,0;
67        0,rho_lp/rho_l;
68        0,rho_p/rho_l];
69 Sx = S*rho.';
70 % mean of bonds and stock minus interest
71 mu = [sigma_s*(P+Pl*l+Pb*b);q];
72 % T is the ambiguity parameter
73 tau = 0:1:10;
74
75 %% Simulation
76 i = 1
77 check = 1;
78 while i<=length(tau)
79     syms a_1 b1_1 b2_1 b3_1 q11_1 q12_1 q13_1 q21_1 q22_1 q23_1 q31_1 q32_1 q33_1
80     A = a_1;
81     B = [b1_1;b2_1;b3_1];

```

```

82 Q = [q11_1 , q12_1 , q13_1 ; q21_1 , q22_1 , q23_1 ; q31_1 , q32_1 , q33_1 ] ;
83 t1_1 = tau(i) ; t2_2 = tau(i) ; t3_3 = tau(i) ;
84 T = [t1_1 , 0 , 0 ; 0 , t2_2 , 0 ; 0 , 0 , t3_3 ]
85
86 % portfolio constants
87 phi_1=S*(rho.'*T*rho+gamma*I2)*S.' ;
88 phi_2=Sx*(T-I3+gamma*I3)*Sy.' ;
89 phi_3 = Sy*((1-gamma)/(1-psi))*I3+I3-(1/(1-psi))*T)*Sy.' ;
90 phi_4 = (1/(1-psi))*phi_2.'*inv(phi_1)*phi_2+phi_3 ;
91
92 % phi_1 components
93 vphi0 = t1_1*rho_l^2*rho_sp^2 + gamma*rho_l^2 - t3_3*rho_p^2*rho_sp^2+ ...
94         t3_3*rho_p^2 - t2_2*rho_lp^2*rho_sp^2 + t2_2*rho_lp^2 ;
95 vphi1 = - gamma*rho_l^2*rho_sp^2 + gamma*rho_l^2 - t3_3*rho_p^2*rho_sp^2+...
96         t3_3*rho_p^2 - t2_2*rho_lp^2*rho_sp^2 + t2_2*rho_lp^2 ;
97
98 % mu.'*inv(phi_1)*mu
99 J0 = (P^2*vphi0-2*(gamma+t1_1)*(rho_sp*rho_l^2)*qr*P+...
100      qr^2*(rho_l^2)*(gamma+t1_1))/((gamma+t1_1)*vphi1) ;
101 J1 = -(2*((gamma+t1_1)*qr*(rho_sp*(rho_l^2))-P*vphi0))/((gamma+t1_1)*vphi1)*[
102      P1;0;Pb] ;
103 J2 = (2*vphi0)/((gamma+t1_1)*vphi1)*[P1^2 0 P1*Pb;0 0 0; P1*Pb 0 Pb^2] ;
104
105 % phi_2.'*inv(phi_1)*mu
106 H = phi_2.'*inv(phi_1) ;
107 F = sigma_s*H(:,1) ; R = qr.*sigma_p.*H(:,2) ;
108
109 g0 = P.*F+R ;
110 g1 = [P1*F(1) 0 Pb*F(1) ;
111       P1*F(2) 0 Pb*F(2) ;
112       P1*F(3) 0 Pb*F(3) ] ;
113
114 % Solving the System of equations
115 k1 = vphi ;
116 j=1 ;
117 while j == 1
118     syms a_1 b1_1 b2_1 b3_1 q11_1 q12_1 q13_1 q21_1 q22_1 q23_1 q31_1 q32_1
119           q33_1
120     A = a_1 ;
121     B = [b1_1 ; b2_1 ; b3_1] ;
122     Q = [q11_1 , q12_1 , q13_1 ; q21_1 , q22_1 , q23_1 ; q31_1 , q32_1 , q33_1 ] ;

```

```

122     k1_old = k1
123     k0 = k1*(1-log(k1));
124     const = -k0-psi*k1*log(vphi)+k1*A+vphi*psi+(g0-kbar) .* B...
125             -0.5*B. '(Sy*Sy.' - phi_4)*B-0.5*trace(Sy*Sy.'*Q) -0.5*(psi-1)*J0;
126
127     y = k1*B+(g1+k) .* B-Q*(Sy*Sy.' - phi_4)*B+Q*(g0-kbar) -0.5*(psi-1)*J1-(psi
128             -1)*vrho;
129
130
131     y2 = k1*Q+(g1+k) .* Q+Q. '(g1+k)-Q. '(Sy*Sy.' - phi_4)*Q-0.5*(psi-1)*J2;
132
133
134     if check ==1
135         q1 = -0.5*(psi-1)*J2;
136         q2 = k1+(g1+k) .';
137         q3 = (g1+k);
138         q4 = -(Sy*Sy.' - phi_4);
139         A1 = q1; A2 = q2+transpose(q3); A3 = q4;
140         quaddisc = A2.^2-4.*A3.*A1;
141         quadeqn = (-A2+quaddisc)*0.5*inv(A3)
142         check = check +1;
143     end
144
145     % solve Q
146     fprintf('Starting solution\n')
147     tic
148     x = [const;y;y2(:)];
149     %     varsx = [a_1,b_1_1,b_2_1,b_3_1,q11_1,q12_1,q13_1,q21_1,q22_1,q23_1,q31_1,
150     %     q32_1,q33_1];
151     %     initx = zeros(1,length(varsx));
152     %     Xsol = vpsolve(x==0,varsx,initx)
153     Xsol = vpsolve(y2==0)
154     if sum(size(Xsol.q11_1))>2
155         if psi < 1
156             idx = intersect(find(Xsol.q11_1>0),find(Xsol.q12_1==0))
157             if length(idx)>1
158                 idx = intersect(find(Xsol.q33_1<0),idx)
159             end
160         else
161             idx = intersect(find(Xsol.q11_1<0),find(Xsol.q12_1==0))
162             if length(idx)>1
163                 idx = intersect(find(Xsol.q33_1>0),idx)
164             end
165         end
166     end

```

```

162         end
163         q11_1=Xsol.q11_1(idx); q12_1=Xsol.q12_1(idx); q13_1=Xsol.q13_1(idx);
164         q21_1=Xsol.q21_1(idx); q22_1=Xsol.q22_1(idx); q23_1=Xsol.q23_1(idx);
165         q31_1=Xsol.q31_1(idx); q32_1=Xsol.q32_1(idx); q33_1=Xsol.q33_1(idx);
166     %         b1_1 = Xsol.b1_1(idx); b2_1= Xsol.b2_1(idx); b3_1 = Xsol.b3_1(idx);
167     %         a_1 = Xsol.a_1(idx);
168     else
169         q11_1=Xsol(1); q12_1=Xsol(2); q13_1=Xsol(3);
170         q21_1=Xsol(4); q22_1=Xsol.q22_1; q23_1=Xsol.q23_1;
171         q31_1=Xsol.q31_1; q32_1=Xsol.q32_1; q33_1=Xsol.q33_1;
172     %         b1_1 = Xsol.b1_1; b2_1= Xsol.b2_1; b3_1 = Xsol.b3_1;
173     %         a_1 = Xsol.a_1;
174     end
175     Xsol = vpasolve(subs(y)==0);
176     b1_1 = Xsol.b1_1; b2_1= Xsol.b2_1; b3_1 = Xsol.b3_1;
177     Xsol = vpasolve(subs(const)==0);
178     a_1 = Xsol;
179     % change to double
180     a=double(a_1); b1=double(b1_1); b2=double(b2_1); b3=double(b3_1);
181     c11=double(q11_1); c12=double(q12_1); c13=double(q13_1);
182     c21=double(q21_1); c22=double(q22_1); c23=double(q23_1);
183     c31=double(q31_1); c32=double(q32_1); c33=double(q33_1);
184
185     A_new = a
186     B_new =[b1;b2;b3]
187     Q_new = [c11,c12,c13;c21,c22,c23;c31,c32,c33]
188     Y = [1;r;b];
189     k1_new = exp(k0+k1*psi*log(vphi)-k1*(A_new+B_new.'*Y+0.5*Y.'*Q_new*Y));
190     k1 = k1_new
191     deltak = k1_new-k1_old
192     if abs(deltak) < 10^(-4)
193         syms a_1 b1_1 b2_1 b3_1 q11_1 q12_1 q13_1 q21_1 q22_1 q23_1 q31_1
194             q32_1 q33_1
195         A = a_1;
196         B = [b1_1;b2_1;b3_1];
197         Q = [q11_1,q12_1,q13_1;q21_1,q22_1,q23_1;q31_1,q32_1,q33_1];
198         k1=k1_new;
199         k0 = k1*(1-log(k1));
200         const = -k0-psi*k1*log(vphi)+k1*A+vphi*psi+(g0-kbar).'*B...
201             -0.5*B.'*(Sy*Sy.'-phi_4)*B-0.5*trace(Sy*Sy.'*Q)-0.5*(psi-1)*J0;

```

```

202     y = k1*B+(g1+k) .* B-Q*(Sy*Sy.' - phi_4)*B+Q*(g0-kbar) - 0.5*(psi - 1)*J1 - (
        psi - 1)*vrho;
203
204     y2 = k1*Q+(g1+k) .* Q+Q .* (g1+k) - Q .* (Sy*Sy.' - phi_4)*Q - 0.5*(psi - 1)*J2;
205
206     x = [const;y;y2(:)];
207 %     varsx = [a_1, b1_1, b2_1, b3_1, q11_1, q12_1, q13_1, q21_1, q22_1, q23_1,
        q31_1, q32_1, q33_1];
208 %     initx = zeros(1, length(varsx));
209 %     Xsol = vpasolve(x==0, varsx, initx)
210 Xsol = vpasolve(y2==0)
211 if sum(size(Xsol.q11_1))>2
212     if psi < 1
213         idx = intersect(find(Xsol.q11_1>0), find(Xsol.q12_1==0))
214         if length(idx)>1
215             idx = intersect(find(Xsol.q33_1<0), idx)
216         end
217     else
218         idx = intersect(find(Xsol.q11_1<0), find(Xsol.q12_1==0))
219         if length(idx)>1
220             idx = intersect(find(Xsol.q33_1>0), idx)
221         end
222
223     end
224     q11_1=Xsol.q11_1(idx); q12_1=Xsol.q12_1(idx); q13_1=Xsol.q13_1(idx)
        ;
225     q21_1=Xsol.q21_1(idx); q22_1=Xsol.q22_1(idx); q23_1=Xsol.q23_1(idx)
        ;
226     q31_1=Xsol.q31_1(idx); q32_1=Xsol.q32_1(idx); q33_1=Xsol.q33_1(idx)
        ;
227 %     b1_1 = Xsol.b1_1(idx); b2_1= Xsol.b2_1(idx); b3_1 = Xsol.b3_1(
        idx);
228 %     a_1 = Xsol.a_1(idx);
229 else
230     q11_1=Xsol.q11_1; q12_1=Xsol.q12_1; q13_1=Xsol.q13_1;
231     q21_1=Xsol.q21_1; q22_1=Xsol.q22_1; q23_1=Xsol.q23_1;
232     q31_1=Xsol.q31_1; q32_1=Xsol.q32_1; q33_1=Xsol.q33_1;
233 %     b1_1 = Xsol.b1_1; b2_1= Xsol.b2_1; b3_1 = Xsol.b3_1;
234 %     a_1 = Xsol.a_1;
235 end
236 Xsol = vpasolve(subs(y)==0);
237 b1_1 = Xsol.b1_1; b2_1= Xsol.b2_1; b3_1 = Xsol.b3_1;

```

```

238     Xsol = vpsolve(subs(const)==0);
239     a_1 = Xsol;
240     % change to double
241     a=double(a_1);b1=double(b1_1);b2=double(b2_1);b3=double(b3_1);
242     c11=double(q11_1);c12=double(q12_1);c13=double(q13_1);
243     c21=double(q21_1);c22=double(q22_1);c23=double(q23_1);
244     c31=double(q31_1);c32=double(q32_1);c33=double(q33_1);
245     A = a
246     B = [b1;b2;b3]
247     Q = [c11 , c12 , c13 ; c21 , c22 , c23 ; c31 , c32 , c33 ]
248     j=2;
249     end
250 end
251 % finding the portfolios
252 gy = B+Q*Y;
253 Phi = (1/(1-psi))*inv(phi_1)*phi_2*gy+inv(phi_1)*mu;
254 Phi_s = inv(phi_1)*mu;
255 Phi_l = (1/(1-psi))*inv(phi_1)*phi_2*[gy(1);0;0];
256 Phi_r = (1/(1-psi))*inv(phi_1)*phi_2*[0;gy(2);0];
257 Phi_b = (1/(1-psi))*inv(phi_1)*phi_2*[0;0;gy(3)];
258
259 pi_p(i)=Phi(2)
260 pi_s(i) = Phi(1)
261 % speculative demandgamma
262 pi_p_s(i)=Phi_s(2);pi_s_s(i) = Phi_s(1);
263 % hedging demand from unobservability , observability and stochastic
264 % interest rates
265 pi_p_l(i) = Phi_l(2); pi_s_l(i) = Phi_l(1);
266 pi_p_r(i) = Phi_r(2);pi_s_r(i) = Phi_r(1);
267 pi_p_b(i) = Phi_b(2);pi_s_b(i) = Phi_b(1);
268 % consumption to wealth ratio
269 cwr(i)=k1
270
271 i = i+1
272 end
273 save('simulationEZeis4.3.mat','pi_s','pi_p','pi_s_s','pi_p_s'...
274     , 'pi_s_l','pi_p_l','pi_s_r','pi_p_r','pi_s_b','pi_p_b','cwr')
275 fprintf('Variables saved!\n')
276 % time counter end
277 toc
278
279 %% Plots

```

```

280 close all
281 clear all
282 clc
283 tau = linspace(0,10,11);
284 figure
285 subplot(2,1,1)
286 hold on
287 load('simulationEZeis1_40.mat')
288 plot(tau, pi_s, '->')
289 load('simulationEZeis1_20.mat')
290 plot(tau, pi_s, '-o')
291 load('simulationEZeis0_25.mat')
292 plot(tau, pi_s, '-d')
293 load('simulationEZeis0_5.mat')
294 plot(tau, pi_s, '-s')
295 load('simulationEZeis4_3.mat')
296 plot(tau, pi_s, '-^')
297 ylim([-5 1])
298 ylabel('Wealth Allocation'), xlabel('\theta_1=\theta_2=\theta_3\in[0,10]')
299 legend('\psi=1/40', '\psi=1/20', '\psi=1/\gamma', '\psi=1/2', '\psi=4/3')
300 title('(a) Optimal Stock Allocation')
301 %
302 subplot(2,1,2)
303 hold on
304 load('simulationEZeis1_40.mat')
305 plot(tau, pi_p, '->')
306 load('simulationEZeis1_20.mat')
307 plot(tau, pi_p, '-o')
308 load('simulationEZeis0_25.mat')
309 plot(tau, pi_p, '-d')
310 load('simulationEZeis0_5.mat')
311 plot(tau, pi_p, '-s')
312 load('simulationEZeis4_3.mat')
313 plot(tau, pi_p, '-^')
314 ylim([-5 1])
315 xlabel('\theta_1=\theta_2=\theta_3\in[0,10]')
316 ylabel('Wealth Allocation')
317 legend('\psi=1/40', '\psi=1/20', '\psi=1/\gamma', '\psi=1/2', '\psi=4/3')
318 title('(b) Optimal Bond Allocation')
319 %
320 figure
321 subplot(2,1,1)

```

```

322 hold on
323 load('simulationEZeis1_40.mat')
324 plot(tau, pi_s_s, '>')
325 load('simulationEZeis1_20.mat')
326 plot(tau, pi_s_s, '-o')
327 load('simulationEZeis0_25.mat')
328 plot(tau, pi_s_s, '-d')
329 load('simulationEZeis0_5.mat')
330 plot(tau, pi_s_s, '-s')
331 load('simulationEZeis4_3.mat')
332 plot(tau, pi_s_s, '^')
333 ylim([-0.5 0.5])
334 xlabel('\theta_1=\theta_2=\theta_3\in[0,10]')
335 ylabel('Wealth Allocation')
336 legend('\psi=1/40', '\psi=1/20', '\psi=1/\gamma', '\psi=1/2', '\psi=4/3')
337 title('(a) Myopic Stock Demand')
338 %
339 subplot(2,1,2)
340 hold on
341 load('simulationEZeis1_40.mat')
342 plot(tau, pi_p_s, '>')
343 load('simulationEZeis1_20.mat')
344 plot(tau, pi_p_s, '-o')
345 load('simulationEZeis0_25.mat')
346 plot(tau, pi_p_s, '-d')
347 load('simulationEZeis0_5.mat')
348 plot(tau, pi_p_s, '-s')
349 load('simulationEZeis4_3.mat')
350 plot(tau, pi_p_s, '^')
351 ylim([-0.5 0.5])
352 xlabel('\theta_1=\theta_2=\theta_3\in[0,10]')
353 ylabel('Wealth Allocation')
354 legend('\psi=1/40', '\psi=1/20', '\psi=1/\gamma', '\psi=1/2', '\psi=4/3')
355 title('(b) Myopic Bond Demand')
356 %
357 figure
358 subplot(2,1,1)
359 hold on
360 load('simulationEZeis1_40.mat')
361 plot(tau, pi_s_l, '>')
362 load('simulationEZeis1_20.mat')
363 plot(tau, pi_s_l, '-o')

```

```

364 load('simulationEZeis0_25.mat')
365 plot(tau, pi_s_l, '-d')
366 load('simulationEZeis0_5.mat')
367 plot(tau, pi_s_l, '-s')
368 load('simulationEZeis4_3.mat')
369 plot(tau, pi_s_l, '-^')
370 ylim([-2 0.5])
371 xlabel('\theta_1=\theta_2=\theta_3\in[0,10]')
372 ylabel('Wealth Allocation')
373 legend('\psi=1/40', '\psi=1/20', '\psi=1/\gamma', '\psi=1/2', '\psi=4/3')
374 title('(a) Stock Allocation from Observable Parameter Hedging Demand')
375 %
376 subplot(2,1,2)
377 hold on
378 load('simulationEZeis1_40.mat')
379 plot(tau, pi_p_l, '->')
380 load('simulationEZeis1_20.mat')
381 plot(tau, pi_p_l, '-o')
382 load('simulationEZeis0_25.mat')
383 plot(tau, pi_p_l, '-d')
384 load('simulationEZeis0_5.mat')
385 plot(tau, pi_p_l, '-s')
386 load('simulationEZeis4_3.mat')
387 plot(tau, pi_p_l, '-^')
388 ylim([-2 0.5])
389 xlabel('\theta_1=\theta_2=\theta_3\in[0,10]')
390 ylabel('Wealth Allocation')
391 legend('\psi=1/40', '\psi=1/20', '\psi=1/\gamma', '\psi=1/2', '\psi=4/3')
392 title('(b) Bond Allocation from Observable Parameter Hedging Demand')
393 %
394 figure
395 subplot(2,1,1)
396 hold on
397 load('simulationEZeis1_40.mat')
398 plot(tau, pi_s_r, '->')
399 load('simulationEZeis1_20.mat')
400 plot(tau, pi_s_r, '-o')
401 load('simulationEZeis0_25.mat')
402 plot(tau, pi_s_r, '-d')
403 load('simulationEZeis0_5.mat')
404 plot(tau, pi_s_r, '-s')
405 load('simulationEZeis4_3.mat')

```

```

406 plot(tau, pi_s_r, '^')
407 ylim([-0.1 0.5])
408 xlabel('\theta_1=\theta_2=\theta_3\in[0,10]')
409 ylabel('Wealth Allocation')
410 legend('\psi=1/40', '\psi=1/20', '\psi=1/\gamma', '\psi=1/2', '\psi=4/3')
411 title('(a) Stock Allocation from Interest Rate Hedging Demand')
412 %
413 subplot(2,1,2)
414 hold on
415 load('simulationEZeis1_40.mat')
416 plot(tau, pi_p_r, '>')
417 load('simulationEZeis1_20.mat')
418 plot(tau, pi_p_r, '-o')
419 load('simulationEZeis0_25.mat')
420 plot(tau, pi_p_r, '-d')
421 load('simulationEZeis0_5.mat')
422 plot(tau, pi_p_r, '-s')
423 load('simulationEZeis4_3.mat')
424 plot(tau, pi_p_r, '^')
425 ylim([-0.1 0.5])
426 xlabel('\theta_1=\theta_2=\theta_3\in[0,10]')
427 ylabel('Wealth Allocation')
428 legend('\psi=1/40', '\psi=1/20', '\psi=1/\gamma', '\psi=1/2', '\psi=4/3')
429 title('(b) Bond Allocation from Interest Rate Hedging Demand')
430 %
431 figure
432 subplot(2,1,1)
433 hold on
434 load('simulationEZeis1_40.mat')
435 plot(tau, pi_s_b, '>')
436 load('simulationEZeis1_20.mat')
437 plot(tau, pi_s_b, '-o')
438 load('simulationEZeis0_25.mat')
439 plot(tau, pi_s_b, '-d')
440 load('simulationEZeis0_5.mat')
441 plot(tau, pi_s_b, '-s')
442 load('simulationEZeis4_3.mat')
443 plot(tau, pi_s_b, '^')
444 ylim([-4 0.5])
445 xlabel('\theta_1=\theta_2=\theta_3\in[0,10]')
446 ylabel('Wealth Allocation')
447 legend('\psi=1/40', '\psi=1/20', '\psi=1/\gamma', '\psi=1/2', '\psi=4/3')

```

```

448 title(' (a) Stock Allocation from Unobservable Parameter Hedging Demand')
449 %
450 subplot(2,1,2)
451 hold on
452 load('simulationEZeis1_40.mat')
453 plot(tau, pi_p_b, '->')
454 load('simulationEZeis1_20.mat')
455 plot(tau, pi_p_b, '-o')
456 load('simulationEZeis0_25.mat')
457 plot(tau, pi_p_b, '-d')
458 load('simulationEZeis0_5.mat')
459 plot(tau, pi_p_b, '-s')
460 load('simulationEZeis4_3.mat')
461 plot(tau, pi_p_b, '-^')
462 ylim([-4 0.5])
463 xlabel('\theta_1=\theta_2=\theta_3\in [0,10]')
464 ylabel('Wealth Allocation')
465 legend('\psi=1/40', '\psi=1/20', '\psi=1/\gamma', '\psi=1/2', '\psi=4/3')
466 title(' (b) Bond Allocation from Unobservable Parameter Hedging Demand')
467 %
468 figure
469 hold on
470 load('simulationEZeis1_40.mat')
471 plot(tau, cwr, '->')
472 load('simulationEZeis1_20.mat')
473 plot(tau, cwr, '-o')
474 load('simulationEZeis0_25.mat')
475 plot(tau, cwr, '-d')
476 load('simulationEZeis0_5.mat')
477 plot(tau, cwr, '-s')
478 load('simulationEZeis4_3.mat')
479 plot(tau, cwr, '-^')
480 xlabel('\theta_1=\theta_2=\theta_3\in [0,10]')
481 ylabel('Wealth Allocation')
482 legend('\psi=1/40', '\psi=1/20', '\psi=1/\gamma', '\psi=1/2', '\psi=4/3')
483 title('Optimal Consumption to Wealth Ratio')

```

REFERENCES

- [1] Hansjorg Albrecher and Martin Predota. On pricing Asian options on NIG Lévy processes. *Journal of Computational and Applied Mathematics*, 2004.
- [2] Jennifer Alonso-Garvía, Oliver. Wood, and Jonathan Ziveyi. Pricing and hedging guaranteed minimum withdrawal benefits under a general Lévy framework using the COS method. *Quantitative Finance*, 2017.
- [3] Louis. Bachelier. The theory of speculation. Master’s thesis, L’Ecole Normale Superieure, 1900.
- [4] Fred E. Benth and Paul Kruhner. Integrability of multivariate subordinated Lévy processes in Hilbert space. *Stochastic: An International Journal of Probability and Stochastic Processes*, 2015.
- [5] Fischer Black and Myron Scholes. The pricing of options and liabilities. *The Journal of Political Economy*, 1973.
- [6] Bruno Bouchard, Romuald Elie, and Nizar Touzi. Stochastic target problems with controlled loss. *SIAM Journal on Control and Optimization*, 2008.
- [7] Wissem Boughamoura, Anand N. Pandey, and Faozi Trabelsi. Pricing and hedging of asian option under jumps. *International Journal of Applied Mathematics*, 2011.
- [8] John. Campbell, Yeung. Chan, and Luis Viceira. A multivariate model of strategic asset allocation. *Journal of Financial Economics*, 2003.
- [9] John Campbell, Jorge Rodriguez, and Luis Viceira. Strategic asset allocation in a continuous-time var model. *Journal of Economic Dynamics and Control*, 2004.
- [10] John Campbell and Luis Viceira. Consumption and portfolio decisions when expected returns are time varying. *Quarterly Journal of Economics*, 1999.

- [11] John Campbell and Luis Viceira. *Strategic Asset Allocation: Portfolio Choice for Long-Term Investors*. Oxford Press, 2001.
- [12] Peter K. Clark. A subordinated stochastic process model with finite variance for speculative prices. *Econometrica*, 1973.
- [13] C.W. Clenshaw and A.R. Curtis. A method for numerical integration on an automatic computer. *Numerische Mathematik*, 1960.
- [14] Rama Cont and Peter Tankov. *Financial Modelling With Jump Processes*. CRC Press, 2003.
- [15] Rama Cont and Peter Tankov. *Calibration of Jump-Diffusion Option Pricing Models: A Robust Non-Parametric Approach*. Ecole Polytechnique, Centre de Mathématiques Appliquées, 2002.
- [16] Paresh Date and Ksenia Ponomareva. Linear and nonlinear filtering in mathematical finance: a review. *IMA Journal of Management Mathematics*, 2011.
- [17] Griselda Deelstra, Alexandre Petkovic, and Michèle Vanmaele. Pricing and hedging Asian basket spread options. *Journal of Computational and Applied Mathematics*, 2010.
- [18] Sondermann Dieter. *Introduction to Stochastic Calculus for Finance A New Didactic Approach*. Springer, 2006.
- [19] Darrell Duffie and Larry G Epstein. Stochastic differential utility. *Econometrica*, 1992.
- [20] Marcos Escobar, Sebastian Ferrando, and Alexey Rubtsov. Portfolio choice with stochastic interest rates and learning about stock return predictability. *International Review of Economics and Finance*, 2016.
- [21] Christian-Oliver Ewald, Zhaojun Yang, and Yajun Xiao. Implied volatility from Asian options via Monte Carlo methods. *International Journal of Theoretical and Applied Finance*, 2009.
- [22] Fang Fang and Cornelis W. Oosterlee. A novel pricing method for European options based on Fourier-cosine series expansion. *SIAM Journal of Scientific Computing*, 2008.
- [23] Hans Föllmer and Peter Leukert. Quantile Hedging. *Finance and Stochastics*, 1999.
- [24] Hans Föllmer and Martin Schweizer. *Hedging of Contingent Claims Under Incomplete Information*. University of Bonn, 1990.

- [25] Gerard Genotte. Optimal portfolio choice under incomplete information. *The Journal of Finance*, 1986.
- [26] Nengjiu Ju and Jianjun Miao. Ambiguity, learning and asset returns. *Econometrica*, 2012.
- [27] Adam Kolkiewicz and Yan Liu. Semi-static hedging for GMWB in variable annuities. *North American Actuarial Journal*, 2012.
- [28] Weiqing Li and Su Chen. Pricing and hedging of arithmetic Asian options via the Edgeworth series expansion approach. *Journal of Finance and Data Science*, 2016.
- [29] Robert Liptser and Albert Shiryaev. *Statistics of Random Processes, Volume I*. Springer, 2000.
- [30] Robert Liptser and Albert Shiryaev. *Statistics of Random Processes, Volume II*. Springer, 2000.
- [31] Hening Liu. Robust consumption and portfolio choice for time varying investment opportunities. *Annals of Finance*, 2010.
- [32] Pascal J. Maenhout. Robust portfolio rules and asset pricing. *The Review of Financial Studies*, 2004.
- [33] Robert Merton. An intertemporal capital asset model. *Econometrica*, 1973.
- [34] Robert Merton. Theory of rational option pricing. *The Bell Journal of Economics and Management Science*, 1973.
- [35] John R. Michael, William R. Schucany, and Roy W. Haas. Generating random variables using transformations with multiple roots. *The American Statistician*, 1976.
- [36] Ludovic Moreau. Stochastic target problems with controlled loss in jump diffusion models. *SIAM Journal on Control and Optimization*, 2011.
- [37] Bernt Karsten Øksendal and Agnès Sulem. *Applied stochastic control of jump diffusions*, volume 498. Springer, 2005.
- [38] Tommaso Pellegrino and Sabino Piergiacomo. Pricing and hedging multi-asset spread options by a three-dimensional Fourier cosine series expansion method. *Journal of Energy Markets*, 2014.
- [39] George Pennachi. *The Theory of Asset Pricing*. Pearson Education, 2008.

- [40] Philip E. Protter. Stochastic Differential Equations. In *Stochastic Integration and Differential Equations*, pages 249–361. Springer, 2005.
- [41] Sheldon M Ross. *Simulation*. Academic Press, 2012.
- [42] Marjon J. Ruijter and Cornelis W. Oosterlee. Two-dimensional Fourier cosine series expansion method for pricing financial options. *SIAM Journal of Scientific Computing*, 2012.
- [43] Alexander Schied. *Lecture Notes of a Minicourse held at the 8th Symposium on Probability and Stochastic Processes at Universidad de las Américas, Puebla*. TU Berlin, Institut für Mathematik, 2004.
- [44] Wim Schoutens. *Lévy processes in Finance*. John Wiley & Sons, 2003.
- [45] Martin Schweizer. Mean-variance hedging for general claims. *Annals of Applied Probability*, 1992.
- [46] Virginia Torczon. On the convergence of pattern search algorithms. *SIAM Journal of Optimization*, 1997.
- [47] Bowen Zhang and Cornelis W. Oosterlee. Efficient pricing of european-style asian options under exponential Lévy processes based on Fourier cosine expansions. *SIAM Journal of Financial Mathematics*, 2013.