Numerical Methods for Pricing Exotic Options

by

Hardik Dave - 00517958
Supervised by Dr. Daniel Kuhn
Second Marker: Professor Berç Rustem

Submitted in partial fulfilment of the requirements for the MSc Degree in Advanced Computing of Imperial College London

September 2008
Abstract

Since the introduction of Fischer Black and Myron Scholes’ famous option pricing model [3] in 1973, several authors have proposed alternative models and methodologies for pricing options accurately. In this project, we evaluate and extend two such recently proposed methodologies.

In the semi-parametric approach for pricing options, originally proposed by Lo in [22] and subsequently extended by Bertsimas and Popescu in [1] and [2] and Gotoh and Konno in [9], the option price is identified with associated semidefinite programming (SDP) problems, the solution of which provides tight upper and lower bounds, given the first $n$ moments of the distribution of the underlying security price. While Gotoh and Konno used their own cutting plane algorithm for solving the associated SDP problems in [9], we derive and solve such problems with our own software in this project.

Subsequently, in [21], Lasserre, Prieto-Rumeau and Zervos introduced a new methodology for numerical pricing of exotic derivatives such as Asian and down-and-out Barrier options. In their methodology, the underlying asset price dynamics are modeled by geometric Brownian motion or other mean-reverting processes. For pricing derivatives, they solve a finite dimensional SDP problem (utilizing their own software GloptiPoly), derived from another infinite dimensional linear programming problem with moments of appropriate measures.

In this project we investigate and build upon the theory of both of the above approaches. We also extend the theory developed by Lasserre, Prieto-Rumeau and Zervos to model the SDP problems for pricing European, up-and-out and double Barrier options. Direct implementation of software for solving the associated SDP problems for pricing the European, Asian and (down-and-out, up-and-out and double) Barrier options is given along with detailed analysis of the results. Secondly, reimplementation of numerical methodology given by Gotoh and Konno is also given, solving the associated SDP problem directly, without using their Cutting Plane algorithm. The efficiency of the above two methods is analysed and compared with other standard option pricing techniques such as Monte Carlo methods.
Acknowledgements

First of all, I would like to acknowledge Dr. Daniel Kuhn who provided me with constant guidance and encouragement throughout the project. I am deeply indebted to Dr. Kuhn for embarking with me on this project journey. I could not have wished for better supervisor and collaborator. Your contributions, insightful comments and awe inspiring conversations have been of great value to me.

I would also like to acknowledge Professor Berç Rustem for his constructive comments and agreeing to be the second marker for this project.

I also acknowledge my family and friends for their moral and intellectual support during this project. Lastly, I would like to thank Nikita for being the indefatigable and untiring source of strength for me, when I needed it most.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td></td>
<td>vi</td>
</tr>
<tr>
<td>List of Tables</td>
<td></td>
<td>vii</td>
</tr>
<tr>
<td>1</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Two Recent Methodologies</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Contributions</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Options and Valuation Techniques</td>
<td>4</td>
</tr>
<tr>
<td>2.1</td>
<td>Class of Options</td>
<td>4</td>
</tr>
<tr>
<td>2.1.1</td>
<td>European Options</td>
<td>4</td>
</tr>
<tr>
<td>2.1.2</td>
<td>Asian Options</td>
<td>5</td>
</tr>
<tr>
<td>2.1.3</td>
<td>Barrier Options</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Models for Underlying Asset Price</td>
<td>7</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Geometric Brownian Motion</td>
<td>7</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Ornstein-Uhlenbeck Process</td>
<td>8</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Cox Ingersoll Ross process</td>
<td>9</td>
</tr>
<tr>
<td>2.3</td>
<td>Monte Carlo Methods</td>
<td>10</td>
</tr>
<tr>
<td>2.3.1</td>
<td>European Options</td>
<td>10</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Asian Options</td>
<td>11</td>
</tr>
<tr>
<td>2.3.3</td>
<td>Barrier Options</td>
<td>12</td>
</tr>
<tr>
<td>2.3.4</td>
<td>Advantages and Disadvantages of Monte Carlo Methods</td>
<td>15</td>
</tr>
<tr>
<td>2.4</td>
<td>Black Scholes PDE and Analytical Solution</td>
<td>15</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Analytical Solution for European Options</td>
<td>16</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Analytical Solution for Barrier Options</td>
<td>16</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Assumptions and Shortcomings of Black Scholes Formula</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>Semi-Parametric Approach for Bounding Option Price</td>
<td>20</td>
</tr>
<tr>
<td>3.1</td>
<td>Review of Basic Results</td>
<td>20</td>
</tr>
<tr>
<td>3.2</td>
<td>SDP Formulation of the Bounding Problems</td>
<td>25</td>
</tr>
<tr>
<td>3.3</td>
<td>Computing the Moments</td>
<td>29</td>
</tr>
<tr>
<td>4</td>
<td>Method of Moments and SDP Relaxations</td>
<td>31</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>31</td>
</tr>
<tr>
<td>4.2</td>
<td>Deriving Infinite-Dimensional LP Problem</td>
<td>32</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Basic Adjoint Equation</td>
<td>32</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Martingale Moment Conditions</td>
<td>34</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Using Available Moments</td>
<td>35</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Using Martingale Moments Conditions</td>
<td>35</td>
</tr>
<tr>
<td>4.3</td>
<td>Polynomial Optimisation and Problem of Moments</td>
<td>36</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Notation and Definition</td>
<td>36</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Necessary Moment conditions</td>
<td>37</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Value of European Option at Expiration</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>Value of Barrier Call Option</td>
<td>7</td>
</tr>
<tr>
<td>4.1</td>
<td>Definition of Measures for European Options</td>
<td>41</td>
</tr>
<tr>
<td>4.2</td>
<td>Definition of Measures for Asian Options</td>
<td>44</td>
</tr>
<tr>
<td>4.3</td>
<td>Definition of Measures for Down-And-Out Barrier Options</td>
<td>48</td>
</tr>
<tr>
<td>4.4</td>
<td>Definition of Measures for Up-And-Out Barrier Options</td>
<td>50</td>
</tr>
<tr>
<td>4.5</td>
<td>Definition of Measures for Double Barrier Options</td>
<td>51</td>
</tr>
<tr>
<td>5.1</td>
<td>European Call Option: Geometric Brownian Motion</td>
<td>57</td>
</tr>
<tr>
<td>5.2</td>
<td>European Call Option: Geometric Brownian Motion</td>
<td>59</td>
</tr>
<tr>
<td>5.3</td>
<td>European Call Option: Ornstein-Uhlenbeck Process</td>
<td>61</td>
</tr>
<tr>
<td>5.4</td>
<td>European Call Option: Cox Ingersoll Ross Process</td>
<td>63</td>
</tr>
<tr>
<td>5.5</td>
<td>Asian Call Option: Geometric Brownian Motion</td>
<td>65</td>
</tr>
<tr>
<td>5.6</td>
<td>Asian Call Option: Ornstein Uhlenbeck Process</td>
<td>67</td>
</tr>
<tr>
<td>5.7</td>
<td>Asian Call Option: Cox Ingersoll Ross Process</td>
<td>69</td>
</tr>
<tr>
<td>5.8</td>
<td>Down and Out Barrier Call Option: Ornstein Uhlenbeck Process</td>
<td>71</td>
</tr>
<tr>
<td>5.9</td>
<td>Down and Out Barrier Call Option: Cox Ingersoll Ross Process</td>
<td>73</td>
</tr>
<tr>
<td>5.10</td>
<td>Up and Out Barrier Call Option: Ornstein Uhlenbeck Process</td>
<td>75</td>
</tr>
<tr>
<td>5.11</td>
<td>Up and Out Barrier Call Option: Cox Ingersoll Ross Process</td>
<td>77</td>
</tr>
<tr>
<td>5.12</td>
<td>Combined Barrier Call Option: Ornstein Uhlenbeck Process</td>
<td>79</td>
</tr>
<tr>
<td>5.13</td>
<td>Combined Barrier Call Option: Cox Ingersoll Ross Process</td>
<td>81</td>
</tr>
<tr>
<td>5.14</td>
<td>Semi-parametric Approach, European Call Option: Geometric Brownian Motion</td>
<td>83</td>
</tr>
<tr>
<td>5.15</td>
<td>Semi-parametric Approach, European Call Option: Geometric Brownian Motion</td>
<td>85</td>
</tr>
<tr>
<td>6.1</td>
<td>Example of Complex Double Barrier Option</td>
<td>87</td>
</tr>
<tr>
<td>A.1</td>
<td>Semi-parametric Approach, European Call Option: Geometric Brownian Motion</td>
<td>91</td>
</tr>
<tr>
<td>A.2</td>
<td>European Call Option: Geometric Brownian Motion</td>
<td>93</td>
</tr>
<tr>
<td>A.3</td>
<td>European Call Option: Ornstein-Uhlenbeck Process</td>
<td>95</td>
</tr>
<tr>
<td>A.4</td>
<td>European Call Option: Cox Ingersoll Ross Process</td>
<td>97</td>
</tr>
<tr>
<td>A.5</td>
<td>Asian Call Option: Geometric Brownian Motion</td>
<td>99</td>
</tr>
<tr>
<td>A.6</td>
<td>Asian Call Option: Ornstein Uhlenbeck Process</td>
<td>101</td>
</tr>
<tr>
<td>A.7</td>
<td>Asian Call Option: Cox Ingersoll Ross Process</td>
<td>103</td>
</tr>
<tr>
<td>A.8</td>
<td>Down and Out Barrier Call Option: Ornstein Uhlenbeck Process</td>
<td>105</td>
</tr>
<tr>
<td>A.9</td>
<td>Down and Out Barrier Call Option: Cox Ingersoll Ross Process</td>
<td>107</td>
</tr>
<tr>
<td>A.10</td>
<td>Up and Out Barrier Call Option: Ornstein Uhlenbeck Process</td>
<td>109</td>
</tr>
<tr>
<td>A.11</td>
<td>Up and Out Barrier Call Option: Cox-Ingersoll-Ross Process</td>
<td>111</td>
</tr>
<tr>
<td>A.12</td>
<td>Double Barrier Call Option: Ornstein Uhlenbeck Process</td>
<td>113</td>
</tr>
<tr>
<td>A.13</td>
<td>Double Barrier Call Option: Cox Ingersoll Ross Process</td>
<td>115</td>
</tr>
</tbody>
</table>
List of Tables

A.1 European Call Option Values: Geometric Brownian Motion ................. 90
A.2 European Call Option Values: Geometric Brownian Motion ................. 92
A.3 European Call Option Values: Ornstein-Uhlenbeck Process ................. 94
A.4 European Call Option: Cox Ingersoll Ross Process ......................... 96
A.5 Asian Call Option: Geometric Brownian Motion ........................... 98
A.6 Asian Call Option: Ornstein Uhlenbeck Process ............................ 100
A.7 Asian Call Option: Cox Ingersoll Ross Process ............................. 102
A.8 Down and Out Barrier Call Option: Ornstein Uhlenbeck Process .......... 104
A.9 Down and Out Barrier Call Option: Cox Ingersoll Ross Process .......... 106
A.10 Up and Out Barrier Call Option: Ornstein Uhlenbeck Process .......... 108
A.11 Up and Out Barrier Call Option: Cox Ingersoll Ross Process .......... 110
A.12 Double Barrier Call Option: Ornstein Uhlenbeck Process ............... 112
A.13 Double Barrier Call Option: Cox Ingersoll Ross Process ............... 114
Chapter 1

Introduction

An option is defined as a financial instrument that gives its holder the right, but not the obligation, to buy (call option) or sell (put option) a predefined underlying security at a set exercise price at a specified maturity date. Options are a form of derivative security, as they derive their values from their underlying asset. An example of a very common option is a single asset option which consists of one underlying asset, for example stock option which includes a defined number of shares in a listed public company.

There are several types of options, differentiated on the basis of maturity date, the method of pay off calculation, etc. For example, European options can only exercised at the maturity date, as compared to American option which may be exercised on any trading day on or before expiration. Other type of options include Bermudan option which may be exercised only on specified dates on or before expiration and Barrier option which depends on underlying security’s price reaching some barrier before the exercise can occur. Although above type of options are most common, they are not exhaustive. There are several numerical variations on the basic design of above options for control of risk as perceived by investors and which eases execution and book keeping.

Based on the concept of risk neutral pricing and using stochastic calculus, variety of numerical models and methodologies have been used to determine the theoretical value of an option. In their pioneering Noble Prize winning work, Fischer Black and Myron Scholes devised the first quantitative model (called Black-Scholes option pricing model in [3]) for pricing of a variety of simple option contracts. The basis if their technique was Black-Scholes partial differential equation (PDE) which must be satisfied by the price of any derivative dependent on a non dividend paying stock. By creating a risk neutral portfolio that replicates the returns of holding an option, a closed-form solution for an European option’s theoretical price was obtained.

However, it was discovered that the Black-Scholes model made several assumptions which lead to non eligible pricing biases. Two important tenets on which Black-Scholes model was built were that there were no-arbitrage opportunities, and that the underlying asset (stock) followed a geometric Brownian motion. However, it is well known and proved that stock prices do not follow geometric Brownian motion(for example, see [14]). Therefore, there has been research into pricing models that do not have the underlying log-normal distribution assumption.

1.1 Two Recent Methodologies

Recently, there have been approaches using semidefinite programming to derive bounds based on the moments of the underlying asset price. An excellent introduction to semidefinite programming is given by Vandenbergh in [28] and for a detailed reference, reader is encouraged to refer [28]. It has been shown that bounds on the option prices can be derived from only the no-arbitrage assumption(see [22]).

In [22], Lo derived a distribution-free upper bound on single call and put options given the
mean and variance of the underlying asset price. These bounds were termed as semi-parametric because they depended only on the mean and variance of the terminal asset price and not its entire distribution. In [4], Boyle and Lin extended Lo’s result to the multi-asset case where they utilized the correlations between the assets for determining the upper bound of a call option. It is to be noted that in both Lo’s and Boyle and Lin’s approach, the assumption that price of underlying asset following a lognormal distribution is not required and hence the upper bounds achieved are distribution-free. Using mean and covariance matrix of the underlying asset prices, Boyle and Lin show that the upper bound can be achieved by solving the associated SDP (solved by an interior point algorithm).

Extending Boyle and Lin’s work, Bertsimas and Popescu, in [2] and [1], developed a distribution-free tight upper bound on a standard single asset option, given the first $n$ moments of the underlying asset price. They also showed that this upper bound can be found by solving it as an SDP problem.

Subsequently, the work of Bertsimas and Popescu was extended by Gotoh and Konno in [9], by finding a tight lower bound for a single asset option by solving another SDP problem. They solved the SDP problem with a cutting plane algorithm which they presented in [8], an approach similar to what was developed in [11]. The central idea of their algorithm was to view a semidefinite programming problem as a linear program with an infinite number of linear constraints. They achieved impressive numerical results in real time and derived upper and lower bounds which were extremely close to the actual option price, especially as the number of moments became large, i.e. $n \geq 4$.

In [10], Han, Li and Sun generalized the above two approaches, to find tight upper bound of the price of a multi-asset European call option by solving a sequence of semidefinite programming problems. In particular their work generalized Boyle and Lin’s work as they have first $n$ moments rather than only two moments. Their work also generalized Bertsimas and Popescu’s and Gotoh and Konno’s work as they derive the bounds for multi-assets and not single assets. Their results showed that as the dimension of the semidefinite relaxations increases, the bound becomes more accurate and converges to the tight bound.

In a related approach, we also choose to extend and reimplement the work done by Lasserre, Prieto-Rumeau and Zervos in [21]. In their original paper, the authors focus on fixed-strike, arithmetic average Asian and down-and-out Barrier options. The underlying asset price dynamics in their model is not restricted to geometric Brownian motion, as their model is applicable to number of other diffusions with a special property. Their technique is based on methodology of moments introduced by Dawson in [7] for solving measure-valued martingale problems which represented geostochastic systems.

In brief, the technique is described as follows. The price of an exotic option has to be identified with a linear combination of moments of appropriately defined probability measures. In the next step, an infinite system of linear equations is derived using the martingale property of associated stochastic integrals. These infinite linear equations involve the moments of associated measures as its variables. In order to solve this infinite dimensional linear problem, finite dimensional semidefinite relaxations are obtained by involving only finite number or moments. This is done by introducing extra constraints called moment conditions (in form of Moment and Localizing matrices) which reflect necessary conditions for a set of scalars to be identified with moments of a measure supported on a given set. The approach of using SDP relaxations and problem of moments was introduced by Lasserre in [18] for global optimization of polynomials. The authors have given their own reference implementation, a MATLAB software GloptiPoly in [21], which in turn used SeDuMi as the underlying SDP solver developed in [16].
1.2 Contributions

The major contributions and achievements of this project are summarised below.

- We develop and elaborate the theory for the two recently proposed methodologies, specifically for semi-parametric approach by Gotoh and Konno in chapter 3 and for method of moments and SDP relaxations by Lasserre, Prieto-Rumeau and Zervos in chapter 4.

- The theory of semi-parametric approach is handled with more detailed explanations in chapter 3 than given originally by Gotoh and Konno in [9].

- The theory given by Lasserre, Prieto-Rumeau and Zervos is extended to develop the SDP problems for pricing European, up-and-out and double Barrier options in chapter 4.

- The theory in chapter 4 also includes explanation for problem of moments and SDP relaxations for global optimization of polynomials.

- While developing the theory in chapter 4, we also correct several minor errors which were present in the original paper [21] by Lasserre, Prieto-Rumeau and Zervos.

- The numerical methodology given by Gotoh and Konno was reimplemented and associated SDP problems were solved using our own programs directly, without using their Cutting Plane algorithm. The detailed numerical results are analysed in section 5.2 and appendix A.

- The SDP problems obtained in chapter 4 were solved using our own reference implementation for pricing following 5 type of options: European, Asian and (down-and-out, up-and-out and double) Barrier options. The comprehensive numerical results are analysed in section 5.1 and appendix A.

- In chapter 2, we introduce the type of options and the underlying asset price processes considered in this project. We also have a look at some of the standard option pricing techniques such as Monte Carlo methods, and present several analytical formulae based on Black-Scholes PDE for calculating option prices.

- Finally, we analyse and compare the above two methods for efficiency and accuracy with other standard option pricing techniques such as Monte Carlo methods in detail, in chapter 5 and present our conclusions in chapter 6. The comprehensive data obtained from numerical experiments done with the two approaches developed in chapters 3 and 4, is listed in Appendix A in tabular and graphical format.
Chapter 2

Options and Valuation Techniques

In this section, we present the options on which our analysis focuses in this project. We also introduce the standard analytical and numerical techniques used for pricing those options. We use these techniques for comparison with the two approaches (semi-parametric approach and method of moments with SDP relaxations) which we introduce and implement in later chapters. Specifically, we introduce the type of options in section 2.1 briefly. Then we explain the type of underlying asset price diffusions in section 2.2 along with analytical solution of geometric Brownian motion and Ornstein Uhlenbeck process. We then introduce the Monte Carlo methods for pricing European, Asian and Barrier Options in section 2.3. Finally, the Black-Scholes framework and analytical formulas for pricing call and put versions of European and (down-and-out and up-and-out) Barrier options are explained in section 2.4.

2.1 Class of Options

An option is a contract, or a provision of a contract, that gives one party (the option holder) the right, but not the obligation, to perform a specified transaction with another party (the option issuer or option writer) according to specified terms. Option contracts are a form of derivative instrument and could be linked to a variety of underlying assets, such as bonds, currencies, physical commodities, swaps, or baskets of assets.

Options take many forms. The two most common are:

1. Call options, which provide the holder the right to purchase an underlying asset at a specified price;
2. Put options, which provide the holder the right to sell an underlying asset at a specified price.

In the next few sections, we introduce the three category of options which we interact with in this project: European, Asian and Barrier options.

2.1.1 European Options

The strike price of a call (put) option is the contractual price at which the underlying asset will be purchased (sold) in the event that the option is exercised. The last date on which an option can be exercised is called the expiration date and it can be categorised in two types as below, on the basis of exercise time:

1. With American exercise, the option can be exercised at any time up to the expiration date.
2. With European exercise, the option can be exercised only on the expiration date.
2 Options and Valuation Techniques

(a) Call option

(b) Put option

Figure 2.1: Value of European Option at Expiration

If underlying asset price is denoted by $X$, the value of European call Option written on $X$ is given by

$$v_E(x_0) = e^{-\rho T}E[(X_T - K)^+]$$

(2.1.1)

where $(X_T - K)^+$ denotes $\max(X_T - K, 0)$, $T \geq 0$ is the option's maturity time, $K$ is the strike price, $\rho$ is the constant discounting factor and $x_0$ is the initial underlying asset price. The value of European call and put options are shown in figure 2.1.

2.1.2 Asian Options

An Asian option (also known as average option) is an option whose payoff is related to the average value of the underlying for specific duration in the life of the option. The two basic type of Asian options are:

1. The payoff of average rate option is based on the difference between a the average value of the underlying during the life of the option and a fixed strike.

2. An average strike option similar to European option except that its strike is set equal to the average value of the underlying during the life of the option.

Asian options are attractive to end users because of following reasons:

1. Asian Options are usually less expensive and sell at lower premiums because volatility of average value of underlying is usually less than volatility in price of underlying.
2.1 Class of Options

2. Asian options offer protection, as manipulation in price of underlying asset is not possible over prolonged period of time.

3. End users of commodities which tend to be exposed to average prices over time find Asian options useful.

If underlying asset price is denoted by \( X \), the value of fixed strike arithmetic-average Asian call option written on \( X \) is given by

\[
v_A(x_0) = e^{-\rho T} \mathbb{E} \left[ \left( \frac{1}{T} \int_0^T X_T dt - K \right)^+ \right]
\]

(2.1.2)

where \((X - Y)^+\) denotes \(\max(X - Y, 0)\), \(T \geq 0\) is the option’s maturity time, \(K\) is the strike price, \(\rho\) is the constant discounting factor and \(x_0\) is the initial underlying asset price.

2.1.3 Barrier Options

A barrier option is a path dependent option that has one of the following two features:

1. A knock-out feature causes the option to immediately terminate if the underlying asset’s price reaches a specified (upper or lower) barrier level, or

2. A knock-in feature causes the option to become effective only if the underlying asset’s price first reaches a specified (upper or lower) barrier level.

Hence for a simple Barrier option, we can derive eight different flavours, i.e. put and call for each of the following:

1. Up and in
2. Down and in
3. Up and out
4. Down and out

Due to the contingent nature of the option, they tend to be priced lower than for a corresponding vanilla option.

If underlying asset price is denoted by \( X \), the value of Barrier call option written on \( X \) is given by

\[
v_A(x_0) = e^{-\rho T} \mathbb{E} \left[ (X_T - K)^+ I_{\{\tau = T\}} \right]
\]

(2.1.3)

where \((X_T - K)^+\) denotes \(\max(X_T - K, 0)\), \(T \geq 0\) is the option’s maturity time, \(K\) is the strike price, \(\rho\) is the constant discounting factor, \(x_0\) is the initial underlying asset price and \(\tau\) is the \((\mathcal{F}_t)\)-stopping time defined by

\[
\tau_d = \inf \{t \geq 0 | X_t \leq H_l \} \land T \\
\tau_u = \inf \{t \geq 0 | X_t \geq H_u \} \land T \\
\tau_c = \inf \{t \geq 0 | X_t \leq H_l \land X_t \geq H_u \} \land T
\]

(2.1.4)

for down-and-out, up-and-out and double Barrier options, respectively, and \(H_l, H_u\) are lower and upper knock-out barriers. As an example, the value of down-and-out Barrier call option is shown in figure 2.2.
2 Options and Valuation Techniques

(a) Down-and-Out Call option

Figure 2.2: Value of Barrier Call Option

2.2 Models for Underlying Asset Price

Given the filtered probability space, \((\Omega, \mathcal{F}, \mathcal{F}_t, P)\), where \(\Omega\) is given sample space, \(\sigma\)-algebra \(\mathcal{F}\) of subsets of \(\Omega\), a measure \(P\) on \((\Omega, \mathcal{F})\), such that \(P(\Omega) = 1\) and a filtration \(\mathcal{F}_t\) on \(\mathcal{F}\). For this project we will consider that the underlying asset prices of derivatives, which we mentioned in previous section, would satisfy the stochastic differential equation:

\[
dX_t = b(X_t) \, dt + \sigma(X_t) \, dW_t, \quad X_0 = x_0 \in I \tag{2.2.1}
\]

where \(W_t\) is the Wiener process, \(I\) is either \((0, +\infty)\) or \(\mathbb{R}\) and \(b, \sigma : I \to I\) are given functions and which enable equation (2.2.1) to have a strong\(^1\) solution contained in \(I\), for all \(t \geq 0\), probability almost surely.

Specifically, for this project, we assume that underlying asset price \(X\) is given by three models: geometric Brownian motion, Ornstein-Uhlenbeck process and Cox-Ingersoll-Ross process which are termed as Model 1, Model 2 and Model 3, respectively in this project. We describe each of above diffusions in detail in following sections and give analytical formula for solution of geometric Brownian motion and Ornstein-Uhlenbeck process.

Before proceeding to next section, we note that any diffusions, other than the ones mentioned above, could be considered aptly for analysis of numerical methodology proposed by Lasserre, Prieto-Rumeau and Zervos in [21] provided that they satisfy this important assumption: the infinitesimal generator of this diffusion should map polynomials into polynomials of same or smaller degree.

2.2.1 Geometric Brownian Motion

A continuous-time stochastic process in which the logarithm of the randomly varying quantity follows a Brownian motion, or a Wiener process is called geometric Brownian motion (GBM). In other words, a stochastic process \(X_t\) is said to follow a GBM, if it satisfies the following stochastic differential equation:

\[
dX_t = \mu X_t \, dt + \sigma X_t \, d\epsilon(t) \tag{2.2.2}
\]

where \(d\epsilon(t) = \epsilon(t) \sqrt{dt}\) is a Wiener process, \(\epsilon(t)\) is a standardized normal random variable, \(\mu\) is called as the percentage drift and \(\sigma\) is called the percentage volatility.

\(^1\)Compared to weak solution which consists of a probability space and a process that satisfies the SDE, a strong solution is a process that satisfies the equation and is defined on a given probability space.
We will now derive an analytical formula for $X_T$, the price of underlying asset at time $T$, when we are given $X_0$ and other parameters of GBM. Now Ito’s Lemma states that for a random process $x$ defined by the Itô process

$$dx(t) = a(x,t)dt + b(x,t)dz$$ (2.2.3)

where $z$ is a standard Wiener process, process $y(t)$ defined by $y(t) = F(x,t)$ satisfies the Itô equation

$$dy(t) = \left(a \frac{\partial F}{\partial x} + \frac{\partial F}{\partial t} + b^2 \frac{1}{2} \frac{\partial^2 F}{\partial x^2}\right) dt + b \frac{\partial F}{\partial x} dz$$ (2.2.4)

where $z$ is the same Wiener process in (2.2.2).

If we apply Itô’s Lemma to the process $F(X(t)) = \ln X(t)$, then we identify

$$a = \mu X, \quad b = \sigma X, \quad \frac{\partial F}{\partial X} = \frac{1}{X}, \quad \text{and} \quad \frac{\partial^2 F}{\partial X^2} = -\frac{1}{X^2}$$

Substituting these values in (2.2.4), we get

$$d \ln X = \left(\frac{a}{X} - \frac{1}{2} \frac{b^2}{X^2}\right) dt + \frac{b}{X} dz$$ (2.2.5)

Ito-integrating both sides, we get

$$\ln (X_t) - \ln (X_0) = \left(\mu - \frac{1}{2} \sigma^2\right) (t - 0) + \sigma dz$$ (2.2.6)

or finally, using the fact that $dz = \epsilon(t) \sqrt{\Delta t}$ for a Wiener process.

$$X_t = X_0 e^{(\mu - \frac{1}{2} \sigma^2) \Delta t + \sigma \epsilon(t) \sqrt{\Delta t}}, \quad \epsilon(t) \sim N(0,1)$$ (2.2.7)

### 2.2.2 Ornstein-Uhlenbeck Process

The diffusion $X_t$ governed by following stochastic differential equation,

$$dX_t = \gamma (\theta - X_t)dt + \sigma dz$$ (2.2.8)

where $dz = \epsilon(t) \sqrt{\Delta t}$ is a Wiener process and $\epsilon(t)$ is a standardized normal random variable, is known as Ornstein-Uhlenbeck process. The parameters of this diffusion are $\gamma \geq 0$ which is the rate of mean reversion, $\theta$ which is the long term mean of the process and $\sigma$ is the volatility or average magnitude, per square-root time, of the random fluctuations that are modelled as Brownian motions. The Vasicek one-factor equilibrium model of interest rates described in [24] is an example which involves Ornstein-Uhlenbeck process.

Ornstein Uhlenbeck process has an important property called mean reversion property. We can see that $X_t$ has an overall drift towards the mean value $\theta$, if we ignore the random fluctuations in the process due to $dz$. The rate of reversion of diffusion $X_t$ to mean $\theta$ is at an exponential rate $\gamma$, and magnitude directly proportional to the distance between the current value of $X_t$ and $\theta$. 

8
2 Options and Valuation Techniques

We can also observe that for any fixed \( T \), the random variable \( X_T \), conditional upon \( X_0 \), is normally distributed with

\[
\begin{align*}
\text{mean} &= \theta + (X_0 - \theta)e^{-\gamma T} \\
\text{variance} &= \frac{\sigma^2}{2\gamma}(1 - e^{-2\gamma T})
\end{align*}
\]

(2.2.9)

We will now derive an analytical formula for \( X_T \), the price of underlying asset at time \( T \), as we did for geometric Brownian motion, when we are given \( X_0 \) and other parameters of Ornstein-Uhlenbeck process. Introducing the change of variable \( Y_t = X_t - \theta \), then \( Y_t \) satisfies the equation

\[
dY_t = dX_t = -\gamma Y_t dt + \sigma dz
\]

(2.2.10)

In (2.2.8), \( Y_t \) shows a drift towards the value zero, at exponential rate \( \gamma \), hence we introduce the change of variable

\[
Y_t = e^{-\gamma t}Z_t \leftrightarrow Z_t = e^{\gamma t}Y_t
\]

(2.2.11)

which eliminates the drift. Now applying the product rule for Ito Integrals we obtain

\[
\begin{align*}
dZ_t &= \gamma e^{\gamma t}Y_t dt + e^{\gamma t}dY_t \\
&= \gamma e^{\gamma t}Y_t dt + e^{\gamma t}(-\gamma Y_t dt + \sigma dz) \\
&= 0 dt + \sigma e^{\gamma t}dz
\end{align*}
\]

(2.2.12)

Ito-Integrating both sides from 0 to \( t \) we get

\[
Z_t = Z_0 + \sigma \int_0^t e^{\gamma z}dz
\]

(2.2.13)

Next step would be to reverse the change of variables to obtain

\[
Y_t = e^{-\gamma t}Z_t \\
= e^{-\gamma(t-0)}Z_0 + \sigma e^{-\gamma t} \int_0^t e^{\gamma z}dz
\]

(2.2.14)

\[
= e^{-\gamma t}Y_0 + \sigma \int_0^t e^{-\gamma(t-z)}dz
\]

thus we obtain original variable \( X_t \) as

\[
X_t = \theta + Y_t \\
= \theta + e^{-\gamma t}(X_0 - \theta) + \sigma \int_0^t e^{-\gamma(t-z)}dz
\]

(2.2.15)

which can further be simplified to

\[
X_t \sim \theta + e^{-\gamma t}(X_0 - \theta) + \sigma \sqrt{\frac{1 - e^{-2\gamma t}}{2\gamma}} \epsilon(t), \quad \epsilon(t) \sim N(0,1)
\]

(2.2.16)

### 2.2.3 Cox Ingersoll Ross process

In the Vasicek model which involves Ornstein-Uhlenbeck process (2.2.8), the short term interest rate \( X_t \), can become negative. To overcome this drawback, an alternative one-factor equilibrium model was proposed by Cox, Ingersoll and Ross in [15] for short term or instantaneous interest rate in which rates would always remain positive.
Specifically, a diffusion (short-term or instantaneous interest rate in Cox-Ingersoll-Ross interest rate model) which satisfies the following stochastic differential equation

\[ dX_t = \gamma(\theta - X_t)dt + \sigma\sqrt{X_t}dz \]  

(2.2.17)

where \( dz = \epsilon(t)\sqrt{dt} \) is a Wiener process and \( \epsilon(t) \) is a standardized normal random variable, is known as Cox-Ingersoll-Ross process. The parameters of this diffusion are same as the ones in Ornstein-Uhlenbeck process except the fact that the standard deviation factor, \( \sigma\sqrt{X_t} \) does not allow the interest rate \( X_t \) to become negative. Hence we observe that, when the value of \( X_t \) is low, the standard deviation becomes close to zero, in turn cancelling the effect of the random shock on the interest rate. We also note that \( \gamma \theta \geq \frac{1}{2}\sigma^2 \) is necessary and sufficient for non explosiveness of solution of (2.2.17), or in other words, for instantaneous rate to assume the value zero at infinite time with probability 1.

2.3 Monte Carlo Methods

Considering valuation of an option as computing its expected value, we can use repeated random sampling in order to compute this expected value. Monte Carlo methods are a class of algorithms which could be used for such repeated random sampling and thus calculating the expected value of the derivative price. We describe Monte Carlo algorithms for pricing European options in section 2.3.1, Asian Options in section 2.3.2 and (down-and-out and up-and-out) Barrier options in section 2.3.3. Also note that, although the Monte Carlo methods in this section are given for geometric Brownian motion only, they can be adapted very easily for Ornstein-Uhlenbeck and Cox-Ingersoll-Ross diffusions.

2.3.1 European Options

In this section we will give Monte Carlo algorithm for valuation of European put and call options. Pseudo random number generators can be used to compute the estimates of expected values. Consider an European style option with payoff which is some function \( F \) of asset price \( X_t \). Now we get the model of asset price to be described by geometric Brownian motion (2.2.2), and using the risk neutrality approach, option value can be found by calculating \( e^{-rT}F(X_T) \), or in other words, we intend to calculate

\[ e^{-rT}F(X_0e^{(\mu-\frac{1}{2}\sigma^2)dt+\sigma\epsilon(t)\sqrt{dt})} \]  

(2.3.1)

where \( \epsilon(t) \sim N(0,1) \).

The resulting Monte Carlo algorithm can be described as below
### Algorithm 1: Monte Carlo method for valuation of European options

**Data:** Initial stock price $X_0$, strike price $K$, risk-free rate $r$, expiration time $T$, volatility $\sigma$, sample size $M$, path size $N$

**Result:** Call and put value of European option

```plaintext
begin
    // Step 0: Initializations
    Set callValue = 0
    Set putValue = 0

    // Step 1: Compute samples
    for $i = 1$ to $M$ do
        Compute an $N(0,1)$ sample $\epsilon_i$
        Set $X_i = X_0 e^{(\mu - \frac{1}{2} \sigma^2)dt + \sigma \sqrt{dt} \epsilon_i}$
        Set $V_i = e^{-rT} F(X_i)$

    // Step 2, Calculate option value
    Set $V = \frac{1}{M} \sum_{i=1}^{M} V_i$
    if $V \geq K$ then
        Set callValue = $V - K$
    else
        Set putValue = $K - V$

    // Step 3, Calculate approximately 95% confidence intervals
    Set $B = \frac{1}{M-1} \sum_{i=1}^{M} (V_i - V)^2$
    Set Confidence Interval = $\left[ V - \frac{1.96B}{\sqrt{M}}, V + \frac{1.96B}{\sqrt{M}} \right]$
end
```

---

### 2.3.2 Asian Options

In the previous algorithm, we were interested in option price $X_T$ at time $T$, whereas for Asian options, we need to calculate the average of all the values taken by the option price between time 0 and $T$. This is evident from the Monte Carlo algorithm which we have described below for pricing an Asian option (of type average strike).
2.3 Monte Carlo Methods

Algorithm 2: Monte Carlo method for valuation of Asian option

**Data:** Initial stock price $X_0$, strike price $K$, risk-free rate $r$, expiration time $T$, volatility $\sigma$, sample size $M$, path size $N$

**Result:** Call and put values of Asian option

```plaintext
begin
    // Step 0: Initializations
    Set callValue = 0
    Set putValue = 0
    Set $\Delta T = \frac{T}{N-1}$
    Set $XInc = X_0 * \text{Array}(M)$
    Set $XPathArray = X_0 * \text{Array}(M)$
    // Step 1: Compute samples
    for $i = 1$ to $N - 1$ do
        for $j = 1$ to $M$ do
            Compute an $N(0,1)$ sample $\epsilon_j$
            Set $XInc_j = XInc_j e^{(\mu - \frac{1}{2} \sigma^2)\Delta T + \sigma \sqrt{\Delta T} \epsilon_j}$
            Set $XPathArray_j = XPathArray_j + XInc_j$
        for $i = 1$ to $M$ do
            Set $X_{i mean} = \frac{XPathArray_i}{N}$
            Set $V_i = e^{-rT} \max(K - X_{i mean}, 0)$
        // Step 2, Calculate option value
        Set $V = \frac{1}{M} \sum_{i=1}^{M} V_i$
        if $V \geq K$ then
            Set callValue = $V - K$
        else
            Set putValue = $K - V$
        // Step 3, Calculate approximately 95 % confidence intervals
        Set $B^2 = \frac{1}{M-1} \sum_{i=1}^{M} (V_i - V)^2$
        Set ConfidenceInterval = $\left[ V - \frac{1.96B}{\sqrt{M}}, V + \frac{1.96B}{\sqrt{M}} \right]$ end
```

2.3.3 Barrier Options

In the Monte Carlo algorithm 3 given below, we describe the method for pricing put and call values for down-and-out Barrier options. Please note that the below mentioned algorithm is an modified and more efficient version of algorithm taken from [6].

The above algorithm can easily adapted to price any kind of put and call Barrier options, from the eight categories mentioned in section 2.1.3. As an example, the Monte Carlo algorithm 4 for double Barrier options is also given below:
Algorithm 3: Monte Carlo method for valuation of down-and-out Barrier option

**Data:** Initial stock price $X_0$, strike price $K$, lower barrier $H_L$, risk-free rate $r$, expiration time $T$, volatility $\sigma$

**Result:** Put and call values of down-and-out Barrier option

begin
  // Step 0: Initializations
  Set callValue = 0
  Set putValue = 0
  Set $\Delta T = \frac{T}{N-1}$
  Set $XInc = X_0 * Array(M)$
  Set $XPathArray = X_0 * Array(M)$
  Set BarrierBrokenFlag = 0 * Array(M)
  // Step 1: Compute samples
  for $i = 1$ to $N-1$ do
    for $j = 1$ to $M$ do
      if $\text{BarrierBrokenFlag}_j \neq 1$ then
        Compute an $N(0,1)$ sample $\epsilon_j$
        Set $XInc_j = XInc_j e^{(\mu - \frac{1}{2} \sigma^2)dt + \sigma \sqrt{\Delta T} \epsilon_j}$
        if $XInc_j \leq H_L$ then
          Set $\text{BarrierBrokenFlag}_j = 1$
          Set $XPathArray_j = 0$
        else
          Set $XPathArray_j = XPathArray_j + XInc_j$
      end
    end
  for $i = 1$ to $M$ do
    Set $X_{i, \text{mean}} = \frac{XPathArray_i}{N}$
    Set $V_i = e^{-rT} \max(K - X_{i, \text{mean}}, 0)$
  // Step 2, Calculate option value
  Set $V = \frac{1}{M} \sum_{i=1}^{M} V_i$
  if $V \geq K$ then
    Set callValue = $V - K$
  else
    Set putValue = $K - V$
  // Step 3, Calculate approximately 95% confidence intervals
  Set $B^2 = \frac{1}{M-1} \sum_{i=1}^{M} (V_i - V)^2$
  Set ConfidenceInterval = $[V - \frac{1.96B}{\sqrt{M}}, V + \frac{1.96B}{\sqrt{M}}]$
end
Algorithm 4: Monte Carlo method for valuation of double Barrier options

Data: Initial stock price $X_0$, strike price $K$, upper barrier $H_U$, lower barrier $H_L$, risk-free rate $r$, expiration time $T$, volatility $\sigma$

Result: Put and call values of double Barrier Option

begin
    // Step 0: Initializations
    Set callValue = 0
    Set putValue = 0
    Set $\Delta T = \frac{T}{N-1}$
    Set $X_{Inc} = X_0 \ast \text{Array}(M)$
    Set $X_{PathArray} = X_0 \ast \text{Array}(M)$
    Set $\text{BarrierBrokenFlag} = 0 \ast \text{Array}(M)$

    // Step 1: Compute samples
    for $i = 1$ to $N-1$ do
        for $j = 1$ to $M$ do
            if $\text{BarrierBrokenFlag}_j \neq 1$ then
                Compute an N(0,1) sample $\epsilon_j$
                Set $X_{Inc}_j = X_{Inc}_j e^{(\mu - \frac{1}{2} \sigma^2) \Delta T + \sigma \sqrt{\Delta T} \epsilon_j}$
                if $X_{Inc}_j \leq H_L$ OR $X_{Inc}_j \geq H_U$ then
                    Set $\text{BarrierBrokenFlag}_j = 1$
                    Set $X_{PathArray}_j = 0$
                else
                    Set $X_{PathArray}_j = X_{PathArray}_j + X_{Inc}_j$
            end
        end
    end

    for $i = 1$ to $M$ do
        Set $X_{mean}_i = \frac{X_{PathArray}_i}{N}$
        Set $V_i = e^{-rT} \max(K - X_{mean}_i, 0)$
    end

    // Step 2, Calculate option value
    Set $V = \frac{1}{M} \sum_{i=1}^{M} V_i$
    if $V \geq K$ then
        Set callValue = $V - K$
    else
        Set putValue = $K - V$
    end

    // Step 3, Calculate approximately 95% confidence intervals
    Set $B^2 = \frac{1}{M-1} \sum_{i=1}^{M} (V_i - V)^2$
    Set ConfidenceInterval = $[V - \frac{1.96B}{\sqrt{M}}, V + \frac{1.96B}{\sqrt{M}}]$
2.3.4 Advantages and Disadvantages of Monte Carlo Methods

Even though Monte Carlo algorithms are simple and elegant to implement, they have certain merits and demerits of their own, as a class of algorithms which are enlisted below:

Advantages

• The major advantage of Monte Carlo methods is that they are very simple to implement and adapt. Usually an expression of solution or its analytical properties are not required for performing Monte Carlo simulations.

• Owing to their genericity and simplicity in implementation, Monte Carlo methods offer unrestricted choice of functions for implementation.

• As Monte Carlo simulations intrinsically involve certain type of repetitive tasks, they can be easily parallelized.

Disadvantages

• The most important drawback of Monte Carlo algorithms is that they are slow when the dimension of integration is low. We can prove that for Monte Carlo integration the probabilistic error bound is inversely proportional to the square root of the number of samples, hence in order to achieve one more decimal digit of precision, we would require 100 times more the number of samples.

• The probabilistic error bound is also dependent on variance. Hence it is not a good representation of the error when the underlying probability distributions are skewed.

• The error in values computed through Monte Carlo algorithms also depend on the underlying probability distributions. As we can consider that Monte Carlo methods try to estimate the expectation of certain values by sampling, this estimated value may be erroneous, if the underlying probability distribution is heavily skewed or has heavier-than-normal tails. This disadvantage can be taken care of by using appropriate non-random numerical methods.

• As Monte-Carlo methods typically follow a Black-box approach, it is difficult to perform sensitivity analysis for the input parameters, which would be otherwise possible with certain types of analytical approximation techniques.

Effects of several of above disadvantages can be decreased there by increasing the accuracy of results in Monte Carlo simulations. For example, there are several techniques for reducing variance in the sampled values, such as use of antithetic and control variates, importance sampling, etc. However, as these topics are not in the scope of this project, we will not elaborate on them. Interested reader is encouraged to refer [25] for comprehensive treatment of Monte Carlo methods.

2.4 Black Scholes PDE and Analytical Solution

A mathematical model of market for equity was given by Black-Scholes, where the underlying is a stochastic process and the price of derivative on the underlying satisfies a partial differential equation called Black-Scholes PDE. We briefly summarize the Black Scholes model, PDE and solution formula obtained by solving the Black-Scholes PDE for European put and call options below. Suppose that underlying $X_t$ is governed by geometric Brownian motion (2.2.2):

$$dX_t = \mu X_t dt + \sigma X_t dz$$  \hspace{1cm} (2.4.1)
2.4 Black Scholes PDE and Analytical Solution

where $dz = \epsilon(t)\sqrt{dt}$ is a Wiener process and $\epsilon(t)$ is a standardized normal random variable.

Now let $V$ be some sort of option on $X$, or in other words, consider $V$ as a function of $X$ and $t$. Hence, $V(X, t)$ is the value of the option at time $t$ if the price of the underlying stock at time $t$ is $X$. We also assume that $V(X, t)$ is differentiable twice with respect to $X$ and once with respect to $t$. Then $V(X, t)$ must satisfy the Black-Scholes partial differential equation:

$$\frac{\partial V}{\partial t} + rX \frac{\partial V}{\partial X} + \sigma^2X^2 \frac{1}{2} \frac{\partial^2 V}{\partial X^2} = rV$$

(2.4.2)

where $r$ is the interest rate.

2.4.1 Analytical Solution for European Options

By solving (2.4.2) for price of European call and put options (for solution techniques and more details please refer [14], [6] and [26]), we get

$$C_t = X_t \Phi(d_1) - Ke^{-rT} \Phi(d_2)$$

$$P_t = Ke^{-rT} \Phi(d_2) - X_t \Phi(d_1)$$

(2.4.3)

where

$$d_1 = \ln\left(\frac{X_t}{K}\right) + T \left(r + \frac{\sigma^2}{2}\right)$$

$$d_2 = \ln\left(\frac{X_t}{K}\right) + T \left(r - \frac{\sigma^2}{2}\right)$$

$$= d_1 - \sigma \sqrt{T}$$

(2.4.4)

$$\Phi(d_n) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{d_n} e^{-\frac{x^2}{2}} dx$$

or in other words, $\Phi(.)$ is the standard normal cumulative distribution function.

2.4.2 Analytical Solution for Barrier Options

We present here the analytical formulae used in this project for comparison with bounds obtained by following the numerical approach given by Lasserre, Prieto-Rumeau and Zervos in [21]. By solving (2.4.2) for price of different types of Barrier call and put options (for solution techniques please refer [27]), we get the following solutions:

**Down-and-out and down-and-in call options:**

When $H \leq K$, we get

$$C^{di}_t = X_0 \left(\frac{H}{X_0}\right)^{2e_1} \Phi(e_2) - Ke^{-rT} \left(\frac{H}{X_0}\right)^{2e_1-2} \Phi(e_2 - \sigma \sqrt{T})$$

(2.4.5)
where

\[ e_1 = \frac{r + \sigma^2}{\sigma^2} \]
\[ e_2 = \ln \left( \frac{H^2}{X_0 K} \right) + e_1 \sqrt{T} \]
\[ f_1 = \frac{\ln \left( \frac{X_0}{H} \right)}{\sigma \sqrt{T}} + e_1 \sqrt{T} \]
\[ f_2 = \frac{\ln \left( \frac{H}{X_0} \right)}{\sigma \sqrt{T}} + e_1 \sqrt{T} \]

(2.4.6)

We also quote the known fact, that the sum of any type of out and in calls is equal to value of regular call given in (2.4.3). Hence the value of down-and-out Barrier call option is given by

\[ C_{do} = C_t - C_{di} \]  
\[ (2.4.7) \]

On the other hand if \( H \geq K \), we have

\[
C_{do} = -X_0 \Phi(f_1) - Ke^{-rT} \Phi(f_1 - \sigma \sqrt{T})
- X_0 \left( \frac{H}{X_0} \right)^{2e_1} \Phi(-e_2 - f_2)
+ Ke^{-rT} \left( \frac{H}{X_0} \right)^{2e_1-2} \Phi(f_2 - \sigma \sqrt{T})
\]

(2.4.8)

and

\[ C_{di} = C_t - C_{do} \]  
\[ (2.4.9) \]

**Up-and-out and up-and-in call options:**

When \( H \geq K \)

\[
C_{ui} = X_0 \Phi(f_1) - Ke^{-rT} \Phi(f_1 - \sigma \sqrt{T})
- X_0 \left( \frac{H}{X_0} \right)^{2e_1} \left[ \Phi(-e_2) - \Phi(-f_2) \right]
+ Ke^{-rT} \left( \frac{H}{X_0} \right)^{2e_1-2} \left[ \Phi(-e_2 + \sigma \sqrt{T}) - \Phi(f_2 + \sigma \sqrt{T}) \right]
\]

(2.4.10)

and

\[ C_{oi} = C_t - C_{ui} \]
\[ (2.4.11) \]

where as when \( H \leq K \),

\[ C_{ou} = 0 \]
\[ (2.4.12) \]

and

\[ C_{oi} = C_t \]
\[ (2.4.13) \]

**Up-and-out and up-and-in put options:**

When \( H \leq K \)

\[
P_{ou} = -X_0 \left( \frac{H}{X_0} \right)^{2e_1} \Phi(-e_2) + Ke^{-rT} \left( \frac{H}{X_0} \right)^{2e_1-2} \Phi(-e_2 + \sigma \sqrt{T})
\]

(2.4.14)
and
\[ P_t^{uo} = P_t - P_t^{ui} \]  
(2.4.15)
where as when \( H \leq K \),
\[
    P_t^{uo} = -X_0\Phi(f_1) + Ke^{-rT}\Phi(-f_1 + \sigma\sqrt{T}) \\
    + X_0 \left( \frac{H}{X_0} \right)^{2e_1} \Phi(-f_2) \\
    - Ke^{-rT} \left( \frac{H}{X_0} \right)^{2e_1-2} \Phi(-f_2 + \sigma\sqrt{T}) 
\]  
(2.4.16)
and
\[
    P_t^{ui} = P_t - P_t^{ui} 
\]  
(2.4.17)

**Down-and-out and Down-and-in put options:**
When \( H \geq K \)
\[
    P_t^{di} = -X_0\Phi(-f_1) + Ke^{-rT}\Phi(-f_1 + \sigma\sqrt{T}) \\
    + X_0 \left( \frac{H}{X_0} \right)^{2e_1} \left[ \Phi(e_2) - \Phi(f_2) \right] \\
    - Ke^{-rT} \left( \frac{H}{X_0} \right)^{2e_1-2} \left[ \Phi(e_2 - \sigma\sqrt{T}) - \Phi(f_1 - \sigma\sqrt{T}) \right] 
\]  
(2.4.18)
and
\[
    P_t^{do} = P_t - P_t^{di} 
\]  
(2.4.19)
where as when \( H \leq K \),
\[
    P_t^{do} = 0 
\]  
(2.4.20)
and
\[
    P_t^{di} = P_t 
\]  
(2.4.21)

### 2.4.3 Assumptions and Shortcomings of Black Scholes Formula

There are several assumptions under which above formulation and results were obtained, which are mentioned as below (for more details, please refer [17], [14], [5] and [6]):

- The price of the underlying asset follows a Wiener process, and the price changes are log-normally distributed.
- The stock does not pay a dividend and there are no transaction costs or taxes.
- It is possible to short sell the underlying stock and trading in the stock is continuous. Sudden and large changes in stock prices do not occur.
- It is possible to borrow and lend cash at a risk-free interest rate which remains constant over time. Also, there are no arbitrage opportunities.
- The volatility of stock price also remains constant over time.

Some of the assumptions above can be changed, and corresponding Black-Scholes formula can be derived (for example, for extension of Black-Scholes model to price options on dividend paying stocks, please refer to chapter 13 in [14]). On the other hand, certain assumptions cannot be corrected and often lead to an error in the computed option value. For example, it is a well known fact that probability distribution of underlying asset is not lognormal which leads to an
incorrect bias in pricing done by Black-Scholes method. The most prevalent assumption, for example, when underlying asset is equity, is that equity price has a heavier left tail and less heavier right tail in comparison with lognormal model.

Many alternative models have been proposed for overcoming the above disadvantages. In the next two sections, we give two such methodologies for pricing simple and exotic options.
Chapter 3

Semi-Parametric Approach for Bounding Option Price

In their original article [1], Bertsimas and Popescu addressed the problem of deriving optimal inequalities for $E[\phi(X)]$ for a random variable $X$ when given collection of general moments $E[f_i(X)] = q_i$. Utilizing semidefinite and convex optimization methods in order to derive optimal bounds on the probability that variable $X$ belongs in a given set, given first few moments for the random variable $X$, they substantiate their theory by finding optimal upper bounds for option prices with general payoff. The advantage of their approach is that probability distribution of underlying asset price is not required, only the moments are sufficient for deriving the optimal bounds. Later on, Gotoh and Konno [9] extended the above approach to obtain tight lower bound for the option price with general payoff, given its first few moments.

In this section, we investigate and elaborate the semi-parametric approach for pricing options originally proposed by Lo in [22] and extended by Bertsimas and Popescu in [2] and [1] and modified by Gotoh and Konno in [9] for obtaining tighter upper and lower bounds on European type call option price, when we are given the first $n$ moments of the underlying security price. The main idea in this approach is to view a semidefinite programming problem as a linear program with an infinite number of linear constraints.

3.1 Review of Basic Results

Introducing the notation: $T$ and $K$ are exercise time and exercise price of European type call option, respectively. $S_t$ and $r$ are stock price at time $t$, and risk-free rate, respectively. Also $\pi$ is the risk-neutral probability of $S_T$ and $E_\pi[.]$ denotes the expected value of random variable under the probability measure $\pi$. Then the call option price $C_t$ is given by

$$C_t = e^{-r(T-t)}E_\pi[\max(0, S_T - K)] \tag{3.1.1}$$

If we are given the first $n$ moments $q_i, i = 1, ..., n$ of $\pi$, then the upper and lower bounds of the call option price can be obtained by solving the following linear programming problems with infinite number of variables $\pi(S_T)$:

$$\text{maximize } e^{-rt} \int_0^\infty \max(0, S_T - K) \times \pi(S_T) dS_T$$

s.t. $e^{-rT} \int_0^\infty S_T^j \pi(S_T) dS_T = q_j$, $j = 0, 1, ..., n, \pi(S_T) \geq 0. \tag{3.1.2}$
minimize \( e^{-r\tau} \int_{0}^{\infty} \max(0, S_T - K) \times \pi(S_T)dS_T \) \[ (3.1.3) \]

s.t. \( e^{-r\tau} \int_{0}^{\infty} S_T^2 \pi(S_T)dS_T = q_j, \quad j = 0, 1, ..., n, \quad \pi(S_T) \geq 0. \)

where \( q_0 = 1 \) and \( \tau = T - t. \) Also to be noted is the fact that above problems are feasible. We now intend to derive the dual problems of (3.1.2) and (3.1.3). Denoting \( x = e^{-r\tau}S_T \) and dual variables as \( y_i, i = 0, 1, ..., n, \) we can derive their dual problems:

minimize \( \pi(S_T) \sum_{i=0}^{n} q_i y_i \) \[ (3.1.4) \]

s.t. \( \sum_{i=0}^{n} x^iy_i \geq \max(0, x - e^{-r\tau}K), \quad \forall x \in \mathbb{R}_1^+. \)

maximize \( \pi(S_T) \sum_{i=0}^{n} q_i y_i \) \[ (3.1.5) \]

s.t. \( \sum_{i=0}^{n} x^iy_i \leq \max(0, x - e^{-r\tau}K), \quad \forall x \in \mathbb{R}_1^+. \)

We will now reiterate the propositions given by Bertsimas and Popescu which will assist us to convert (3.1.4) and (3.1.5) into semidefinite programming problems.

**Proposition 1.** The polynomial \( g(x) = \sum_{r=0}^{2k} y_r x^r \) satisfies \( g(x) \geq 0 \) for all \( x \in \mathbb{R}_1^+ \) if and only if there exists a positive semidefinite matrix \( X = [x_{ij}] \), \( i, j = 0, ..., k, \) such that

\[
y_r = \sum_{i,j:i+j=r} x_{ij}, \quad r = 0, ..., 2k, \quad X \succeq 0. \]

**Proof.** We will first prove the if part, i.e. we assume that (3.1.6) holds, and we need to prove that \( g(x) = \sum_{r=0}^{2n} y_r x^r \) satisfies \( g(x) \geq 0 \) for all \( x \in \mathbb{R}_1^+ \). Let \( x_k' = (1, x, x^2, ..., x^k) \). Then

\[
g(x) = \sum_{r=0}^{2k} \sum_{i+j=r} x_{ij} x^r
= \sum_{i=0}^{k} \sum_{j=0}^{k} x_{ij} x^i x^j
= x_k' X x_k
\geq 0
\]

and last step is obtained because we are given that \( X \) is positive semidefinite, i.e. \( X \succeq 0. \) Now we will prove the only-if part. We assume that the polynomial \( g(x) \geq 0, \quad \forall x \in \mathbb{R}_1^+ \) and degree of \( g(x) \) is \( 2k. \) Then, we observe the fact that real roots of \( g(x) \) should have even multiplicity. This is because if they have odd multiplicity then \( g(x) \) would alter sign in an neighborhood of a root. Denoting \( \lambda_i, i = 1, ..., r \) as the real roots of \( g(x) \) and \( 2m_i \) as its corresponding multiplicity, its complex roots can then be arranged in conjugate pairs, \( a_j + ib_j, a_j - ib_j, j = 1, ..., h. \) Hence

\[
g(x) = y_{2k} \prod_{i=1}^{r} (x - \lambda_i)^{2m_i} \prod_{j=1}^{h} ((x - a_j)^2 + b_j^2).
\]
We note the fact that the leading coefficient \( y_{2k} \) needs to be positive. Thus, by expanding the terms in the products, we can see that \( g(x) \) can be written as a sum of squares of polynomials, of the form

\[
g(x) = \sum_{i=0}^{k} \left( \sum_{j=0}^{k} x_{ij} x^j \right)^2
= x_k^t X x_k
\]

which proves that \( X \) is positive semidefinite, i.e. \( X \succeq 0 \), as we assumed that \( g(x) \geq 0 \).

**Proposition 2.** The polynomial \( g(x) = \sum_{r=0}^{k} y_r x^r \) satisfies \( g(x) \geq 0 \) for all \( x \in \mathbb{R}^1 \) if and only if there exists a positive semidefinite matrix \( X = [x_{ij}] \). \( i, j = 0, ..., k \), such that

\[
\sum_{i,j: i+j=2l-1} x_{ij} = 0, \ l = 1, ..., k, \\
\sum_{i,j: i+j=2l} x_{ij} = y_l, \ l = 0, ..., k, \quad (3.1.7)
\]

\[X \succeq 0\]

**Proof.** We observe that \( g(x) \geq 0 \) for all \( x \geq 0 \) if and only if \( g(t^2) \geq 0 \), \( \forall t \), which is a direct result of applying Proposition 1. Now in order to prove that \( g(t^2) \geq 0 \), \( \forall t \), we evaluate \( g(t^2) \) as below:

\[
g(t^2) = y_0 t^0 + y_1 t^2 + y_2 t^4 + ... + y_k t^{2k},
= y_0 + 0.1 + y_1 t^2 + 0.1^3 + y_2 t^4 + ... + 0.1^{2k-1} + y_k t^{2k},
\]

from which we observe that all terms can be amalgamated into following two terms:

\[0.1 + 0.1^3 + ... + 0.1^{2k-1}, \]

and

\[y_0 + y_2 t^4 + ... + y_k t^{2k},\]

hence we obtain

\[
g(t^2) = \sum_{r=0}^{k} \sum_{i,j: i+j=2r-1} x_{ij} t^{2r-1} + \sum_{r=0}^{k} \sum_{i,j: i+j=2r} x_{ij} t^{2r}
\]

Now finally we see that \( g(t^2) \geq 0 \), \( \forall t \) if and only if

\[
\sum_{r=0}^{k} \sum_{i,j: i+j=2r-1} x_{ij} t^{2r-1} + \sum_{r=0}^{k} \sum_{i,j: i+j=2r} x_{ij} t^{2r} \geq 0
\]

\[\Leftrightarrow \sum_{r=0}^{k} \sum_{i,j: i+j=2r-1} x_{ij} t^{2r-1} \geq 0 \quad \text{and} \quad \sum_{r=0}^{k} \sum_{i,j: i+j=2r} x_{ij} t^{2r} \geq 0
\]

\[\Leftrightarrow \sum_{r=0}^{k} \sum_{i,j: i+j=2r-1} x_{ij} t^{2r-1} = 0 \quad \text{and} \quad \sum_{r=0}^{k} \sum_{i,j: i+j=2r} x_{ij} t^{2r} \geq 0
\]

which proves the proposition, noting the fact that \( t^2 \geq 0 \), \( \forall t \in \mathbb{R}^1 \). \( \Box \)
Proposition 3. The polynomial \( g(x) = \sum_{r=0}^{2k} y_r x^r \) satisfies \( g(x) \geq 0 \) for all \( x \in [0, a] \) if and only if there exists a positive semidefinite matrix \( X = [x_{ij}] \), \( i, j = 0, \ldots, k \), such that

\[
0 = \sum_{i, j: i+j=2l-1} x_{ij}, \quad l = 1, \ldots, k,
\]

\[
\sum_{r=0}^{l} y_r \left( \begin{array}{c}
  k - r \\
  l - r
\end{array} \right) a^r = \sum_{i, j: i+j=2l} x_{ij}, \quad l = 0, \ldots, k,
\]

\[
X \succeq 0 \tag{3.1.8}
\]

Proof. We observe that \( g(x) \geq 0 \) for all \( x \in [0, a] \) if and only if

\[
(1 + t^2)^k g \left( \frac{at^2}{1 + t^2} \right) \geq 0, \quad \forall \ t
\]

Now in order to prove the above fact, we evaluate the following expression:

\[
(1 + t^2)^k g \left( \frac{at^2}{1 + t^2} \right) = \sum_{r=0}^{k} y_r a^r t^{2r} (a + t^2)^{k-r} = (1 + t^2)^k \left[ y_0 + y_2 a^2 t^4 (1 + t^2)^2 + \ldots + y_k a^k t^{2k} (1 + t^2)^k \right]
\]

\[
= \sum_{r=0}^{k} y_r a^r \sum_{l=0}^{k-r} \left( \begin{array}{c}
  k - r \\
  l
\end{array} \right) t^{2(l+r)} = \sum_{j=0}^{k} t^{2j} \left( \sum_{r=0}^{j} y_r a^r \right)
\]

Using the equality conditions on semidefinite matrix \( X \) given in Proposition 3, we get the final expression as

\[
(1 + t^2)^k g \left( \frac{at^2}{1 + t^2} \right) \geq 0, \quad \forall \ t
\]

if and only if

\[
\sum_{j=0}^{k} t^{2j} \left( \sum_{r=0}^{j} y_r \left( \begin{array}{c}
  k - r \\
  j - r
\end{array} \right) a^r \right) \geq 0
\]

\[
\Leftrightarrow \sum_{r=0}^{k} \sum_{i, j: i+j=2r} x_{ij} t^{2r} \geq 0
\]

which we have proved to be true in Proposition 2. \( \square \)

Proposition 4. The polynomial \( g(x) = \sum_{r=0}^{2k} y_r x^r \) satisfies \( g(x) \geq 0 \) for all \( x \in [a, \infty) \) if and only if there exists a positive semidefinite matrix \( X = [x_{ij}] \), \( i, j = 0, \ldots, k \), such that

\[
0 = \sum_{i, j: i+j=2l-1} x_{ij}, \quad l = 1, \ldots, k,
\]

\[
\sum_{r=l}^{k} y_r \left( \begin{array}{c}
  r \\
  l
\end{array} \right) a^{r-l} = \sum_{i, j: i+j=2l} x_{ij}, \quad l = 1, \ldots, k,
\]

\[
X \succeq 0 \tag{3.1.9}
\]
3.1 Review of Basic Results

Proof. We observe that \( g(x) \geq 0, \ \forall x \in [0, a] \) if and only if
\[
g(a + t^2) \geq 0, \ \forall \ t
\]
Now in order to prove that \( g(a + t^2) \geq 0 \ \forall \ t \), we evaluate \( g(a + t^2) \) as below:
\[
g(a + t^2) = \sum_{r=0}^{k} y_r (a + t^2)^r
\]
\[
= \sum_{r=0}^{k} y_r \sum_{l=0}^{r} \binom{r}{l} a^{r-l} t^{2l}
\]
\[
= \sum_{l=0}^{k} t^{2l} \left( \sum_{r=l}^{k} y_r \binom{r}{l} a^{r-l} \right)
\]
Using the equality conditions on semidefinite matrix \( X \) given in Proposition 4, we get the final expression as
\[
g(a + t^2) \geq 0, \ \forall \ t
\]
if and only if
\[
\sum_{l=0}^{k} t^{2l} \left( \sum_{r=l}^{k} y_r \binom{r}{l} a^{r-l} \right) \geq 0
\]
\[
\iff \sum_{r=0}^{k} \sum_{i,j: i+j=2l} x_{ij} t^{2r} \geq 0
\]
which we have proved to be true in Proposition 2. \( \square \)

Proposition 5. The polynomial \( g(x) = \sum_{r=0}^{2k} y_r x^r \) satisfies \( g(x) \geq 0 \) for all \( x \in (-\infty, a] \) if and only if there exists a positive semidefinite matrix \( X = [x_{ij}] \), \( i,j = 0,...,k \), such that
\[
0 = \sum_{i,j: i+j=2l-1} x_{ij}, l = 1,...,k,
\]
\[
(-1)^l \sum_{r=l}^{k} y_r \binom{r}{l} a^{r-l} = \sum_{i,j: i+j=2l} x_{ij}, l = 1,...,k, \tag{3.1.10}
\]
\[
X \succeq 0
\]
Proof. The proof is similar to that given for Proposition 4, but we just replace \( t^2 \) by \( -t^2 \) in \( g(a + t^2) \). We observe that \( g(x) \geq 0, \ \forall x \in [0, a] \) if and only if
\[
g(a - t^2) \geq 0 \ \forall \ t
\]
Now in order to prove that \( g(a - t^2) \geq 0 \ \forall \ t \), we evaluate \( g(a - t^2) \) as below:
\[
g(a - t^2) = \sum_{r=0}^{k} y_r (a - t^2)^r
\]
\[
= \sum_{r=0}^{k} y_r \sum_{l=0}^{r} \binom{r}{l} a^{r-l} (-t^2)^l
\]
\[
= \sum_{l=0}^{k} t^{2l} \left( (-1)^l \sum_{r=l}^{k} y_r \binom{r}{l} a^{r-l} \right)
\]
we obtain (3.1.10) by applying (3.1.6) in a similar fashion to Proposition 4. \( \square \)
Proposition 6. The polynomial \( g(x) = \sum_{r=0}^{2k} y_r x^r \) satisfies \( g(x) \geq 0 \) for all \( x \in [a,b] \) if and only if there exists a positive semidefinite matrix \( X = [x_{ij}] \), \( i,j = 0,\ldots,k \), such that

\[
0 = \sum_{i,j: i+j=2l-1} x_{ij}, \quad l = 1,\ldots,k, \\
\sum_{m=0}^{l} \sum_{r=m}^{k+m-l} y_r \binom{r}{m} a^{r-m} b^m = \sum_{i,j: i+j=2l} x_{ij}, \quad l = 1,\ldots,k, \\
X \succeq 0
\]

(3.1.11)

Proof. We observe that \( g(x) \geq 0, \forall x \in [a,b] \) if and only if

\[
(1 + t^2)^k g \left( a + (b-a) \frac{t^2}{1+t^2} \right) \geq 0, \forall \ t
\]

Since

\[
(1 + t^2)^k g \left( a + (b-a) \frac{t^2}{1+t^2} \right) = \sum_{r=0}^{k} y_r (a + bt^2)^r \frac{t^2}{1+t^2} (1 + t^2)^{k-r}
\]

\[
= \sum_{r=0}^{k} y_r \sum_{m=0}^{r} \binom{r}{m} a^{r-m} b^m t^{2m} \sum_{j=0}^{k-r} \binom{k-r}{j} t^{2j}
\]

\[
= \sum_{l=0}^{k} t^{2l} \left( \sum_{m=0}^{l} \sum_{r=m}^{k+m-l} y_r \binom{r}{m} a^{r-m} b^m \right)
\]

we obtain (3.1.11) by applying (3.1.6). \( \square \)

3.2 SDP Formulation of the Bounding Problems

Now that we have seen the basic results we are equipped to convert problems (3.1.4) and (3.1.5) which have infinite constraints, into equivalent SDP problems.

Theorem 1. An optimal solution of problem (3.1.4) can be obtained by solving the following SDP problem

\[
\begin{align*}
\text{minimize} & \quad \pi(S_T) \sum_{i=0}^{n} q_i y_i \\
\text{s.t.} & \quad \sum_{i,j: i+j=2l-1} x_{ij} = 0, \quad l = 1,\ldots,n \\
& \quad y_l = \sum_{i,j: i+j=2l} x_{ij} = 0, \quad l = 0,\ldots,n \\
& \quad \sum_{i,j: i+j=2l-1} z_{ij} = 0, \quad l = 1,\ldots,n \\
& \quad y_0 - z_{00} = -\bar{K}, \\
& \quad y_1 - \sum_{i,j: i+j=2} z_{ij} = 1 \\
& \quad y_1 - \sum_{i,j: i+j=2} z_{ij} = 0, \quad l = 2,\ldots,n \\
& \quad X, Z \succeq 0
\end{align*}
\]

(3.2.1)

where \( X \succeq 0, Z \succeq 0 \) denote that \( X \) and \( Y \) are positive semidefinite and \( \bar{K} = e^{-r\tau} K \).
3.2 SDP Formulation of the Bounding Problems

**Proof.** We wish to convert the infinite constraints of problem (3.1.4) in finite constraints, thus obtaining a SDP problem. Representing the feasible region of (3.1.4) as

\[ \sum_{i=0}^{n} q_i y_i \geq 0, \ \forall \ x \geq 0 \]  
(3.2.2)

and

\[ (y_0 + K) + (y_1 - 1)x + \sum_{i=2}^{n} q_i y_i \geq 0, \ \forall \ x \geq 0 \]  
(3.2.3)

Using Proposition 2, we conclude that (3.2.2) is true if and only if there exists a positive semidefinite matrix \( X = [x_{ij}] \), \( i, j = 0, ..., n \), such that

\[
\sum_{i,j: i+j=2l-1} x_{ij} = 0, \ l = 1, ..., n, \\
\sum_{i,j: i+j=2l} x_{ij} = y_l, \ l = 0, ..., n, \ \\
X \succeq 0
\]  
(3.2.4)

Using Proposition 2 again for (3.2.3) we conclude that (3.2.3) is true if and only if there exists a positive semidefinite matrix \( Z = [z_{ij}] \), \( i, j = 0, ..., n \), such that

\[
\sum_{i,j: i+j=2l-1} z_{ij} = 0, \ l = 1, ..., n, \\
\sum_{i,j: i+j=2l} z_{ij} - K = y_l, \ l = 0, \\
\sum_{i,j: i+j=2l} z_{ij} + 1 = y_l, \ l = 1, \\
\sum_{i,j: i+j=2l} z_{ij} = y_l, \ l = 2, ..., n, \\
X \succeq 0
\]  
(3.2.5)

Hence we obtain the constraints for our SDP problem by simplifying (3.2.4) and (3.2.5).

Now we will present another way of simplifying the constraints in (3.1.4), for obtaining a SDP problem.

**Theorem 2.** An optimal solution of problem (3.1.4) can be obtained by solving the following...
**SDP problem**

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=0}^{n} q_i y_i \\
\text{s.t.} & \quad \sum_{i,j:i+j=2l-1} x_{ij} = 0, \ l = 1, \ldots, n \\
& \quad \sum_{h=0}^{l} \left( \binom{n-h}{l-h} \bar{K}^h \right) = \sum_{i,j:i+j=2l} x_{ij}, \ l = 0, \ldots, n \\
& \quad \sum_{i,j:i+j=2l-1} z_{ij} = 0, \ l = 1, \ldots, n \\
& \quad \sum_{h=0}^{n} \bar{K}^h y_h = z_{00}, \\
& \quad \sum_{h=1}^{n} h \bar{K}^{h-1} y_h - 1 = \sum_{i,j:i+j=2} z_{ij}, \\
& \quad \sum_{h=l}^{n} \left( \binom{h}{l} \bar{K}^{h-l} y_h \right) = \sum_{i,j:i+j=2l} z_{ij}, \ l = 2, \ldots, n \\
& \quad X, Z \succeq 0
\end{align*}
\]  

(3.2.6)

**Proof.** If we partition \( x \in [0, +\infty) \) in two parts, i.e. \( x \in [0, \bar{K}) \) and \( x \in [\bar{K}, +\infty) \). Using this fact we note that the feasible region of (3.1.4) can be represented as

\[
\sum_{i=0}^{n} q_i y_i \geq 0, \ \forall \ x \in [0, \bar{K}) \tag{3.2.7}
\]

and

\[
(y_0 + \bar{K}) + (y_1 - 1)x + \sum_{i=2}^{n} q_i y_i \geq 0, \ \forall \ x \in [\bar{K}, +\infty) \tag{3.2.8}
\]

Using Proposition 3, we conclude that (3.2.7) is true if and only if there exists a positive semidefinite matrix \( X = [x_{ij}], \ i, j = 0, \ldots, n, \) such that

\[
0 = \sum_{i,j: i+j=2l-1} x_{ij}, \ l = 1, \ldots, n,
\]

\[
\sum_{r=0}^{l} y_r \left( \binom{n-r}{l-r} \bar{K}^r \right) = \sum_{i,j: i+j=2l} x_{ij}, \ l = 0, \ldots, n,
\]

(3.2.9)

Using Proposition 4, we conclude that (3.2.8) is true if and only if there exists a positive semidef-
inite matrix $Z = [z_{ij}], \ i, j = 0, ..., n$, such that

$$
\begin{align*}
\sum_{i, j : i + j = 2l - 1} z_{ij} &= 0, \ l = 1, ..., n, \\
\sum_{h=l}^{n} y_h \left( \frac{h}{l} \right) \bar{K}^{h-l} &= \sum_{i, j : i + j = 2l} z_{ij}, \ l = 0, \\
\sum_{h=l}^{n} y_h \left( \frac{h}{l} \right) \bar{K}^{h-l} - 1 &= \sum_{i, j : i + j = 2l} z_{ij}, \ l = 1, \\
\sum_{h=l}^{n} y_h \left( \frac{h}{l} \right) \bar{K}^{h-l} &= \sum_{i, j : i + j = 2l} z_{ij}, \ l = 2, ..., n
\end{align*}
$$

(3.2.10)

Hence we obtain the constraints for our SDP problem by simplifying (3.2.9) and (3.2.10).

We now turn our attention to convert problem (3.1.5) which has infinite constraints, into equivalent SDP problem.

**Theorem 3.** An optimal solution of problem (3.1.5) can be obtained by solving the following SDP problem

$$
\begin{align*}
\text{maximize} & \quad \pi \left( S^T \right) \\
\text{subject to} & \quad \sum_{i=0}^{n} q_i y_i \geq 0, \forall x \in [0, \bar{K}) \\
& \quad \sum_{h=0}^{l} y_h \left( \frac{n - h}{l - h} \right) \bar{K}^{h} = \sum_{i, j : i + j = 2l} x_{ij}, \ l = 0, ..., n \\
& \quad \sum_{i, j : i + j = 2l - 1} z_{ij} = 0, \ l = 1, ..., n \\
& \quad \sum_{h=0}^{n} \bar{K}^{h} y_h + z_{00} = 0, \\
& \quad \sum_{h=1}^{n} h \bar{K}^{h-1} y_h - 1 = \sum_{i, j : i + j = 2} z_{ij}, \\
& \quad \sum_{h=l}^{n} \left( \frac{h}{l} \right) \bar{K}^{h-l} y_h = \sum_{i, j : i + j = 2l} z_{ij}, \ l = 2, ..., n \\
& \quad X, Z \succeq 0
\end{align*}
$$

(3.2.11)

**Proof.** We will follow a procedure similar to one used in Theorem 2 for proving this Theorem. If we partition $x \in [0, +\infty)$ in two parts, i.e. $x \in [0, \bar{K})$ and $x \in [\bar{K}, +\infty)$. Using this fact we note that the feasible region of (3.1.5) can be represented as

$$
- \sum_{i=0}^{n} q_i y_i \geq 0, \quad \forall \ x \in [0, \bar{K})
$$

(3.2.12)

and

$$
-(y_0 + \bar{K}) - (y_1 - 1)x - \sum_{i=2}^{n} q_i y_i \geq 0, \quad \forall \ x \in [\bar{K}, +\infty)
$$

(3.2.13)
Using Proposition 3, we conclude that (3.2.12) is true if and only if there exists a positive semidefinite matrix \( X = [x_{ij}] \), \( i, j = 0, ..., n \), such that

\[
0 = \sum_{i, j: i+j=2l-1} x_{ij}, l = 1, ..., n, \\
\sum_{r=0}^{l} y_{r} \left( \frac{n-r}{l-r} \right) \tilde{K}^{r} = - \sum_{i, j: i+j=2l} x_{ij}, l = 0, ..., n, \tag{3.2.14}
\]

\( X \succeq 0 \)

Using Proposition 4, we conclude that (3.2.13) is true if and only if there exists a positive semidefinite matrix \( Z = [z_{ij}] \), \( i, j = 0, ..., n \), such that

\[
\sum_{i, j: i+j=2l-1} z_{ij} = 0, l = 1, ..., n, \\
\sum_{h=1}^{n} y_{h} \left( \begin{array}{c}
 h \\
 l
\end{array} \right) \tilde{K}^{h-l} = - \sum_{i, j: i+j=2l} z_{ij}, l = 0, \tag{3.2.15}
\]

\[
\sum_{h=1}^{n} y_{h} \left( \begin{array}{c}
 h \\
 l
\end{array} \right) \tilde{K}^{h-l} = 1 = - \sum_{i, j: i+j=2l} z_{ij}, l = 1, \\
\sum_{h=1}^{n} y_{h} \left( \begin{array}{c}
 h \\
 l
\end{array} \right) \tilde{K}^{h-l} = - \sum_{i, j: i+j=2l} z_{ij}, l = 2, ..., n
\]

\( Z \succeq 0 \)

Hence we obtain the constraints for our SDP problem by simplifying (3.2.14) and (3.2.15). \( \square \)

### 3.3 Computing the Moments

In order to implement the SDP problems given in 3.2.1, 3.2.6 and 3.2.11, we need to compute the moments \( q_{i} \), \( i = 0, ..., n \) of the underlying asset price. For reference implementation, we will assume that the underlying asset price follows geometric Brownian motion (GBM) and thus compare the derived option values with Black Scholes results. We now present an explicit formula to calculate the \( n^{th} \) moment of underlying asset price \( X_{t} \) as below. Assuming that \( X_{t} \) follows a standard geometric Brownian motion, it satisfies

\[
dX_{t} = \mu X_{t} dt + \sigma X_{t} dz \tag{3.3.1}
\]

where \( dz = \epsilon(t) \sqrt{dt} \) is a Wiener process, \( \epsilon(t) \) is a standardized normal random variable. We know the analytical solution for Geometric Brownian motion is given by:

\[
E_{\pi} [(X_{T})] = X_{0} e^{\nu \tau + \sigma \sqrt{\tau} \epsilon} \]

where \( dz \sim N(0, \sqrt{dt}) \) and \( \nu = \mu - \frac{1}{2} \sigma^{2} \). Now, we intend to calculate \( E_{\pi} [(X_{T})^{n}] \) as below:

\[
E_{\pi} [(X_{T})^{n}] = E_{\pi} \left[ \left( X_{0} e^{\nu \tau + \sigma \sqrt{\tau} \epsilon} \right)^{n} \right] \quad \text{where} \quad \phi_{\epsilon}(z) = \frac{e^{-\frac{1}{2} z^{2}}}{\sqrt{2\pi}}
\]

\[
= \int_{-\infty}^{+\infty} X_{0}^{n} e^{\nu n \tau + n \sigma \sqrt{\tau} \epsilon} e^{-\frac{1}{2} z^{2}} \sqrt{2\pi} dz
\]

\[
= X_{0}^{n} e^{n \nu \tau} \int_{-\infty}^{+\infty} e^{n \sigma \sqrt{\tau} \epsilon - \frac{1}{2} z} \sqrt{2\pi} dz
\]
Knowing the fact that

\[ n\sigma \sqrt{\tau}z - \frac{1}{2}z^2 = -\frac{1}{2}(z - n\sigma \sqrt{\tau})^2 + \frac{n^2\sigma^2\tau}{2} \]

we get

\[ E_\pi[(X_T)^n] = X_0^n e^{n\nu\tau} \frac{e^{\frac{n^2\sigma^2\tau}{2}}}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-\frac{1}{2}(z-n\sigma \sqrt{\tau})^2} \frac{dz}{\sqrt{2\pi}} \]

\[ = X_0^n e^{n\nu\tau + \frac{n^2\sigma^2\tau}{2}} \int_{-\infty}^{+\infty} e^{-\frac{1}{2}(z-n\sigma \sqrt{\tau})^2} \frac{dz}{\sqrt{2\pi}} \]

because

\[ \int_{-\infty}^{+\infty} e^{-\frac{1}{2}(z-n\sigma \sqrt{\tau})^2} \frac{dz}{\sqrt{2\pi}} = 1 \]

Finally we conclude that the \( n^{th} \) moment of \( X_T \), \( E_\pi[(X_T)^n] \), is given by,

\[ E_\pi[(X_T)^n] = X_0^n e^{n\nu\tau + \frac{n^2\sigma^2\tau}{2}} \] (3.3.2)
Chapter 4

Method of Moments and SDP Relaxations

In this section, we will cover the numerical methods proposed by Lasserre, Prieto-Rumeau and Zervos in [21] for pricing exotic options. These methods are applied to options where underlying asset price follows a geometric Brownian motion or other mean-reverting processes of particular interest. The main idea of this method is to identify the derivative price with infinite-dimensional linear programming problems. These problems involve moments of appropriate measures. Finally we develop a finite-dimensional relaxation which can be converted to a semidefinite programming (SDP) problem indexed by the number of moments involved.

4.1 Introduction

The whole approach proposed by Lasserre, Prieto-Rumeau and Zervos in [21] can be briefly described as below:

1. The price of a Exotic option has to be identified with a linear combination of moments of suitably defined measures.

2. The martingale property of certain associated stochastic integrals is then exploited to derive an infinite system of linear equations involving the moments of the measures considered.

3. Hence the value of an exotic option is characterized with the solution of an infinite-dimensional linear programming (LP) problem. The variables of this LP problem involves the moments of certain suitably defined measures.

4. A finite-dimensional relaxation is obtained by restricting the infinite-dimensional LP problem to one that involves only a finite number of moments. Extra constraints called moment conditions are introduced in the LP problem (which thus converts it to a SDP problem) for achieving the finite-dimensional relaxation. These conditions reflect necessary conditions for a set of scalars to be identified with moments of a measure supported on a given set.

5. The result is a LP problem or a semidefinite programming (SDP) problem, which depends on the choice of moment conditions. By extremizing the resulting problems, upper and lower bounds for the value of the option under consideration are obtained. Also the quality of such bounds is enhanced as the number of moments increases.
4.2 Deriving Infinite-Dimensional LP Problem

Suppose we are given a diffusion $Z_t$

$$dZ_t = \beta(Z_t) \, dt + e(Z_t) \, dB_t, \quad Z_0 = z_0 \in \mathbb{R}^n$$  \hspace{1cm} (4.2.1)

We intend to approximate a function of diffusion $Z_t$ given above. In this section methodology of moments will be developed as a first step in achieving this goal. We will first derive an infinite-dimensional optimization problem, solution of which will give us the approximate value of this functional. As this infinite-dimensional problem is unsolvable, the next step is to derive finite dimensional relaxations of these problems. The variables in these optimization problems are moments defined on appropriate measures. When we derive the finite dimensional relaxation, we replace these unknown moments by scalars, and add necessary conditions (called moment conditions) for these scalars to be identical with the moments of measures with appropriate supports.

4.2.1 Basic Adjoint Equation

In order to derive the Basic Adjoint Equation, necessary for developing the moments approach, we will first introduce following notations and terminology.

**Diffusion being considered:** Given the filtered probability space, $(\Omega, \mathcal{F}, \mathcal{F}_t, P)$, where $\Omega$ is given sample space, $\sigma$-algebra $\mathcal{F}$ of subsets of $\Omega$, a measure $P$ on $(\Omega, \mathcal{F})$, such that $P(\Omega) = 1$ and a filtration $\mathcal{F}_t$ on $\mathcal{F}$. Define $Z_t$ as an $n$-dimensional diffusion defined on $(\Omega, \mathcal{F}, \mathcal{F}_t, P)$

$$dZ_t = \beta(Z_t) \, dt + e(Z_t) \, dB_t, \quad Z_0 = z_0 \in \mathbb{R}^n$$  \hspace{1cm} (4.2.2)

where $\beta: \mathbb{R}^n \to \mathbb{R}^n$ and $e: \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^m$ are deterministic functions such that SDE (4.2.2) has unique strong solution.

**Infinitesimal Generator:** The infinitesimal generator of diffusion $Z_t$ is given by

$$A f(z) = \sum_i \beta_i(z) \frac{\partial f}{\partial z_i} + \frac{1}{2} \sum_{i,j} \left( \sigma(z) \sigma(z)^T \right) \frac{\partial^2 f}{\partial z_i \partial z_j}(z)$$  \hspace{1cm} (4.2.3)

or in terms of the gradient and scalar and Frobenius inner products,

$$A f(z) = b(z) \cdot \nabla f(z) + \frac{1}{2} \left( \sigma(z) \sigma(z)^T \right) \cdot \nabla^2 f(z)$$  \hspace{1cm} (4.2.4)

where $f \in D(A)$, and $D(A)$ contains set of all twice-continuously differentiable functions $f: \mathbb{R}^n \to \mathbb{R}$ with compact support. We now impose two assumptions for further development of this approach.

**Assumption 1:** The entries of the vector $\beta(z)$ and the matrix $(ee^T)(z)$ are polynomials in $z$, hence $A$ maps polynomials into polynomials.

**Assumption 2:** Sum of all moments for diffusion $Z_t$ is finite, i.e.

$$\sup_{t \in [0,T]} \sum_{i=0}^n \mathbb{E} \left[ |Z_t^i|^k \right] \leq \infty, \quad \forall \ T \geq 0, \quad \forall \ k \in \mathbb{N}$$  \hspace{1cm} (4.2.5)

Now we consider the following two measures on diffusion $Z_t$

**Expected Occupation Measure:** The expected occupation measure $\mu(.) = \mu(.; z_0)$ of diffusion $Z$ up to time $\tau$ that is defined by

$$\mu(B) := \mathbb{E} \left[ \int_0^\tau I_{\{Z_s \in B\}} ds \right], \quad B \in \mathcal{B} (\mathbb{R}^n),$$  \hspace{1cm} (4.2.6)

1Compared to weak solution which consists of a probability space and a process that satisfies the SDE, a strong solution is a process that satisfies the equation and is defined on a given probability space.
where $\mathcal{B}(\mathbb{R}^n)$ is the Borel $\sigma$-algebra on $\mathbb{R}^n$.

**Exit Location Measure**: The exit location measure $\nu(\cdot) = \nu(\cdot; z_0)$ that is defined by

$$\nu(B) := P(Z_\tau \in B), \quad B \in \mathcal{B}(\mathbb{R}^n),$$

which is nothing but probability distribution of $Z_\tau$.

We are now ready to derive the Basic Adjoint Equation given the definitions and assumptions above. Specifically, given the Assumptions 1 and 2 above the following process

$$M_t^f := f(Z_t) - f(z_0) - \int_0^t Af(Z_s)ds$$

$$= \int_0^t [e^T \nabla_z f]^T (Z_s) dBS, \quad t \geq 0,$$

where $f: \mathbb{R}^n \rightarrow \mathbb{R}, f \in D(\mathcal{A})$ as defined above, is a square integrable martingale.

Now, Doob’s optional sampling theorem\(^2\) states: For $X = \{X_t\}_{t \geq 0}$, which is an $(\mathcal{F}_t, \mathbb{P})$ martingale, and $S, T$ are stopping times bounded by constant $c$, with $S \leq T$, almost surely, then

$$E_{\mathbb{P}}[X_T|\mathcal{F}_S] = X_S, \quad \mathbb{P} \text{ almost surely.}$$

Applying Doob’s optional sampling theorem to (4.2.8) at times 0 and $T$ we get,

$$E[M_t^f] = E[M_0^f]$$

$$E[f(Z_t) - f(z_0) - \int_0^t Af(Z_s)ds] = E[f(Z_0) - f(z_0) - \int_0^0 Af(Z_s)ds]$$

and by simple calculations we know that,

$$f(Z_0) - f(z_0) - E\left[\int_0^0 Af(Z_s)ds\right] = 0$$

hence we get the following result

$$E[f(Z_t)] - f(z_0) - E\left[\int_0^t Af(Z_s)ds\right] = 0$$

Using the definitions of expected occupation measure $\mu(\cdot)$, and exit location measure $\nu(\cdot)$ as given above, we can rewrite (4.2.11) as below

$$\int_{\mathbb{R}^n} f(z) \nu(dz) - f(z_0) - \int_{\mathbb{R}^n} Af(z) \mu(dz) = 0$$

which is called the basic adjoint equation. We can consider this equation as defining the relationship between the measures $\mu$ and $\nu$ associated with the generator $\mathcal{A}$ (for more details please refer to article [12] by Helmes, Rohl and Stockbridge).

\(^2\)We can roughly interpret Doob’s optional sampling theorem implying that in a Casino in a fair world, where returns are martingales and gambler is restricted to quit at stopping times, then there is almost zero probability to improve the expected return by judicious choice of stopping time.
4.2 Martingale Moment Conditions

Given a multiindex $\alpha \in \mathbb{N}$, if $f$ is the monomial

$$f(z) = z^\alpha = \prod_{j=1}^{n} z_j^{\alpha_j} \quad (4.2.13)$$

it can be seen from then Assumption 1 (4.2.5) that there exists a finite collection $c_{\beta}(\alpha)$ of real numbers such that

$$A f(z) = \sum_{\beta} c_{\beta}(\alpha) z^\beta, \quad \forall z \in \mathbb{R}^n \quad (4.2.14)$$

Defining $\mu^\alpha = \mu^\alpha(z_0)$ and $\nu^\alpha = \nu^\alpha(z_0)$ as the moments of $\mu$ and $\nu$ respectively, which we assume are finite, i.e.

$$\mu^\alpha = \int_{\mathbb{R}^n} z^\alpha \mu(dz) \leq \infty \quad (4.2.15)$$

and

$$\nu^\alpha = \int_{\mathbb{R}^n} z^\alpha \nu(dz) \leq \infty \quad (4.2.16)$$

then basic adjoint equation (4.2.12) gives us the following infinite system of linear equations which link the moments $\mu$ and $\nu$,

$$\nu^\alpha - \sum_{\beta} c_{\beta}(\alpha) \mu^\beta = z_0^\alpha, \quad \forall \alpha \in \mathbb{N} \quad (4.2.17)$$

Consider the functional $J(z_0)$ of the diffusion $Z_t$ that is defined by

$$J(z_0) := \mathbb{E}[p(Z_\tau)] = \sum_{j=1}^{k} \int_{K_j} j(z) \nu(dz), \quad (4.2.18)$$

where $\{K_j, j = 0, \ldots, k\}$ is a given Borel measurable partition of $\mathbb{R}^n$, and $p(z)$ and $p_j(z)$ are defined as

$$p(z) := \sum_{j=1}^{k} p_j(z) I_{K_j}(z), \quad z \in \mathbb{R}^n, \quad (4.2.19)$$

and

$$p_j(z) := \sum_{\alpha} p_{j\alpha} z^\alpha, \quad z \in K_j, \quad \forall j = 1, \ldots, k, \quad (4.2.20)$$

Considering $\nu_j$ as partitions of $\nu$, each defined on $K_j$, for $j = 1, \ldots, k$, i.e.

$$\nu_j(.) := \nu_j(:; z_0) := \nu_j(:; z_0)|_{K_j} \quad (4.2.21)$$

then from (4.2.19) and (4.2.20) we can see that

$$J(z_0) = \sum_{j=1}^{k} \sum_{\alpha} p_{j\alpha} \nu_j^\alpha \quad (4.2.22)$$

which is a linear combination of the moments $\{\nu_j^\alpha\} = \{\nu_j^\alpha(z_0)\}$ of the measures $\nu_j$.  

34
4.2.3 Using Available Moments

If the moments \{\nu_\alpha^j\} of the measure \nu^\alpha are easily computable (which is the case in European and Asian options as elucidated in subsequent sections), we can bound the functional \(J(z_0)\) defined in (4.2.18) by obtaining the upper and lower bounds of infinite dimensional LP problem \(Q^I(z_0)\) defined by

\[
\begin{align*}
\text{extremize} & \sum_{j=1}^k \sum_{\alpha} \sum_{a} p_{ja} \nu_j^\alpha, \\
\text{subject to} & \nu_j^\alpha = \nu^\alpha, \quad \alpha \in \mathbb{N}^n \\
& \nu_j \in \mathcal{M}(K_j), \quad j = 1, \ldots, k,
\end{align*}
\]

(4.2.23)

noting that space of all Borel measures with finite moments of all orders that are supported on a given Borel measurable set \(K \subseteq \mathbb{R}^n\) is denoted by \(\mathcal{M}(K_j)\). We also note that the constraints defining the feasible region of \(Q^I(z_0)\) are necessary moment conditions and hence

\[\inf Q^I(z_0) \leq J(z_0) \leq \sup Q^I(z_0)\]  \hspace{1cm} (4.2.24)

When \(\nu\) is moment-determinate\(^3\) then (4.2.24) is satisfied with equality because measure \(\nu(\cdot)\) is unique.

4.2.4 Using Martingale Moments Conditions

More generally when we cannot compute moments \(\nu\) easily, we consider the infinite dimensional LP problem \(Q^{II}(z_0)\) defined by

\[
\begin{align*}
\text{extremize} & \sum_{j=1}^k \sum_{\alpha} \sum_{a} p_{ja} \nu_j^\alpha, \\
\text{subject to} & \nu_j^\alpha = \nu^\alpha, \quad \alpha \in \mathbb{N}^n \\
& \nu_j - \sum_{\beta} c_\beta(\alpha) \mu_\beta^\alpha = z_0^\alpha, \quad \forall \alpha \in \mathbb{N} \\
& \mu_j \in \mathcal{M}(\mathbb{R}^n), \quad \mu_j \in \mathcal{M}(K_j), \quad j = 1, \ldots, k,
\end{align*}
\]

(4.2.25)

where the martingale moment conditions in (4.2.17) are taken into consideration as constraints for LP problem \(Q^{II}(z_0)\). Similar to (4.2.23), we again note the fact that the constraints for LP problem \(Q^{II}(z_0)\) are necessary moment conditions and hence

\[\inf Q^{II}(z_0) \leq J(z_0) \leq \sup Q^{II}(z_0)\]  \hspace{1cm} (4.2.26)

When \(\nu\) is moment-determinate then (4.2.26) is satisfied with equality and \(Q^{II}\) is equal to the value of \(J(z_0)\).

Now that we have derived the relevant infinite dimensional LP problem, we would try to address the problem of deriving the finite dimensional relaxations of problems (4.2.23) and (4.2.25). For doing that, development of the background material is presented in following section is essential.

\(^3\)Moment determinacy can be defined as equality of \(\nu\) and \(\mu\), for given measure \(\nu\) on \(\mathbb{R}^n\) having finite moments of all orders and whenever \(\int z^\alpha \mu(dz) = \int z^\alpha \nu(dz)\). Example of moment determinate probability distributions are normal and noncentral \(\chi^2\) distributions, whereas lognormal distribution is not moment-determinate.
4.3 Polynomial Optimisation and Problem of Moments

In [18], Lasserre showed that problem of finding unconstrained global optimum of a real valued polynomial, \( p(x) : \mathbb{R}^n \to \mathbb{R} \) can be reduced to solving finite sequence of convex Linear Matrix Inequalities (LMI) (or semidefinite programming (SDP)) problems. The implementation was given in [19] by Lasserre and Henrion. The same approach can be applied to \( p(x) \) over a compact set \( K \) defined by polynomial inequalities.

To elaborate, given a real-valued polynomial \( p(x) : \mathbb{R}^n \to \mathbb{R} \), the following problem needs to be solved

\[
\min_{x \in \mathbb{R}^n} p(x)
\]

(4.3.1)

Also when \( K \) is a compact set (convexity assumption not required) defined by polynomial inequalities \( g_i(x) \geq 0, i = 1, \ldots, r \), we obtain the following problem.

\[
\min_{x \in K} p(x)
\]

(4.3.2)

The primal LMI relaxation \( Q_i \) of \( p(x) \) given below aims to find the moments of a probability measure with mass concentrated on some global minimizer of \( p(x) \)

\[
Q_i := \min_{\mu} \left\{ \int p(x)d\mu \mid \mu(K) = 1 \right\}
\]

(4.3.3)

where minimum is taken over all probability measures on the feasible set of \( p(x) \), \( \mathbb{R}^n \) and \( K \) for (4.3.1) and (4.3.1) respectively.

4.3.1 Notation and Definition

Let \( s(r) = \left( \begin{array}{c} n + r \\ r \end{array} \right) \), and

\[1, x_1, x_2, \ldots, x_n, x_1^2, x_1x_2, \ldots, x_1x_n, x_2^2, \ldots, x_n^2, \ldots, x_m^m \]

be the basis for the \( m \)-degree real-valued polynomials \( p(x) : \mathbb{R}^n \to \mathbb{R} \), and let \( s(2m) \) be its dimension. We write the polynomial \( p(x) : \mathbb{R}^n \to \mathbb{R} \) as below

\[ p(x) = \sum_{\alpha} p_{\alpha} x^\alpha \]

with \( x^\alpha := x_1^{\alpha_1}x_2^{\alpha_2}\ldots x_n^{\alpha_n} \)

(4.3.5)

and \( \alpha_i \in \mathbb{N} \), \( \sum_i \alpha_i \leq m, \; i = 1, \ldots, n \)

where coefficient vector of \( p(x) \) in the basis (4.3.4) has been denoted by \( p = \{p_{\alpha}\} \in \mathbb{R}^{s(m)} \).

**Moment Matrix:** Assuming that we are given a vector \( y := \{y_\alpha\} \) of length \( s(2m) \) which confirms with indexing implied by the basis (4.3.4), we define a matrix of dimensions \( s(m) \times s(m) \) called moment matrix \( M_k(y) \) which has rows and columns labelled by (4.3.4). To be more precise, \( M_k(y) \) is defined as below:

\[
M_k(y)(1, i) = M_k(y)(i, 1) = y_{i-1} \quad \text{for} \; i = 1, \ldots, k + 1
\]

\[
M_k(y)(1, j) = y_\alpha \quad \text{and} \quad M_k(y)(i, 1) = y_\beta \Rightarrow M_k(y)(i, j) = y_{\alpha+\beta}
\]

(4.3.6)

In other words we can say that matrix \( M_k(y) \) is the block matrix \( \{M_k(y)\}_{0 \leq i, j \leq 2m} \) defined by

\[
M_{i,j}(y) = \begin{bmatrix}
y_{i+j,0} & y_{i+j-1,1} & \cdots & y_{i,j} \\
y_{i+j,1} & y_{i+j-2,2} & \cdots & y_{i-1,j+1} \\
\vdots & \vdots & \ddots & \vdots \\
y_{j,i} & y_{i+j-1,1} & \cdots & y_{0,i+j}
\end{bmatrix}
\]

(4.3.7)
where $y_{i,j}$ represents $(i+j)^{th}$-order moment $\int x^iy^j d\mu(x,y)$ for some probability measure $\mu$.

To understand the concept of moment matrix, we give 3 examples:

**Example 1:** When $n = 1$, then $y = \{y_0, y_1, y_2, \ldots \}$ and $M_k(y)$ is same as **Hankel Matrix**

$$H_k(y)(i,j) = y_{i+j-2}, \quad i,j = 1, \ldots, k + 1$$

**Example 2:** When $n = 2$ and $m = 2$, we get

$$M_2(y) = \begin{bmatrix} y_{0,0} & y_{1,0} & y_{2,0} & y_{1,1} & y_{0,2} \\ y_{1,0} & y_{2,0} & y_{1,1} & y_{2,1} & y_{1,2} \\ y_{0,1} & y_{1,1} & y_{0,2} & y_{1,2} & y_{0,3} \\ y_{2,0} & y_{3,0} & y_{2,1} & y_{3,1} & y_{2,2} \\ y_{1,1} & y_{2,1} & y_{1,2} & y_{2,2} & y_{1,3} \\ y_{0,2} & y_{1,2} & y_{0,3} & y_{1,3} & y_{0,4} \end{bmatrix}$$

(4.3.8)

**Example 3:** For the three-dimensional case, definition of $M_k(y)$ is given via blocks $\{M_{i,j,k}(y)\}$, $0 \leq i,j,k \leq 2m$ as done for the case of $n = 2$ and $m = 2$, and so on.

**Localizing Matrix:** Let $q(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ be a real-valued polynomial of degree $w$ with coefficient vector $q \in \mathbb{R}^{(w)}$. If the entry $(i,j)$ of the matrix $M_k(y)$ is $y_{ij}$, let $\beta(i,j)$ denote the subscript $\beta$ of $y_{ij}$. Let $M_k(q,y)$ be the matrix defined by

$$M_k(q,y)(i,j) = \sum_\alpha q_\alpha y_{(\beta(i,j)+\alpha)}$$

(4.3.9)

For example, given

$$M_1(y) = \begin{bmatrix} y_{0,0} & y_{1,0} & y_{0,1} \\ y_{1,0} & y_{2,0} & y_{1,1} \\ y_{0,1} & y_{1,1} & y_{0,2} \end{bmatrix}$$

and $q(x) = ax_1 - bx_1x_2 - cx_2^2$,

(4.3.10)

we obtain

$$M_1(q,y) = \begin{bmatrix} ay_{1,0} - by_{1,1} - cy_{0,2} & ay_{2,0} - by_{2,1} - cy_{1,2} & ay_{1,1} - by_{1,2} - cy_{0,3} \\ ay_{2,0} - by_{2,1} - cy_{1,2} & ay_{3,0} - by_{3,1} - cy_{2,2} & ay_{2,1} - by_{2,2} - cy_{1,3} \\ ay_{1,1} - by_{1,2} - cy_{0,3} & ay_{2,1} - by_{2,2} - cy_{1,3} & ay_{1,2} - by_{1,3} - cy_{0,4} \end{bmatrix}$$

(4.3.11)

**4.3.2 Necessary Moment conditions**

**Proposition 7.** If $y = \{y_\alpha\}$ (with $y_0, \ldots, 0 = 1$) is a vector of moments up to order $2k$ of some probability measure $\mu_y$, then

$$M_k(q,y) \succeq 0$$

(4.3.12)

*Proof.* Consider $v(x) : \mathbb{R}^k \rightarrow \mathbb{R}$, a real valued polynomial of degree at most $k$ and a vector of coefficients $(f_\alpha, |\alpha| \leq k)$, in the basis (4.3.4). Then we have

$$\langle f, M_k(y)f \rangle = \int f^2 d\mu \geq 0$$

(4.3.13)

□

**Proposition 8.** If $y = \{y_\alpha\}$ (with $y_0, \ldots, 0 = 1$) is a vector of moments up to order $2k$ of some probability measure $\mu_y$, then

$$M_k(q,y) \succeq 0$$

(4.3.14)

*Proof.* To prove this consider $v(x) : \mathbb{R}^k \rightarrow \mathbb{R}$, a real valued polynomial of degree at most $k$ and a vector of coefficients $(f_\alpha, |\alpha| \leq k)$, in the basis (4.3.4). Then we have

$$\langle f, M_k(q,y)f \rangle = \int f^2 q d\mu \geq 0$$

(4.3.15)

□
Hence from Proposition 7 (4.3.12) and Proposition 8 (4.3.14), we can say that necessary conditions for elements of \( y = \{ y_\alpha \} \) to be moments of some measure \( \mu_y \) on semialgebraic set \( \mathcal{K} \), i.e. the set of form
\[
\mathcal{K} := \{ x \in \mathbb{R}^n \mid v_i(x) \geq 0, \text{ for all } i = 1, \ldots, l \}
\] (4.3.16)
where \( v_i, i = 1, \ldots, l \) are given polynomials, is given by
\[
M_k(y) \succeq 0 \text{ and } M_k(v_i, y) \succeq 0, \quad i = 1, \ldots, l, \quad k = 1, 2, \ldots
\] (4.3.17)

We also note that above conditions are necessary conditions but not sufficient conditions for elements of \( y = \{ y_\alpha \} \) to be moments of measure \( \mu_y \) supported on set \( \mathcal{K} \). Before going to the next section, we note the \( \mathcal{K} \)-moment problem statement: \( \mathcal{K} \)-moment problem identifies sequences \( y_\alpha \) that are moment-sequences of measure \( \mu \) with support contained in semialgebraic set \( \mathcal{K} \).

### 4.3.3 SDP Relaxations

After deriving the necessary conditions for a vector \( y \) to be moment of measure \( \mu \), we will derive the SDP relaxations for problems \( p(x) \) in (4.3.2) and (4.3.4). For problem \( p(x) \) and \( \mathcal{K} \), let the associated feasibility set be defined as
\[
\mathcal{K} = \{ x \in \mathbb{R}^n \mid q_k(x) \geq 0, \quad k = 0, \ldots, m \}
\] (4.3.18)

Also let degree of \( q_k = 2v_k + 1 \) or \( 2v_k \), depending on its parity. The vectors \( q_k \in \mathbb{R}^{s(2v_k)} \) are extended to vectors of \( \mathbb{R}^{s(2i)} \) by appending extra zeros when needed, for any \( i \geq \max_{k \in \{0, m\}^1} v_k \).

Consider the following family of convex positive semidefinite optimization problems \( Q_i \),
\[
\inf_y \sum_{\alpha} p_\alpha y_\alpha \\
\text{subject to } M_i(y) \succeq 0 \\
M_{i-v_k}(q, y) \succeq 0
\] (4.3.19)

which are also called Linear Matrix Inequality (LMI) problems.

**Assumption 1**: Given semialgebraic set \( \mathcal{K}_q := \{ x \in \mathbb{R}^n \mid q(x) \geq 0 \} \) and a polynomial \( p(x) \), strictly positive on \( \mathcal{K}_q \), then \( p(x) \) can be written as
\[
p(x) = \sum_{i=1}^{r_1} q_i(x)^2 + \theta(x) \sum_{j=1}^{r_2} t_j(x)^2,
\] (4.3.20)

for some polynomials \( q_i(x), t_j(x), i = 1, \ldots, r_1, j = 1, \ldots, r_2 \).

Under the above Assumption, it was proved in by Lasserre in [18], that
\[
\inf Q_i \uparrow \inf p(x), \quad \text{as } i \to \infty
\] (4.3.21)

or in other words, the problem of obtaining the bounds of \( p(x) \) in the original problem (4.3.1) can be accurately solved with desired accuracy by solving problem of obtaining bounds for convex LMI problem \( Q_i \) for every \( i \geq \max_{k \in \{0, m\}^1} v_k \).

### 4.3.4 Summary

To summarise the contents in above section, a result that applies when \( n = 1 \) is stated which will be useful in the following sections.
\(K\)-Moment Problem: Given a vector \(y = (y_0, y_1, \ldots, y_{2r})\) and a set \(K \subseteq \mathbb{R}\), the \(K\)-moment problem investigates necessary and sufficient conditions for \(y\) to be identified with the corresponding vector of moments of some measure supported on \(K\). This problem is called the truncated Hausdorff moment problem if \(K = [a, b]\), and the truncated Stieltjes moment problem if \(K = [a, +\infty)\).

Necessary Conditions: Given a vector \(y = (y_0, y_1, \ldots, y_{2r}) \in \mathbb{R}^{2r+1}\), the following statements are true:

- **Truncated Hausdorff moment problem**: \(M_r(y) \geq 0\) and \(M_r(g, y) \geq 0\) with \(g(x) := (b-x)(x-a)\), are necessary and sufficient conditions for the elements of \(y\) to be the first \(2r+1\) moments of a measure supported on \([a, b]\) for the truncated Hausdorff moment problem.

- **Truncated Stieltjes moment problem**: \(M_r(y) \geq 0\) and \(M_r(g, y) \geq 0\) with \(g(x) := x-a\), are sufficient conditions for the elements of \(y\) to be the first \(2r+1\) moments of a measure supported on \([a, +\infty)\) for the truncated Stieltjes moment problem.

### 4.4 Finite Dimensional Relaxations

We will now derive the finite dimensional relaxations of problems \(Q^I(z^0)\) and \(Q^{II}(z^0)\), given by (4.2.23) and (4.2.25). Given a Borel measurable set \(K \subseteq \mathbb{R}^n\), let \(M(K)\) be the set of all Borel measures with support contained in \(K\), and with all moments finite. Consider a set \(\mathcal{S}_r(K)\) that is defined by appropriate necessary moment conditions for scalars \(\eta_\alpha\), \(|\alpha| \leq 2r\) to be moments of some measure in \(M(K)\), as described in section 4.3.4.

We define another set \(\mathcal{N}_r(K)\), such that \(\mathcal{S}_r(K) \supseteq \mathcal{N}_r(K)\) and

\[
\mathcal{N}_r(K) := \left\{ \int x^\alpha \mu(dx) \middle| \alpha \in \mathbb{N}^n, \ |\alpha| \leq 2r, \ \mu \in M(K) \right\}
\]  

(4.4.1)

where \(|\alpha| := \sum_{i=1}^n \alpha_i \leq 2r\). Then we consider the finite dimensional relaxations of problems \(Q^I(z^0)\) which was defined in (4.2.23), denoted here as \(Q'^I_r(z^0)\) as

\[
\text{extremize} \sum_{j=1}^k \sum_{\alpha} p_{j\alpha} \eta_\alpha^0,
\]

subject to

\[
\sum_{j=1}^k \eta_\alpha^0 = \eta_\alpha, \ |\alpha| \leq 2r
\]

\[
\eta_j = \left\{ \eta_\alpha^0, \ |\alpha| \leq 2r \right\} \in \mathcal{S}_r(K_j), \ j = 1, \ldots, k
\]

and when moments are not known, the finite dimensional relaxation of \(Q^{II}(z^0)\) which was defined in (4.2.23), denoted here as \(Q'^{II}_r(z^0)\) as

\[
\text{extremize} \sum_{j=1}^k \sum_{\alpha} p_{j\alpha} \eta_\alpha^0,
\]

subject to

\[
\sum_{j=1}^k \eta_\alpha^0 - \sum_{\beta} c_{\beta}(\alpha) \vartheta_\beta = z_0^\alpha, \ |\alpha| \leq 2r
\]

\[
\eta_j = \left\{ \eta_\alpha^0, \ |\alpha| \leq 2r \right\} \in \mathcal{S}_r(K_j), \ j = 1, \ldots, k
\]

\[
\vartheta_j = \left\{ \vartheta_\beta^j, \ |\alpha| \leq 2r \right\} \in \mathcal{S}_r(\mathbb{R}^n)
\]

We will now specify the sets \(\mathcal{S}_r(K)\) and \(\mathcal{N}_r(K)\) in accordance with notation introduced in section 4.3.4 for following three cases:
• $\mathcal{K} \subseteq \mathbb{R}$ is the interval $[a, b]$

• $\mathcal{K} \subseteq \mathbb{R}$ is the interval $[a, +\infty)$

• $\mathcal{K} \subseteq \mathbb{R}$ is the interval $(-\infty, a]$

We can clearly see that $\mathcal{S}_r(\mathcal{K})$ is given by necessary moment conditions for a vector $y \in \mathbb{R}^{2r+1}$ as defined below

- When $\mathcal{K} = [a, b]$ the set $\mathcal{S}_r$ is given by:
  \[
  \mathcal{S}_r(\mathcal{K}) = \left\{ y \in \mathbb{R}^{2r+1} \mid M_r(y) \succeq 0 \text{ and } M_{r-1}(g, y) \succeq 0, \quad g(x) := (b - x)(x - a) \right\}
  \]  
  (4.4.4)

- When $\mathcal{K} = [a, +\infty)$ the set $\mathcal{S}_r$ is given by:
  \[
  \mathcal{S}_r(\mathcal{K}) = \left\{ y \in \mathbb{R}^{2r+1} \mid M_r(y) \succeq 0 \text{ and } M_{r-1}(g, y) \succeq 0, \quad g(x) := (x - a) \right\}
  \]  
  (4.4.5)

- When $\mathcal{K} = (-\infty, a]$ the set $\mathcal{S}_r$ is given by:
  \[
  \mathcal{S}_r(\mathcal{K}) = \left\{ y \in \mathbb{R}^{2r+1} \mid M_r(y) \succeq 0 \text{ and } M_{r-1}(g, y) \succeq 0, \quad g(x) := (a - x) \right\}
  \]  
  (4.4.6)

and set $\mathcal{N}_r(\mathcal{K}) \subseteq \mathcal{S}_r(\mathcal{K})$ is the set of all vectors that provide the first $2r+1$ moments of a measure with finite moments of all orders and supported on $\mathcal{K}$ for any given $r \geq 1$.

Before we give the final SDP problems to be solved, we note the relationship between $\mathcal{S}_r(\mathcal{K})$ and $\mathcal{N}_r(\mathcal{K})$ for following two cases:

• **Truncated Hausdorff moment problem**: Since $\mathcal{K} = [a, b]$ and conditions specified by $\mathcal{S}_r(\mathcal{K})$ are also sufficient conditions, we note that $\mathcal{S}_r(\mathcal{K}) = \mathcal{N}_r(\mathcal{K})$.

• **Truncated Stieltjes moment problem**: Define
  \[
  \mathcal{T}_r(\mathcal{K}) := \{ y \in \mathbb{R}^{2r+1} \mid M_r(y) \succ 0 \text{ and } M_{r-1}(g, y) \succ 0 \}
  \]  
  (4.4.7)

Hence as per the given definition we observe that $\mathcal{N}_r(\mathcal{K}) = \mathcal{T}_r(\mathcal{K})$ and from section 4.3.4 we see that $\mathcal{T}_r(\mathcal{K}) \subseteq \mathcal{N}_r(\mathcal{K})$. This results in $\mathcal{N}_r(\mathcal{K}) = \mathcal{S}_r(\mathcal{K})$, for $\mathcal{K} = [a, \infty)$ or $\mathcal{K} = (-\infty, a]$.

### 4.5 SDP Relaxations for European Options

We would now utilize the theory developed in previous sections for solving the problem of pricing the European options. The case of European options is comparatively simpler because it considers measures involving only the process $X_t$, which are supported on subsets of the real line.

Suppose the underlying asset price process satisfies the following stochastic differential equation

\[
\begin{align*}
  dX_t &= b(X_t)dt + \sigma(X_t)dW_t, \quad b, \sigma : \mathbb{R} \to \mathbb{R}, \quad X_0 = x_0 \in \mathbb{R}
\end{align*}
\]  

(4.5.1)

The value of a European call option written on the underlying $X$ is given by

\[
v_E(x_0) := e^{-\rho T} \mathbb{E} \left[ (X_T - K)^+ \right]
\]  

(4.5.2)

where $(X_T - K)^+$ denotes $\max(X_T - K, 0)$, $T > 0$ is the option’s maturity time, $K$ is the option’s strike price, $\rho$ is a constant discounting factor and $x_0$ is the initial underlying asset price. Under
a simplification, we set the constant discounting factor $\rho$ to 0, because its presence would not affect this analysis.

Now using the problem $Q^I(Z_0)$ to solve $v_E(x_0)$, we define $\eta$ as the exit location measure of $X_0$ and consider the restrictions $\eta_1$ and $\eta_2$ of measure $\eta$ on the sets $(-\infty, K)$ and $[K, +\infty)$, respectively as shown in figure 4.1.

Now we can express the equation (4.5.2) as

$$v_E(x_0) := \mathbb{E}[(X_T - K)^+] = \int_{\mathbb{R}} (x - K)^+ \eta(dx) = \eta_1^+ - K \eta_2^0$$

(4.5.3)

We hence conclude that the finite dimensional SDP problem we need to solve to obtain the upper and lower bounds for price of European call option is given by:

$$\text{extremize } \eta_1^+ - K \eta_2^0$$

subject to $\eta_1^j + \eta_2^j = \eta^j, \ j = 0, ..., 2r$

$$M_r(\eta_1) \succeq 0, \ M_{r-1}(g_1(x), \eta_1) \succeq 0,$$

$$M_r(\eta_2) \succeq 0, \ M_{r-1}(g_2(x), \eta_2) \succeq 0,$$

(4.5.4)

where $g_1$ and $g_2$ are the polynomials defined by

$$g_1(x) = (K - x)$$

$$g_2(x) = (x - K)$$

(4.5.5)

since $\eta_1$ and $\eta_2$ are defined on the sets $(-\infty, K]$ and $[K, +\infty)$, respectively. $M_r(\cdot), M_{r-1}(\cdot)$ are the moment and localizing matrices respectively. The last two constraints defining the feasible region of this SDP problem correspond to the necessary SDP moment conditions for $\{\eta_1^j, j = 0, ...2r\}$ and $\{\eta_2^j, j = 0, ...2r\}$ to be moments of measures supported on $(-\infty, K]$ and $[K, +\infty)$, respectively. For $\eta_1$, it is not necessary to consider moment conditions on $(-\infty, K]$ instead of $(-\infty, K)$, because we evaluate the expectation of a function which vanishes at $x = K$. 

Figure 4.1: Definition of Measures for European Options

4 Method of Moments and SDP Relaxations
The derivation of SDP problem for finding upper and lower bounds for price of European put option is very similar; only the objective function is changed in (4.5.4) which is given by $K\eta_0^2 - \eta_1^2$, rest of constraints remain the same.

4.5.1 Moments Computation

We assume that $X$ is the stochastic process defined in previous section. Now we intend to compute the moment of $\eta$, hence we consider the monomial $f(X) = X^i$ which is continuous and differentiable, hence we would apply Taylor’s expansion

$$\Delta f = \left( \frac{df}{dx} \right) \Delta X + \frac{1}{2} \left( \frac{d^2f}{dx^2} \right) \Delta X^2 + \frac{1}{6} \left( \frac{d^3f}{dx^3} \right) \Delta X^3 + ... \quad (4.5.6)$$

to $f(X)$ for a small change $\Delta X$ in $X$. We enlist the following three cases

**Case 1: Geometric Brownian Motion:** Here $\Delta X$ is given by

$$\Delta X_t = bX_t dt + \sigma X_t dW_t \quad (4.5.7)$$

and hence applying (4.5.6) to (4.5.7), we get

$$\Delta f = ix^{i-1}(bxdt + \sigma x dW_t) + \frac{1}{2}i(i-1)x^{i-2}(bxdt + \sigma x dW_t)^2 + \frac{1}{6}i(i-1)(i-2)x^{i-3}(bxdt + \sigma x dW_t)^3 + ... \quad (4.5.8)$$

where step 2 in above equation is obtained by ignoring ($\Delta t$) terms of order more than 1. Now taking the term $\frac{1}{2}i(i-1)\sigma^2 x_i(dW_t)^2$, we note the fact that $dW_t$ has expected value 0 and variance $\Delta t$, hence this last term is of order $\Delta t$ and can’t be ignored. It can be proved that as $\Delta t \to 0$, $(dW_t)^2 \to \Delta t$. Hence moments of $\eta$ denoted as $\eta^i = \mathbb{E}[X^i]$ can be found by solving the following system of ordinary differential equations

$$\frac{d}{dt} \eta^i(t) = ib\eta^i(t) + \frac{1}{2}i(i-1)\sigma^2 \eta^i(t), \quad 0 \leq i \leq 2r \quad (4.5.9)$$

with initial conditions $\eta^i(0) = (X_0)^i$ and the second term $i\sigma x dW_t$ vanishes because $\mathbb{E}[dW_t] = 0$.

**Case 2: Ornstein-Uhlenbeck Process:** Here $\Delta X$ is given by

$$\Delta X_t = \gamma(\theta - X_t)dt + \sigma dW_t \quad (4.5.10)$$

and hence applying (4.5.6) to (4.5.10), we get

$$\Delta f = ix^{i-1}(\gamma(\theta - x)dt + \sigma dW_t) + \frac{1}{2}i(i-1)x^{i-2}(\gamma(\theta - x)dt + \sigma dW_t)^2 + \frac{1}{6}i(i-1)(i-2)x^{i-3}(\gamma(\theta - x)dt + \sigma dW_t)^3 + ... \quad (4.5.11)$$

where similar analysis applies as we did for Model 1 (geometric Brownian motion). Specifically, step 2 in above equation is obtained by ignoring ($\Delta t$) terms of order more than 1. Now taking the fourth term $\frac{1}{2}i(i-1)(x^{i-2}\sigma^2(dW_t)^2)$, we note the fact that $dW_t$ has expected value 0 and...
We introduce the process \( \eta \) with initial conditions \( \eta(0) = (X_0)^i \) and the third term \( i\sigma x^{i-1}dW_t \) vanishes because \( \mathbb{E}[dW_t] = 0. 

**Case 3: Cox-Ingersoll-Ross Process:** Here \( \Delta X \) is given by

\[
\Delta X_t = \gamma(\theta - X_t)dt + \sigma \sqrt{X_t}dW_t
\]

and hence applying (4.6.6) to (4.13), we get

\[
\Delta f = ix^{i-1}(\gamma(\theta - x)dt + \sigma \sqrt{x}dW_t) + \\
\frac{1}{2}i(i - 1)x^{i-2}(\gamma(\theta - x)dt + \sigma \sqrt{x}dW_t)^2 + \\
\frac{1}{6}i(i - 1)(i - 2)x^{i-3}(\gamma(\theta - X_t)dt + \sigma \sqrt{x}dW_t)^3 + 
\]

proceeding similar to approach adopted in Model 1 and Model 2 above, we obtain step 2 in above equation by ignoring \( (\Delta t) \) terms of order more than 1. Now taking the fourth term \( \frac{1}{2}i(i - 1)x^{i-1}\gamma^2(dW_t)^2 \), we note the fact that \( dW_t \) has expected value 0 and variance \( \Delta t \), hence this last term is of order \( \Delta t \) and can’t be ignored. It can be proved that as \( \Delta t \to 0 \), \( (dW_t)^2 \to \Delta t \). Hence moments of \( \eta \) denoted as \( \eta^i = \mathbb{E}[X^i] \) can be found by solving the following system of ordinary differential equations

\[
\frac{d}{dt} \eta^i(t) = i\gamma\theta \eta^{i-1}(t) - i\gamma \eta^i(t) + \frac{1}{2}i(i - 1)\sigma^2 \eta^{i-2}(t), \quad 0 \leq i \leq 2r
\]

with initial conditions \( \eta^i(0) = (X_0)^i \) and can’t be ignored. It can be proved that as \( \Delta t \to 0 \), \( (dW_t)^2 \to \Delta t \). Hence moments of \( \eta \) denoted as \( \eta^i = \mathbb{E}[X^i] \) can be found by solving the following system of ordinary differential equations

\[
\frac{d}{dt} \eta^i(t) = i\gamma\theta \eta^{i-1}(t) - i\gamma \eta^i(t) + \frac{1}{2}i(i - 1)\sigma^2 \eta^{i-2}(t), \quad 0 \leq i \leq 2r
\]

**4.6 SDP Relaxations for Asian Options**

Suppose the underlying asset price process satisfies the following stochastic differential equation

\[
\frac{dX_t}{dt} = b(X_t)dt + \sigma(X_t)dW_t, \quad b, \sigma : \mathbb{R} \to \mathbb{R} \quad X_0 = x_0 \in \mathbb{R}
\]

We introduce the process \( Y_t \) defined by

\[
Y_t = \frac{1}{T} \int_0^t X_sds, \quad t \geq 0
\]

Now consider the process \( \left( \frac{X_t}{Y_t} \right) \) and the stopping time \( \tau = T \). Now if we define \( \eta \) as exit location measure, then we can see that the value of a Asian call option written on the underlying X is given by

\[
v_A(x_0) := e^{-\tau T} \mathbb{E}[(Y_T - K)^+] = \int_{\mathbb{R}} (y - K)^+ \eta_Y(dy)
\]

where \( (X_T - K)^+ \) denotes \( \max(X_T - K, 0) \), \( T > 0 \) is the option’s maturity time, \( K \) is the option’s strike price, \( \rho \) is a constant discounting factor, \( x_0 \) is the initial underlying asset price, and \( \eta_Y(dy) := \eta(\mathbb{R}, dy) \) is the \( Y \)-marginal of \( \eta \) or in other words, the distribution of random variable \( Y_T \). Under a simplification, we set the constant discounting factor \( \rho \) to 0, because its presence would not affect this analysis.

Now using the problem \( Q_{12}(Z_0) \) to solve \( v_A(x_0) \), consider the restrictions \( \eta_{Y_1} \) and \( \eta_{Y_2} \) of measure \( \eta_Y \) on the sets \( (-\infty, K) \) and \( (K, +\infty) \), respectively as shown in figure 4.2.
Now we can express the equation (4.5.2) as

\[ v_A(x_0) := \mathbb{E}[(Y_T - K)^+] = \int_{\mathbb{R}} (x - K)^+ \eta(dx) = \eta Y_2 - K \eta Y_1 \]

(4.6.4)

We hence conclude that the finite dimensional SDP problem we need to solve in order to obtain the upper and lower bounds for price of Asian call option is given by:

\[
\begin{align*}
\text{extremize} & \quad \eta Y_2 - K \eta Y_1 \\
\text{subject to} & \quad \eta Y_1 + \eta Y_2 = \eta Y, \quad j = 0, \ldots, 2r \\
& \quad M_r(\eta Y_1) \succeq 0, \quad M_{r-1}(g_1(x), \eta Y_1) \succeq 0, \\
& \quad M_r(\eta Y_2) \succeq 0, \quad M_{r-1}(g_2(x), \eta Y_2) \succeq 0,
\end{align*}
\]

(4.6.5)

where \( g_1 \) and \( g_2 \) are the polynomials defined by

\[
\begin{align*}
g_1(x) &= (K - x) \\
g_2(x) &= (x - K)
\end{align*}
\]

(4.6.6)

since \( \eta Y_1 \) and \( \eta Y_2 \) are defined on the sets \((-\infty, K] \) and \([K, +\infty) \), respectively. \( M_r(\cdot) \), \( M_{r-1}(\cdot, \cdot) \) are the moment and localizing matrices respectively. The last two constraints defining the feasible region of this SDP problem correspond to the necessary SDP moment conditions for \( \{\eta Y_1, j = 0, \ldots, 2r\} \) and \( \{\eta Y_2, j = 0, \ldots, 2r\} \) to be moments of measures supported on \((-\infty, K] \) and \([K, +\infty) \), respectively. For \( \eta Y_1 \), it is not necessary to consider moment conditions on \((-\infty, K] \) instead of \((-\infty, K) \), because we evaluate the expectation of a function which vanishes at \( x = K \).

The derivation of SDP problem for finding upper and lower bounds for price of Asian put option is very similar; only the objective function is changed in (4.6.5) which is given by \( K \eta Y_1 - \eta Y_2 \), rest of constraints remain the same.
4.6.1 Moments Computation

We assume that $X$ and $Y$ are the stochastic processes defined in previous section. Now we intend to compute the moment of $\eta$, hence we consider the monomial $f(X, Y) = X^i Y^j$ which is continuous and differentiable, hence we would apply Taylor's expansion to two variables

$$
\Delta f = \left( \frac{\partial f}{\partial x} \right) \Delta X + \left( \frac{\partial f}{\partial y} \right) \Delta Y + \frac{1}{2} \left( \frac{\partial^2 f}{\partial x^2} \right) \Delta X^2 + \frac{1}{2} \left( \frac{\partial^2 f}{\partial y^2} \right) \Delta Y^2 + \left( \frac{\partial^2 f}{\partial x \partial y} \right) \Delta X \Delta Y + \ldots
$$

(4.6.7)

to $f(X)$ for a small change $\Delta X$ in $X$ and $\Delta Y$ in $Y$. The change in variable $Y$, $\Delta Y$, is given by

$$
\Delta Y_t = \frac{1}{t} X_t dX_t
$$

(4.6.8)

We will now elaborate on following 3 cases.

**Case 1: Geometric Brownian Motion:** Here $\Delta X$ is given by

$$
\Delta X_t = bX_t dt + \sigma X_t dW_t
$$

(4.6.9)

and hence applying (4.6.9) to (4.6.7), we get

$$
\begin{align*}
\Delta f &= i x^{i-1} y^j (b x dt + \sigma x dW_t) \\
&\quad + j x^i y^{j-1} \left( \frac{1}{T} x dt \right) \\
&\quad + \frac{1}{2} i (i - 1) x^{i-2} y^j (b x dt + \sigma x dW_t)^2 \\
&\quad + \frac{1}{2} j x^i y^{j-1} (b x dt + \sigma x dW_t) \left( \frac{1}{T} x dt \right) \\
&\quad + j (j - 1) x^i y^{j-2} \left( \frac{1}{T} x dt \right)^2 + \ldots \\
&= ib x^i y^j dt + i \sigma x^i y^j dW_t + \frac{1}{T} j x^{i+1} y^{j-1} dt + \frac{1}{2} i (i - 1) \sigma^2 x^i y^j (dW_t)^2
\end{align*}
$$

(4.6.10)

where step 2 in above equation is obtained by ignoring $(\Delta t)$ terms of order more than 1. Now taking the fourth term $\frac{1}{2} i (i - 1) \sigma^2 x^i y^j (dW_t)^2$, we note the fact that $dW_t$ has expected value 0 and variance $\Delta t$, hence this last term is of order $\Delta t$ and can’t be ignored. It can be proved that as $\Delta t \to 0$, $(dW_t)^2 \to \Delta t$. Hence moments of $\eta$ denoted as $\eta^{i,j} = E[X^i Y^j]$ can be found by solving the following system of ordinary differential equations

$$
\frac{d}{dt} \eta^{i,j}(t) = ib \eta^{i,j}(t) + \frac{1}{T} j \eta^{i+1,j-1}(t) + \frac{1}{2} i (i - 1) \sigma^2 \eta^{i,j}(t), \quad 0 \leq i \leq 2r
$$

(4.6.11)

with initial conditions $\eta^{0,j}(0) = 0, j \neq 0$, $\eta^{0,0}(0) = 1$, $\eta^{1,0}(0) = (X_0)^i$ and the second term $i \sigma x^i y^j dW_t$ vanishes because $\mathbb{E}[dW_t] = 0$.

**Case 2: Ornstein-Uhlenbeck Process:** Here $\Delta X$ is given by

$$
\Delta X_t = \gamma (\theta - X_t) dt + \sigma dW_t
$$

(4.6.12)
and hence applying (4.6.9) to (4.6.7), we get

$$\triangle f = ix^{i-1}y^j(\gamma(x-t)dt + \sigma dW_t) + jx^iy^{j-1}\left(\frac{1}{T}xdt\right) + \frac{1}{2}j(i-1)x^{i-2}y^j(\gamma(x-t)dt + \sigma dW_t)^2 + \frac{1}{2}jix^{i-1}y^{j-1}(\gamma(x-t)dt + \sigma dW_t)\left(\frac{1}{T}xdt\right) + j(j-1)x^{i-1}y^{j-2}\left(\frac{1}{T}xdt\right)^2 + ...$$

(4.6.13)

where similar analysis applies as we did for Model 1 (geometric Brownian motion). Specifically, step 2 in above equation is obtained by ignoring ($\triangle t$) terms of order more than 1. Now taking the fifth term $\frac{1}{2}j(i-1)\sigma^2x^iy^j(dW_t)^2$, we note the fact that $dW_t$ has expected value 0 and variance $\triangle t$, hence this last term is of order $\triangle t$ and can’t be ignored. It can be proved that as $\triangle t \to 0$, $(dW_t)^2 \to \triangle t$. Hence moments of $\eta$ denoted as $\eta^{i,j} = E[X^iY^j]$ can be found by solving the following system of ordinary differential equations

$$\frac{d}{dt}\eta^{i,j}(t) = i\gamma\eta^{i-1,j}(t) - i\gamma\eta^{i,j}(t) + \frac{1}{T}j\eta^{i+1,j-1}(t) + \frac{1}{2}j(i-1)\sigma^2\eta^{i,j}(t), \quad 0 \leq i \leq 2r$$

(4.6.14)

with initial conditions$^4$ $\eta^{i,0}(0) = 0$, $j \neq 0$, $\eta^{0,0}(0) = 1$, $\eta^{i,0}(0) = (X_0)^i$ and the third term $i\sigma x^{i-1}y^j(dW_t)$ vanishes because $E[dW_t] = 0$.

**Case 3: Cox-Ingersoll-Ross Process:** Here $\triangle X$ is given by

$$\triangle X_t = \gamma(x, X_t)dt + \sigma \sqrt{X_t}dW_t$$

(4.6.15)

and hence applying (4.6.9) to (4.6.7), we get

$$\triangle f = ix^{i-1}y^j(\gamma(x-t)dt + \sigma \sqrt{X_t}dW_t) + jx^iy^{j-1}\left(\frac{1}{T}xdt\right) + \frac{1}{2}j(i-1)x^{i-2}y^j(\gamma(x-t)dt + \sigma \sqrt{X_t}dW_t)^2 + \frac{1}{2}jix^{i-1}y^{j-1}(\gamma(x-t)dt + \sigma \sqrt{X_t}dW_t)\left(\frac{1}{T}xdt\right) + j(j-1)x^{i-1}y^{j-2}\left(\frac{1}{T}xdt\right)^2 + ...$$

(4.6.16)

proceeding similar to approach adopted in Model 1 and Model 2 above, we obtain step 2 in above equation by ignoring ($\triangle t$) terms of order more than 1. Now taking the fifth term $\frac{1}{2}j(i-1)\sigma^2x^{i-1}y^j(dW_t)^2$, we note the fact that $dW_t$ has expected value 0 and variance $\triangle t$, hence this

$^4$In Lasserre, Prieto-Rumeau and Zervos in [21], there is a minor error in specifying the initial conditions for this ODE, which we have corrected here.
last term is of order $\Delta t$ and can’t be ignored. It can be proved that as $\Delta t \to 0$, $(dW_t)^2 \to \Delta t$.
Hence moments of $\eta$ denoted as $\eta^{ij} = \mathbb{E}[X^iY^j]$ can be found by solving the following system of ordinary differential equations

$$\frac{d}{dt} \eta^{ij}(t) = i\gamma \eta^{i-1,j}(t) - \frac{1}{2} j \eta^{i,j-1}(t) + \frac{1}{2} i(i-1) \sigma^2 \eta^{i-1,j}(t), \quad 0 \leq i \leq 2r$$

with initial conditions $\eta^{ij}(0) = 0, j \neq 0, \quad \eta^{00}(0) = 1, \quad \eta^{i0}(0) = (X_0)^i$ and the third term $i\sigma x^i \frac{1}{2} \gamma y^j dW_t$ vanishes because $\mathbb{E}[dW_t] = 0$.

### 4.7 SDP Relaxations for Barrier Options

In this section, we focus on applying theory of moments and SDP relaxations for three types of Barrier Options. Specifically we develop the three separate cases for down-and-out, up-and-out and double Barrier options in sections 4.7.1, 4.7.2 and 4.7.3 respectively. Subsequently, in section 4.7.4, we deal with the issue of deriving explicit expression for basis adjoint equation for each of the three models mentioned in section 2.2.

#### 4.7.1 Down-and-out Barrier Options

Suppose the underlying asset price process satisfies the following stochastic differential equation

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t, \quad b, \sigma : \mathbb{R} \to \mathbb{R} \quad X_0 = x_0 \in \mathbb{R}$$

(4.7.1)

We introduce the process $Y_t = t$, the process $Z_t = \begin{pmatrix} X_t \\ Y_t \end{pmatrix}$ and the stopping time $\tau$ defined as

$$\tau = \min \{ t \geq 0 \mid X_t \leq H_L \} \wedge T$$

(4.7.2)

which means that stopping time is that time when price of underlying falls below the lower barrier $H_L$ for the first time or it is $T$, the expiration time, if the price of underlying does not fall below the lower barrier. Now if we define $\eta$ as exit location measure and $\mu$ as expected occupation measure which are defined on $[0, T] \times (H_L, +\infty)$ and $[0, T] \times (H_L, K) \cup (T) \times (H_L, +\infty)$, then we can see that the value of a Barrier call option written on the underlying $X$ is given by

$$v_B(x_0) := e^{\rho T} \mathbb{E} \left[ (X_T - K)^+ I_{(\tau = T)} \right] = \int_{[0,T] \times \mathbb{R}} (x - K)^+ \eta(dt, dx)$$

(4.7.3)

where $(x - K)^+$ denotes $\max(x - K, 0)$, $T > 0$ is the option’s maturity time, $K$ is the option’s strike price, $\rho$ is a constant discounting factor and $x_0$ is the initial underlying asset price. Under a simplification, we set the constant discounting factor $\rho$ to 0, because its presence would not affect this analysis.

Introducing a significant computational simplification, we separate the measure $\eta$ into the sum of three measures $\eta_1, \eta_2$ and $\eta_3$, supported on $[0, T] \times \{H_L\}, \{T\} \times (H_L, K)$ and $\{T\} \times [K, +\infty)$ respectively, as shown in the figure 4.3. This enables us to express $\eta$ as a linear combination of $\eta_1, \eta_2$ and $\eta_3$. In order to understand this better, we can assume that the measures $\eta_1, \eta_2$ and $\eta_3$ are supported on the subsets of the real line $[0, T], (H_L, K)$ and $[K, +\infty)$, respectively.

Now using the problem $Q^\mu (Z_0)$ to solve $v_B(x_0)$, we can express the equation (4.7.3) as

$$v_B(x_0) := \mathbb{E} \left[ (X_T - K)^+ I_{(\tau = T)} \right] = \int_{[0,T] \times \mathbb{R}} (x - K)^+ \eta(dt, dx) = \eta_3^1 - K \eta_3^0$$

(4.7.4)
4.7 SDP Relaxations for Barrier Options

We hence conclude that the finite dimensional SDP problem we need to solve to obtain the upper and lower bounds for price of down-and-out Barrier call option is given by:

\[
\begin{align*}
\text{extremize} & \quad \eta_1^1 - K\eta_3^0 \\
\text{subject to} & \quad \eta^{i,j} - z_0^{i,j} - \sum_{\beta} c_\beta(i,j)\mu^{i,j} = 0, \quad 0 \leq i + j \leq 2r \\
& \quad M_r(\eta_1) \succeq 0, \quad M_{r-1}(g_1(t), \eta_1) \succeq 0, \\
& \quad M_r(\eta_2) \succeq 0, \quad M_{r-1}(g_2(x), \eta_2) \succeq 0, \\
& \quad M_r(\eta_3) \succeq 0, \quad M_{r-1}(g_3(x), \eta_3) \succeq 0, \\
& \quad M_r(\mu) \succeq 0, \quad M_{r-1}(g_4(t,x), \mu) \succeq 0, \quad M_{r-1}(g_5(t,x), \mu) \succeq 0,
\end{align*}
\] (4.7.5)

where\(^5\)

\[
\begin{align*}
g_1(t) &= (T - t)t \\
g_2(x) &= (K - x)(x - H_L) \\
g_3(x) &= x - K \\
g_4(t,x) &= (T - t)t \\
g_5(t,x) &= x - H_L
\end{align*}
\] (4.7.6)

and \(M_r(\cdot)\), \(M_{r-1}(\cdot, \cdot)\) are the moment and localizing matrices respectively. The last four constraints defining the feasible region of this SDP problem correspond to the necessary SDP moment conditions for \(\{\eta_1^j, j = 0, \ldots, 2r\}\), \(\{\eta_2^j, j = 0, \ldots, 2r\}\), \(\{\eta_3^j, j = 0, \ldots, 2r\}\) and \(\{\mu^{i,j}, 0 \leq i + j \leq 2r\}\) to be moments of measures supported on \([0, T], [H_L, K], [K, +\infty)\) and \([0, T] \times [H_L, +\infty)\), respectively. For \(\eta_2\), it is not necessary to consider moment conditions on \([H_L, K]\) instead of \((H_L, K)\), because we evaluate the expectation of a function which vanishes at both \(x = K\) and \(x = H_L\).

\(^5\)In Lasserre, Prieto-Rumeau and Zervos in [21], there is a minor error in specifying the function \(g_5(t,x)\) which we have corrected here.

Figure 4.3: Definition of Measures for Down-And-Out Barrier Options

We hence conclude that the finite dimensional SDP problem we need to solve to obtain the upper and lower bounds for price of down-and-out Barrier call option is given by:
We introduce the process \( Y_t \) Suppose the underlying asset price process satisfies the stochastic differential equation given in (4.7.2) and rest of constraints remain the same.

Barrier put option is very similar; only the objective function is changed in (4.7.5) which is given as below:

\[
v^P_B(x_0) := \mathbb{E} \left[ (K - X_T)^+ I_{(\tau = T)} \right] = \int_{[0,T] \times \mathbb{R}} (K - x)^+ \eta(dt, dx) = \eta_2^{T} - \eta_2^{0}
\]

and rest of constraints remain the same.

4.7.2 Up-and-out Barrier Options

Suppose the underlying asset price process satisfies the stochastic differential equation given in (4.7.1). We introduce the process \( Y_t = t \), the process \( Z_t = \begin{pmatrix} X_t \\ Y_t \end{pmatrix} \) and the stopping time \( \tau \) defined as

\[
\tau = \min \{ t \geq 0 \ | \ X_t \geq H_U \} \wedge T
\]

which means that stopping time is that time when price of underlying raises above the upper barrier \( H_U \) for the first time or it is \( T \), the expiration time, if the price of underlying does not raise above the upper barrier. Now if we define \( \eta \) as exit location measure and \( \mu \) as expected occupation measure which are defined on \([0,T] \times [0,H_U] \) and \([0,T] \times \{ H_U \} \cup \{ T \} \times [0,H_U] \), then we can see that the value of a Barrier call option written on the underlying \( X \) is given by

\[
v_B(x_0) := e^{\rho T} \mathbb{E} \left[ (X_T - K)^+ I_{(\tau = T)} \right] = \int_{[0,T] \times \mathbb{R}} (x - K)^+ \eta(dt, dx)
\]

where \( (x - K)^+ \) denotes \( \max(x - K, 0) \), \( T > 0 \) is the option’s maturity time, \( K \) is the option’s strike price, \( \rho \) is a constant discounting factor and \( x_0 \) is the initial underlying asset price. Under a simplification, we set the constant discounting factor \( \rho \) to 0, because its presence would not affect this analysis.

As we did for the down-and-out Barrier options, we introduce a significant computational simplification, by separating the measure \( \eta \) into the sum of three measures \( \eta_1, \eta_2 \) and \( \eta_3 \), supported on \( \{ T \} \times [0,K), \{ T \} \times (K,H_U) \) and \([0,T] \times \{ H_U \} \) respectively, as shown in the figure 4.4. This enables us to express \( \eta \) as a linear combination of \( \eta_1, \eta_2 \) and \( \eta_3 \). In order to understand this better, we can assume that the measures \( \eta_1, \eta_2 \) and \( \eta_3 \) are supported on the subsets of the real line \( [0,K), (K,H_U) \) and \([0,T] \), respectively.

Now using the problem \( Q^{UI}_r(Z_0) \) to solve \( v_B(x_0) \), we can express the equation (4.7.9) as

\[
v_B(x_0) := \mathbb{E} \left[ (X_T - K)^+ I_{(\tau = T)} \right] = \int_{[0,T] \times \mathbb{R}} (x - K)^+ \eta(dt, dx) = \eta_2^{T} - \eta_2^{0}
\]

We hence conclude that the finite dimensional SDP problem we need to solve to obtain the upper and lower bounds for price of up-and-out Barrier call option is given by:

**extremize** \( \eta_2^{T} - \eta_2^{0} \)

**subject to** \( \eta^{i,j} - z_0^{i,j} - \sum_{\beta} c_\beta(i,j) \mu^{\beta} = 0, \ 0 \leq i + j \leq 2r \)

\[
M_r(\eta_1) \geq 0, \ M_{r-1}(g_1(t,x), \eta_1) \geq 0, \ (4.7.11)
\]

\[
M_r(\eta_2) \geq 0, \ M_{r-1}(g_2(t,x), \eta_2) \geq 0,
\]

\[
M_r(\eta_3) \geq 0, \ M_{r-1}(g_3(t,x), \eta_3) \geq 0,
\]

\[
M_r(\mu) \geq 0, \ M_{r-1}(g_4(t,x), \mu) \geq 0, \ M_{r-1}(g_5(t,x), \mu) \geq 0,
\]
where

\[ g_1(x) = (K - x)x \]
\[ g_2(x) = (H_U - x)(x - K) \]
\[ g_3(t) = (T - t)t \]
\[ g_4(t, x) = (T - t)t \]
\[ g_5(t, x) = (H_U - x)x \]

(4.7.12)

and \( M_r(.,.) \), \( M_{r-1}(.,.) \) are the moment and localizing matrices respectively. The last four constraints defining the feasible region of this SDP problem correspond to the necessary SDP moment conditions for \( \{ \eta_1^1, j = 0, ..., 2r \} \), \( \{ \eta_1^2, j = 0, ..., 2r \} \), \( \{ \eta_1^3, j = 0, ..., 2r \} \) and \( \{ \mu^i, j \leq i + j \leq 2r \} \) to be moments of measures supported on \([0, K], [K, H_U]\) and \([0, T]\) and \([0, T] \times [0, H_U]\), respectively. Note that for \( \eta_1 \) or \( \eta_2 \), we are not restricted to consider intervals which exclude \( H_U \) or \( K \), because we are evaluating the expectation of a function which vanishes at \( x = H_U \) and \( x = K \).

The derivation of SDP problem for finding upper and lower bounds for price of up-and-out Barrier put option is very similar; only the objective function is changed in (4.7.11) which is given as below:

\[ v_B^{up}(x_0) := \mathbb{E} \left[ (K - X_T)^+ I_{\{T = T\}} \right] = \int_{[0, T] \times \mathbb{R}} (K - x)^+ \eta(dt, dx) = \eta_1^1 - K \eta_1^0 \]

(4.7.13)

and rest of constraints remain the same.

### 4.7.3 Double Barrier Options

Suppose the underlying asset price process satisfies the stochastic differential equation given in (4.7.1). Again as in previous two sections, we introduce the process \( Y_t = t \), the process...
\[ Z_t = \left( \begin{array}{c} X_t \\ Y_t \end{array} \right) \] and the stopping time \( \tau \) defined as
\[
\tau = \min \{ t \geq 0 \mid X_t \leq H_L \land X_T \geq H_U \} \land T
\] (4.7.14)
which means that stopping time is that time when price of underlying falls below the lower barrier \( H_L \) or raises above the upper barrier \( H_U \) for the first time or it is \( T \), the expiration time, if the price of underlying does not fall below lower barrier or raise above the upper barrier. Now if we define \( \eta \) as exit location measure and \( \mu \) as expected occupation measure which are defined on \([0,T] \times (H_L, H_U)\) and \([0,T] \times \{ H_L \} \cup [0,T] \times \{ H_U \} \cup \{ T \} \times [0, H_U)\), then we can see that the value of a Barrier call option written on the underlying \( X \) is given by
\[
v_B(x_0) := e^{\rho T} E \left[ (X_T - K)^+ I_{\{ \tau = T \}} \right] = \int_{[0,T] \times \mathbb{R}} (x - K)^+ \eta(dt, dx)
\] (4.7.15)
where \((x - K)^+ \) denotes \(\max(x - K, 0)\), \( T > 0 \) is the option’s maturity time, \( K \) is the option’s strike price, \( \rho \) is a constant discounting factor and \( x_0 \) is the initial underlying asset price. Under a simplification, we set the constant discounting factor \( \rho \) to 0, because its presence would not affect this analysis.

As we did for the down-and-out and up-and-out Barrier options, we again introduce a significant computational simplification by separating measure \( \eta \) into the sum of four measures \( \eta_1, \eta_2, \eta_3 \) and \( \eta_4 \), supported on \([0,T] \times \{ H_L \}, [0,T] \times \{ H_U \}, \{ T \} \times (H_L, K)\) and \( \{ T \} \times [K, H_U)\) respectively, as shown in the figure 4.5. This enables us to express \( \eta \) as a linear combination of \( \eta_1, \eta_2, \eta_3 \) and \( \eta_4 \). In order to understand this better, we can assume that the measures \( \eta_1, \eta_2, \eta_3 \) and \( \eta_4 \) are supported on the subsets of the real line \([0,T], [0,T], (H_L, K)\) and \([K, H_U)\), respectively.

![Figure 4.5: Definition of Measures for Double Barrier Options](image)

Now using the problem \( Q^{II}_r(Z_0) \) to solve \( v_B(x_0) \), we can express the equation (4.7.15) as
\[
v_B(x_0) := E \left[ (X_T - K)^+ I_{\{ \tau = T \}} \right] = \int_{[0,T] \times \mathbb{R}} (x - K)^+ \eta(dt, dx) = \eta_1^1 - K \eta_4^0
\] (4.7.16)
We hence conclude that the finite dimensional SDP problem we need to solve to obtain the upper and lower bounds for price of double Barrier call option is given by:

\[
\begin{align*}
\text{extremize} & \quad \eta_1^L - K\eta_1^U \\
\text{subject to} & \quad \eta^{i,j} - z_0^{i,j} - \sum_{\beta} c_{\beta}(i,j)\mu^{i,j} = 0, \quad 0 \leq i + j \leq 2r \\
& \quad M_r(\eta_1) \geq 0, \quad M_{r-1}(g_1(t), \eta_1) \geq 0, \\
& \quad M_r(\eta_2) \geq 0, \quad M_{r-1}(g_2(t), \eta_2) \geq 0, \\
& \quad M_r(\eta_3) \geq 0, \quad M_{r-1}(g_3(x), \eta_3) \geq 0, \\
& \quad M_r(\mu) \geq 0, \quad M_{r-1}(g_5(t,x), \mu) \geq 0, \\
& \quad M_{r-1}(g_6(t,x), \mu) \geq 0, \quad M_{r-1}(g_6(t,x), \mu) \geq 0,
\end{align*}
\]

where

\[
\begin{align*}
g_1(t) &= (T-t)T \\
g_2(t) &= (T-t)T \\
g_3(x) &= (K-x)(x - H_L) \\
g_4(x) &= (H_U - x)(x - K) \\
g_5(t,x) &= (T-t)T \\
g_6(t,x) &= (H_U - x)(x - H_L)
\end{align*}
\]

and \(M_r(\cdot), M_{r-1}(\cdot, \cdot)\) are the moment and localizing matrices respectively. The last four constraints defining the feasible region of this SDP problem correspond to the necessary SDP moment conditions for \(\{\eta_1^L, j = 0, \ldots, 2r\}, \{\eta_2^L, j = 0, \ldots, 2r\}, \{\eta_3^L, j = 0, \ldots, 2r\}\) and \(\{\mu^{i,j}, 0 \leq i + j \leq 2r\}\) to be moments of measures supported on \([0, T], [0, T], [H_L, K]\) and \([K, H_U]\) and \(T \times (H_L, H_U)\) respectively. Note that for \(\eta_3\) or \(\eta_4\), we are not restricted to consider intervals which exclude \(H_U, H_L\) or \(K\), because we are evaluating the expectation of a function which vanishes at all of following values: \(x = H_U, x = H_L\) and \(x = K\).

The derivation of SDP problem for finding upper and lower bounds for price of double Barrier option is very similar; only the objective function is changed in (4.7.17) which is given as below:

\[
v_B^U(x_0) := \mathbb{E} [(K - X_T)^+ I_{\{\tau = T\}}] = \int_{[0,T] \times \mathbb{R}} (K-x)^+ \eta(dt, dx) = \eta_3^1 - K\eta_3^0
\]

and rest of constraints remain the same.

### 4.7.4 Evaluating Basic Adjoint Equation

We will now derive an explicit expression for evaluating the basis adjoint equation mentioned in SDP problems (4.7.5), (4.7.11) and (4.7.17).

\[
\eta^{i,j} - z_0^{i,j} - \sum_{\beta} c_{\beta}(i,j)\mu^{i,j} = 0, \quad 0 \leq i + j \leq 2r
\]

To calculate \(\eta^{i,j}\) for each of down-and-out, up-and-out and double Barrier options, consider the monomial \(f(t,x) = t^ix^j\),

\[
\eta^{i,j} = \int_{\mathbb{R}^2} f(z)\eta(dz) = \int_{\mathbb{R}^2} f(t,x)\eta(dt, dx) = \int_{\mathbb{R}^2} t^ix^j\eta(dt, dx)
\]

which can be evaluated as below for each of the three type of Barrier options.
Case 1: Down-and-out Barrier Options
\[ \eta_{i,j} = \int_{\mathbb{R}^2} t^i x^j \eta(dt, dx) \]
\[ = H_L^j \int_0^T t^i \eta_1(dt) + T^i \int_{H_L}^K x^j \eta_2(dx) + \int_{H}^K T^i x^j \eta_3(dx) \]
\[ = H_L^j \eta_1^i + T^i \eta_2^j + T^i \eta_3^j \]  
\hspace{4cm} (4.7.22)

Case 2: Up-and-out Barrier Options
\[ \eta_{i,j} = \int_{\mathbb{R}^2} t^i x^j \eta(dt, dx) \]
\[ = T^i \int_0^K x^j \eta_1(dx) + T^i \int_{K}^{H_U} x^j \eta_2(dx) + H_U^j \int_0^T t^i \eta_3(dt) \]
\[ = T^i \eta_1^i + T^i \eta_2^j + H_U^j \eta_3^j \]  
\hspace{4cm} (4.7.23)

Case 3: Double Barrier Options
\[ \eta_{i,j} = \int_{\mathbb{R}^2} t^i x^j \eta(dt, dx) \]
\[ = H_L^j \int_0^T t^i \eta_1(dt) + H_U^j \int_0^T t^i \eta_2(dt) \]
\[ + T^i \int_{H_L}^K x^j \eta_3(dx) + \int_{H}^K T^i x^j \eta_4(dx) \]
\[ = H_L^j \eta_1^i + H_U^j \eta_2^i + T^i \eta_3^j + T^i \eta_4^j \]  
\hspace{4cm} (4.7.24)

We now turn our attention to evaluating the following term
\[ \sum_{\beta} c_{\beta}(i,j) \mu_{i,j} = \int_{\mathbb{R}^n} \mathcal{A}f(z) \mu(dz) \]  
\hspace{4cm} (4.7.25)
which depends on infinitesimal generator
\[ \mathcal{A}f(t, x) := f_t(t, x) + \frac{1}{2} \sigma^2(x) f_{xx}(t, x) + b(x) f_x(t, x) \]  
\hspace{4cm} (4.7.26)
of particular process. Considering the following three cases:

Case 1: Geometric Brownian Motion: Here the underlying \(X_t\) satisfies the following stochastic differential equation:
\[ dX_t = bX_t dt + \sigma X_t dW_t \]  
\hspace{4cm} (4.7.27)
So the infinitesimal generator can be derived as
\[ \mathcal{A}f(t, x) := it^{i-1}x^j + \frac{1}{2} \sigma^2(j (j - 1)) t^i x^{j-2} + b dt \]  
\hspace{4cm} (4.7.28)
and the coefficients (4.7.25) can be calculated from (4.7.26) as below
\[ \sum_{\beta} c_{\beta}(i,j) \mu_{i,j} = it^{i-1} + \frac{1}{2} \sigma^2(j (j - 1)) \mu_{i,j} + b \mu_{i,j} \]  
\hspace{4cm} (4.7.29)

Case 2: Ornstein-Uhlenbeck Process: Here the underlying \(X_t\) satisfies the following stochastic differential equation:
\[ dX_t = \gamma(\theta - X_t) dt + \sigma dW_t \]  
\hspace{4cm} (4.7.30)
So the infinitesimal generator can be derived as
\[
Af(t, x) := it^{i-1}x^j + \frac{1}{2}\sigma^2 j(j-1)t^ix^{j-2} + \gamma\theta jt^ix^{j-1} - \gamma j t^ix^j
\] (4.7.31)
and the coefficients (4.7.25) can be calculated from (4.7.26) as below
\[
\sum_{\beta} c_{\beta}(i, j)\mu^{i,j} = i\mu^{i-1,j} + \frac{1}{2}\sigma^2 j(j-1)\mu^{i,j-2} + \gamma\theta j \mu^{i,j-1} - \gamma j \mu^{i,j}
\] (4.7.32)

**Case 3: Cox-Ingersoll-Ross Process:** Here the underlying \( X_t \) satisfies the following stochastic differential equation:
\[
dX_t = \gamma(\theta - X_t)dt + \sigma\sqrt{X_t}dW_t
\] (4.7.33)
So the infinitesimal generator can be derived as
\[
Af(t, x) := it^{i-1}x^j + \frac{1}{2}(\sigma\sqrt{x})^2 j(j-1)t^ix^{j-2} + \gamma\theta jt^ix^{j-1} - \gamma j t^ix^j
\] (4.7.34)
and the coefficients (4.7.25) can be calculated from (4.7.26) as below
\[
\sum_{\beta} c_{\beta}(i, j)\mu^{i,j} = i\mu^{i-1,j} + \frac{1}{2}\sigma^2 j(j-1)\mu^{i,j-1} + \gamma\theta j \mu^{i,j-1} - \gamma j \mu^{i,j}
\] (4.7.35)
Chapter 5

Numerical Results and Analysis

In this section, we present numerical results obtained for the option pricing problems considered in last two chapters: using semi-parametric approach developed in chapter 3 and SDP moment approach developed in chapter 4. Although fastest implementation of the above approaches would best be done in lower level languages such as C/C++ (and probably Java), the platform of our choice for implementation was MATLAB [23] as it enables developers in quick prototyping by focusing on higher level constructs. Hence, as we focused on understanding and formulating the SDP problems in this project rather than best and most efficient possible ways of implementing those problems (which is another project in itself), the choice of MATLAB as a quick prototyping tool was quite right.

To solve the resulting SDP problems, we used SeDuMi [16] and YALMIP [13] (which acts as an interface to SeDuMi). SeDuMi is a MATLAB software which is designed to solve optimization problems over symmetric cones. The symmetric cones can include linear, quadratic, second order conic and semidefinite optimization, and any combination of these. The language chosen for modelling the SDP programs was YALMIP, which is a MATLAB toolbox designed for solving advanced optimization problems of various types such as linear, quadratic, second order cone, semidefinite, mixed integer conic, geometric, local and global polynomial, multi-parametric and robust programming. YALMIP focuses on the higher level algorithms and internally uses several solvers to solve the optimization problems, and SeDuMi is one of solvers used for solving semidefinite programming problems.

Before we analyse the numerical results, we wish to note that the majority of results for both of above methodologies were precise and encouraging. However, it is also noted that there were numerical problems for certain cases (explained in the subsequent discussions below), and majority of such problems were caused because SeDuMi failed to achieve desired precision (usually of the order $10^{-6}$). This is in conjunction with the fact that for SDP moment approach in chapter 4, authors also faced similar issues in their implementation in [21], (their implementation also used SeDuMi internally for solving SDP problems) and concluded that such errors could be due to the fact that SeDuMi does not perform well with sparse and degenerate optimization problems.

The machine configuration on which all numerical computations were conducted is given as below: Processor: Intel(R) Core(TM)2 Duo CPU, T7100, 1.80 GHz, Memory(RAM): 2038 MB, 32-bit Operating System running Windows Vista Business.

5.1 SDP Moment Approach for Pricing Exotic Options

In this section we will cover the numerical results for pricing European, Asian and (down-and-out, up-and-out and double) Barrier options with SDP Moment approach suggested by Lasserre, Prieto-Rumeau and Zervos in [21].
5.1 SDP Moment Approach for Pricing Exotic Options

5.1.1 European Options

Geometric Brownian Motion
Table A.2 in Appendix A shows the discounted values of European call options (with underlying process following a geometric Brownian motion), obtained by solving SDP problem (4.5.4), for different values of volatility $\sigma$. We also compare the call option values thus obtained with the Black Scholes value because we have an explicit formula available. The data set used for the problem is also mentioned in the table. We note the following points regarding the results:

- The accuracy of results increase with increase in number or relaxations, when other parameters are kept constant. This is expected behaviour because we are adding more information about higher order moments to the SDP problems.

- In general, the accuracy of results is very high with very few relaxations. For example, with $\sigma = 0.05$ and $R=4$ to 8 relaxations, the relative error as defined by

$$\text{Error} = \frac{2 \cdot \text{Upper Bound} - \text{Lower Bound}}{\text{Upper Bound} + \text{Lower Bound}}$$

is only of order $\sim 10^{-6}\%$.

- The degree of improvement in bounds is not uniform and depends on values of volatility $\sigma$, maturity time $T$ and strike price $K$. For example, as we increase the volatility $\sigma$ from 0.05 to 0.25 and keep number of relaxations $R$ constant, we note that distance between upper and lower bounds increases. Although not shown in the data, this increase would also be evident if we keep all other parameter values constant and increase the maturity time $T$.

- Black Scholes value always remains between the upper and lower bound, but the distance between Black Scholes value and the bounds decreases in general as we increase the number of relaxations $R$ (with constant volatility $\sigma$).

- For $\sigma = 0.15$ to 0.25, we observe that the value of upper and lower bounds are diverging, when we increase relaxations $R$ from 7 to 8. This can attributed to following two facts:
  - Lognormal distribution is not moment determinate, and
  - SeDuMi failed in solving the SDP problems with desired accuracy. The results were accurate with precision of $10^{-3}$, but SeDuMi failed to achieve the desired precision of $10^{-6}$.

- It is worth noting the fact that with $R=8$, the SDP problem has two semidefinite (moment) matrices of size $9 \times 9$ and two semidefinite (localizing) matrices of size $8 \times 8$ amongst its constraints, which roughly translates to 290 variables.

- The failure of SeDuMi to solve SDP problems with semidefinite matrices of higher order (of size $8 \times 8$ in our case), is in conjunction with the results obtained by Lasserre, Prieto-Rumeau and Zervos in [21]. We also note that in the semi-parametric approach, authors used a Cutting Plane algorithm developed in their article [8]. This algorithm also failed when the order of semidefinite matrix was over 9 or 10.

We now present the same data with different interpretation as below. The graphs in figure 5.2 show the difference between bounds of higher order moments and Black-Scholes price. There are two graphs, in order to depict the effect of

- adding higher order moments while keeping volatility $\sigma$ constant, and...
5 Numerical Results and Analysis

(a) Volatility=0.05

(b) Volatility=0.25

Figure 5.1: European Call Option: Geometric Brownian Motion

- increasing the volatility $\sigma$ with the same amount of moment information (i.e. keeping $R$ constant).

We observe the following from figure 5.2(a):

- As we increase the number of relaxations, difference between upper/lower bound and Black Scholes value decreases.
- Difference between upper bound and Black-Scholes value is lesser than difference between
lower bound and Black-Scholes value.

- Discrepancy in upper and lower bound increases as strike price $K$ approaches the initial stock price $S_0$, whereas same discrepancy is much less when strike price $K$ is far from initial stock price $S_0$.

Observations from figure 5.2(b):

- Upper bound is relatively less sensitive to change in volatility $\sigma$ as compared to lower bound, which shows more discrepancy.

- As we increase the volatility $\sigma$, difference between upper/lower bound and Black Scholes value decreases.

- Discrepancy in upper and lower bound is highest when value of strike price $K$ approaches the initial stock price $S_0$, whereas same discrepancy is much less when strike price $K$ is far from initial stock price $S_0$. 

5 Numerical Results and Analysis

(a) Difference between bounds of higher order moments and B-S price

(b) Difference between bounds and B-S price for different volatilities

Figure 5.2: European Call Option: Geometric Brownian Motion
Ornstein-Uhlenbeck Process

We now focus our attention on European call options when underlying process follows a Ornstein-Uhlenbeck process. Table A.3 in Appendix A shows values of European call options obtained by solving SDP problem (4.5.4), for different values of volatility $\sigma$. Please note that these values are not discounted and the discounting factor $e^{-\rho T}$ has been dropped. We also compare the call option values thus obtained with the Monte Carlo simulation values. The data set used for the problem is also mentioned in the table. Observation from the results can be summarised as below:

- Monte Carlo values are always between the upper and lower bounds for all values of volatility. Also as we increase the volatility $\sigma$ the distance between the bounds increase.

- For small volatilities ($\sigma = 0.05$ to $0.10$), $R=3$ relaxations are sufficient to obtain sharp bounds with relative error $\leq 1\%$. As we increase volatility, we need higher moments information in order to decrease the relative error.
Figure 5.3: European Call Option: Ornstein-Uhlenbeck Process
5.1 SDP Moment Approach for Pricing Exotic Options

Cox-Ingersoll-Ross Process

Finally we will observe the case for European call options when underlying process follows a Cox-Ingersoll-Ross process. The numerical results are summarized in Table A.4 in Appendix A, which shows values of European call options obtained by solving SDP problem (4.5.4). The values are not discounted as discounting factor $e^{-\rho T}$ has been dropped. Interesting observations are summarised below:

- Very accurate results are achieved when volatility $\sigma$ is small, when $R=3$. For example with $\sigma = 0.05$ or 0.10, the relative error is less than 1 %.
- Monte Carlo values are bounded by the upper and lower bounds in all cases, but the distance between upper lower bound and between bounds and Monte Carlo values increase with increase in volatility.

Finally, it can be noted from the table A.2, A.3 and A.4 that in terms of performance, the SDP algorithm for pricing European options is very fast and excellent bounds were obtained for all relaxations well within 1 or 2 seconds.
5 Numerical Results and Analysis

(a) Volatility=0.05

(b) Volatility=0.25

Figure 5.4: European Call Option: Cox Ingersoll Ross Process
5.1.2 Asian Options

We will now observe the numerical results for Asian options obtained by solving SDP problem (4.6.5) for three different kinds of processes followed by underlying asset price. In all the three cases we note the following common observation:

- As expected and as mentioned in previous section for European options, the results become more accurate on increasing the relaxations, as we add more moment information to the SDP problems.

- The degree of improvement in bounds is not uniform and depends on multiple factors such as values of volatility $\sigma$, maturity time $T$ and strike price $K$. For example, as we increase the volatility $\sigma$ in all three cases in this section and keep number of relaxations $R$ constant, we note that distance between upper and lower bounds increases.

**Geometric Brownian Motion**

Table A.5 in Appendix A shows discounted values of Asian call options obtained by solving SDP problem (4.6.5), for different values of volatility $\sigma$. The underlying asset price process is modeled by Geometric Brownian motion. Monte Carlo simulation values are given for comparison along with 95% confidence interval. Observations from the results can be summarised as below:

- In this case very accurate result is obtained in 2nd relaxation itself for $\sigma = 0.05$. The relative error in this case, as defined by (5.1.1) is only of the order of 0.08%, which decreases as we increase the number of relaxations.

- The bounds are obtained in real time, if we consider up to 4th order relaxation, but when 5th order relaxation is used the SDP program takes 20 seconds. This is still very less as compared with ~10^2 seconds taken by Monte Carlo simulation.

- The Monte Carlo values are closer to the lower bound as compared to upper bound. For example, with $\sigma = 0.25$, the lower bound and Monte Carlo values are almost overlapping as seen in figure 5.5.
5 Numerical Results and Analysis

(a) Volatility=0.05

(b) Volatility=0.25

Figure 5.5: Asian Call Option: Geometric Brownian Motion
Ornstein-Uhlenbeck Process

The numerical data for this case is available in Table A.6. We provide 2 graphs for reference in figure 5.6. The call option values are not discounted and discounting factor $e^{-\rho T}$ is ignored in the calculation. We conclude the following from available numerical results:

- Monte Carlo values are in between the upper and lower bounds or are very near to lower bound in all the cases.

- As compared with geometric Brownian motion case, the bounds in this case are obtained in real time, and maximum time taken for any solving any SDP problem never exceeds 2 seconds.

- In this case, very accurate results are obtained with $3^{rd}$ order relaxation itself. This is evident from relative error which is $10^{-7} \%$ when $\sigma = 0.05$ and $R=3$ relaxations.
5 Numerical Results and Analysis

(a) Volatility=0.05

(b) Volatility=0.25

Figure 5.6: Asian Call Option: Ornstein Uhlenbeck Process
**Cox-Ingersoll-Ross Process**

We summarize the numerical results for Asian call options with underlying process following Cox-Ingersoll-Ross process in Table A.7 in Appendix A. Discounting factor has again been dropped and the observations from the numerical results are summarised as below:

- Monte Carlo value is below the lower bound when $\sigma = 0.05$. However the relative error in this case for SDP approach is 0.01% and variance of the call option price in Monte Carlo simulation is 0.001023, hence the SDP results are more accurate than Monte Carlo value.

- As we increase the volatility $\sigma$ from 0.05 to 0.25, Monte Carlo values move more closer to the lower bound, and for $\sigma = 0.25$, Monte Carlo values and lower bounds coincide with each other with very little difference.

- The maximum time take for any SDP program for this case never exceeded 2 seconds, as compared with that taken for Monte Carlo method, which was of order $10^2$ seconds.
5 Numerical Results and Analysis

(a) Volatility=0.15

(b) Volatility=0.25

Figure 5.7: Asian Call Option: Cox Ingersoll Ross Process
5.1.3 Down-and-out Barrier Options

This section covers numerical results for down-and-out Barrier options, obtained by solving SDP problem (4.7.5) for two different kinds of processes followed by underlying asset price.

**Ornstein-Uhlenbeck Process**

Table A.8 in Appendix A shows discounted values of down-and-out Barrier call options for different values of volatility $\sigma$. The underlying asset price process is modeled by Ornstein-Uhlenbeck process. Monte Carlo simulation values are given for comparison. Graphs in figure 5.8 are given as representative examples from the numerical data in table A.8.

Observations from the numerical results can be summarised as below:

- Monte Carlo values are in between the upper and lower bounds for almost all cases.
- When relaxations are low (between 1 and 3) the relative error is very high (nearly $\sim 200\%$). We get accurate results with $6^{th}$ relaxation onwards. For example, for $\sigma = 0.70$, $6^{th}$ relaxation gives $1.8\%$ relative error.
- The bounds are obtained in real time, if we use up to $4^{th}$ order relaxations, but for $5^{th}$ and $6^{th}$ order relaxations, the SDP program takes 5 and 20 seconds respectively. Even though high, this value is still very less as compared with $\sim 10^2$ seconds taken by Monte Carlo simulation.
5 Numerical Results and Analysis

(a) Volatility=0.30

(b) Volatility=0.70

Figure 5.8: Down and Out Barrier Call Option: Ornstein Uhlenbeck Process
5.1 SDP Moment Approach for Pricing Exotic Options

**Cox-Ingersoll-Ross Process**

Numerical data for this case is exhibited in Table A.8 in Appendix A. The values are not discounted and underlying asset price process is modeled by Cox-Ingersoll-Ross process. Reference graphs are given in 5.9. Observations from the numerical results can be summarised as below:

- As in the case with Ornstein-Uhlenbeck process, Monte Carlo values are either in between the upper and lower bounds for almost all cases, or they are very near to the converged value.

- Relative error is again very high for lower order relaxations (nearly $\sim 200\%$ for $R=1$) and we get accurate results with $6^{th}$ relaxation onwards.

- For $\sigma = 0.30$, $6^{th}$ relaxation gives $0.02\%$ relative error, where as variance for Monte Carlo simulation is $0.024312$ which suggests that SDP approach is more accurate than Monte Carlo simulation.

- As in the previous case, SDP problems take very less time for lower order relaxations (specifically, less than 2 seconds with $4^{th}$ relaxations). For $5^{th}$ and $6^{th}$ order relaxations, the SDP program roughly takes 6 and 18 seconds, respectively.

- We observe a peculiarity when $\sigma = 0.70$ and $R=6$, where upper bound is smaller than lower bound. This observation can be attributed to the difference in level of precision achieved by SeDuMi for these two cases and not as an error in model itself. It is to be noted that the bounds are still converging and relative error is $-0.06\%$. 


5 Numerical Results and Analysis

(a) Volatility=0.30

(b) Volatility=0.70

**Figure 5.9**: Down and Out Barrier Call Option: Cox Ingersoll Ross Process
5.1 SDP Moment Approach for Pricing Exotic Options

5.1.4 Up-and-out Barrier Options

In this section we cover the observations for numerical results for up-and-out Barrier options. We refer to SDP problem (4.7.11) for two cases: when underlying asset price follows Ornstein-Uhlenbeck and Cox-Ingersoll-Ross processes. Before going into detail, we note the following interesting facts in common for both the cases:

- Similar to down-and-out Barrier call options, Monte Carlo values are either in between the upper and lower bounds for both cases, or they are very near to the converged value.

- Relative error rapidly improves with increase in number of relaxations. For R=5 in both cases, relative error is well under 1 %. Lower relaxations give disproportionate error values (R=1 gives nearly 200 % error).

- The relative error increases with increase in volatility $\sigma$, when number of relaxation(R) is kept constant.

**Ornstein-Uhlenbeck Process**

Table A.10 in Appendix A shows values of up-and-out Barrier call options for different values of volatility $\sigma$. The underlying asset price process is modeled by Ornstein-Uhlenbeck process. We refer to figure 5.10 for two sample scenarios from amongst the numerical results obtained in table A.10.

Interesting observations for this particular case from the numerical results can be summarised as below:

- Time taken for solving SDP problems with higher order relaxations increases linearly with increase in number of relaxations. SDP problems with 4$^{th}$ and 6$^{th}$ order relaxation takes under 2 and 7 seconds respectively.

- Again very good accuracy is obtained with higher order relaxations. For example with $\sigma = 0.35$, 5$^{th}$ relaxation gives 0.04 % relative error, where as variance for Monte Carlo simulation is 0.080599, proving the SDP approach more accurate than Monte Carlo simulation.
5 Numerical Results and Analysis

(a) Volatility=0.35

(b) Volatility=0.55

Figure 5.10: Up and Out Barrier Call Option: Ornstein Uhlenbeck Process
Cox-Ingersoll-Ross Process

Numerical results for this case are presented in Table A.11 in Appendix A. The underlying asset price process follows Cox-Ingersoll-Ross process and two example cases are given in figure 5.11. We note the following peculiar observations from the numerical data:

- For $\sigma = 0.45$ and $0.50$, the upper bound is lower than lower bound. But as the sequence of bounds is converging, we attribute this error to the difference in level of precision achieved by SeDuMi for these two cases and not as an error in model itself. It is worth to be noted that the relative errors in both of above cases is $-0.25 \%$ and $-0.26 \%$ respectively.
Figure 5.11: Up and Out Barrier Call Option: Cox Ingersoll Ross Process
5.1.5 Double Barrier Options

We faced most of the numerical difficulties in SDP problems designed for pricing double Barrier options with higher order relaxations. Specifically we solved the SDP problem in (4.7.17) in order to obtain the call option values of double Barrier options. In the following sections we cover the cases when underlying process is modeled by Ornstein-Uhlenbeck and Cox-Ingersoll-Ross diffusions.

**Ornstein-Uhlenbeck Process**

For this case we provide the numerical results in Table A.12 in Appendix A. The underlying asset price process follows Ornstein-Uhlenbeck process. Please refer figure 5.12 for two examples representing the numerical results. We note the following observations from the numerical data:

- For $\sigma = 0.10$ and $0.15$, and relaxations $R = 5, 6, 7$ and $8$, SeDuMi failed with numerical problems, citing failure to achieve desired precision in results as the reason. The values given in the table are the latest bounds available when SeDuMi failed, and hence they don’t represent the accurate values. Hence we see that the upper and lower bounds are not converging to a definite value.

- It is worth noting that the SDP problems with higher relaxations become huge and hence SeDuMi fails to successfully solve them. As an example for $R=5$, the SDP problem involves, amongst other constraints, 4 (moment) matrices of size $5 \times 5$, 1 (moment) matrix of size $21 \times 21$, 4 (localizing) matrices of size $5 \times 5$ and 2 (localizing matrices) of size $15 \times 15$, which involves total 1091 variables in semidefinite matrices.

- We however note that even though SeDuMi failed for higher order relaxations, we obtain encouraging accuracy from $6^{th}$ order relaxation and even though the bounds don’t represent the correct values, they are very close to the Monte Carlo values.

- Finally we also note that in Lasserre, Prieto-Rumeau and Zervos in [21], authors also faced numerical difficulties with SeDuMi for SDP problems for Barrier options when relaxations $R=8$. This fact is in conjunction with the errors faced by us for SDP problems with semidefinite matrices of higher order.
Figure 5.12: Combined Barrier Call Option: Ornstein-Uhlenbeck Process
Cox-Ingersoll-Ross Process

Our final scenario in this section deals with double Barrier options when underlying asset follows Cox-Ingersoll-Ross diffusion. The detailed numerical data is available in Table A.12 in Appendix A and figure 5.13 shows two representative examples from the numerical results. The peculiar observations for this case are cited below:

- Similar to previous section, SeDuMi failed to solve the SDP problems for $\sigma = 0.30$ and 0.40, and relaxations $R = 4, 5$ and 6. The values given in the table are the latest bounds available when SeDuMi failed, which makes their value inaccurate.

- We however note that even though SeDuMi failed for higher order relaxations, we obtain encouraging accuracy for 5th and 6th order relaxations and even though the bounds don’t represent the correct values, they are very close to the Monte Carlo values, with very low relative error (of the range 0.06 % to 1.07 %).

- SDP problems with lower order relaxations (up to 4th order relaxation) achieved the bounds in near real time (≤ 1 second), however SDP problems for 5th and 6th order relaxations failed after 6 and 17 seconds, respectively.
Figure 5.13: Combined Barrier Call Option: Cox Ingersoll Ross Process
5.2 Semi-parametric Approach for Pricing European Options

Finally, before going on to next section, we will analyse the numerical results obtained for European call options by following semi-parametric approach developed in chapter 3. Table A.1 in Appendix A shows the discounted values of European call options (with underlying process following a geometric Brownian motion), obtained by solving SDP problems (3.2.1), for different values of volatility $\sigma$. For comparison, we have presented the Black Scholes value. Two examples from the numerical data are shown in figure 5.14.

We note the following points regarding the numerical results:

- As we increase relaxations $R$, the accuracy of results increase and bounds become more tighter.

- The degree of improvement in bounds is not uniform and depends on values of volatility $\sigma$, maturity time $T$ and strike price $K$. For example, as we increase the volatility $\sigma$ from 0.20 to 0.80 and keep number of relaxations $R$ constant, we note that distance between upper and lower bounds increases.

- When strike price is lower than initial stock price, the lower bound is higher than Black Scholes value. For example when strike price is less that 35, the lower bound is above Black Scholes value and is thus not accurate.

- When strike price is higher (for example above 36 in given numerical example), the Black Scholes value is contained within upper and lower bounds.
Figure 5.14: Semi-parametric Approach, European Call Option: Geometric Brownian Motion
In order to interpret the data better, we now present the two different perspectives from the same data. The two graphs present in figure 5.15 show the difference between bounds of higher order moments and Black-Scholes price when we

- add higher order moments while keeping volatility $\sigma$ constant, and
- increase the volatility $\sigma$ with the same amount of moment information (i.e. keeping $R$ constant).

We observe the following from figure 5.15(a):

- Lower bound solution (obtained from all SDP programs with higher order relaxations) is worse than Black Scholes price when strike price is comparatively lower than initial stock price. This can be seen from the negative difference between Black-Scholes value and lower bound.

- Increase in number of relaxations decreases the difference between upper/lower bound and Black Scholes value.

- Difference between upper bound and Black-Scholes value is lesser than difference between lower bound and Black-Scholes value.

- Discrepancy in upper and lower bound increases as strike price $K$ approaches the initial stock price $S_0$, whereas same discrepancy is much less when strike price $K$ is far from initial stock price $S_0$.

We also observe the following from figure 5.15(b):

- Lower bound is more sensitive to change in volatility $\sigma$ as compared to upper bound.

- Increase in volatility $\sigma$ decreases the difference between upper/lower bound and Black Scholes value.

- Discrepancy in upper and lower bound is highest when value of strike price $K$ approaches the initial stock price $S_0$, whereas same discrepancy is much less when strike price $K$ is far from initial stock price $S_0$. 
5 Numerical Results and Analysis

(a) Difference between bounds of higher order moments and B-S price

(b) Difference between bounds and B-S price for different volatilities

Figure 5.15: Semi-parametric Approach, European Call Option: Geometric Brownian Motion
Chapter 6

Conclusion

It is a well known fact that Black-Scholes formula was derived under several assumptions which are not always true in the real capital market. For example, if the underlying asset price deviates from the geometric Brownian motion, then significant amount of mispricing can occur. Subsequently, several authors have suggested alternative models and methodologies for pricing options which attempt to overcome the shortcomings in Black-Scholes model. In this project, we investigated two such recently proposed methodologies for pricing various types of plain and exotic options.

In the semi-parametric approach proposed originally by Lo in [22] and extended by Bertsimas and Popescu in [1] and Gotoh and Konno in [9], we approximate the value of European call options by solving two semidefinite programming (SDP) problems. For this approach, we need to know the first few moments of the risk neutral probability, and not the probability distribution itself. Very accurate bounds are available in real time as we saw in the numerical results. According to authors, this approach assumes importance in scenarios where closed-form analytical solution is not available for complex derivatives, but first few moments are available. Thus this semi-parametric approach could be applied to more general class of moment problems.

In the SDP moment approach suggested by Lasserre, Prieto-Rumeau and Zervos in [21], several exotic options could be priced, where underlying asset price process is not restricted to be geometric Brownian motion as in Black-Scholes model. This approach also solves two SDP problems for bounding the option price from above and below. Both approaches are similar in the sense that underlying problems to be solved are SDP problems, however the approach to develop those SDP problems are different. The main difference is that in the SDP moment approach, we do not need to know the moments of underlying asset price process, as is the case with semi-parametric approach. However one main restriction of SDP moment approach is that infinitesimal generator of the underlying asset price process must map polynomials into polynomials of equal or lesser degree. Another major assumption for convergence of bounds in SDP moments approach requires that underlying asset price’s probability distribution should be moment determinate. However, even though probability distribution of geometric Brownian motion is not moment determinate, we could observe from the numerical results that impressive bounds were achieved in the cases when asset price was modeled to follow geometric Brownian motion.

6.1 Numerical Remarks

In this project, we implemented semi-parametric approach for European options and SDP moment approach for several class of exotic options. For the semi-parametric approach we obtained very accurate results in real time with first few moments. However the lower bounds were not accurate when strike price was very small as compared to initial stock price.

For the SDP moment approach, we were successfully able to formulate the SDP problems for
European, up-and-out and double Barrier options. We also successfully implemented the SDP problems for European, Asian and (down-and-out, up-and-out and double) Barrier options, when underlying asset price could follow either of the following processes: geometric Brownian motion, Ornstein-Uhlenbeck process or Cox-Ingersoll-Ross process. Comprehensive numerical results were produced and comparison with Monte Carlo values proved that bounds achieved through SDP approach are more accurate. For example, with European, Asian and down-and-out and up-and-out options, $5^{th}$ or $6^{th}$ order relaxations were sufficient to achieve very accurate results (with relative error less than 1 %) and such bounds were frequently more accurate than the corresponding Monte Carlo values.

On the other hand, we faced several numerical difficulties for double Barrier options even when relaxations were very few. In one of the cases, the SDP solver failed in $4^{th}$ relaxation itself. Such problems frequently occurred owing to the fact that SDP solver SeDuMi failed to achieve the desired accuracy. However, despite the failure of SeDuMi to solve the SDP problems, the inaccurate results obtained on premature termination of SeDuMi were very close to Monte Carlo values. Hence such numerical problems could be attributed to SeDuMi’s failure to successfully solve large SDP problems and not an error in the model itself. We believe that such shortcomings are common in current SDP solvers, as substantiated by the fact that Cutting Plane algorithm given by Gotoh and Konno in [8] also could not handle SDP problems with semidefinite matrices of size 9 or more. Finally we note that even though most of the SDP solvers are comparatively recently developed, we could expect more stability from them as there is more progress in numerical analysis in solving SDP optimization problems.

### 6.2 Future Direction of Research

![Example of Complex Double Barrier Option](image)

**Figure 6.1:** Example of Complex Double Barrier Option

To conclude this section, we present several interesting ideas as an extension to this project. Some of the ideas are comparatively trivial and others are major projects in themselves.

- SDP moment approach can be extended to price all of the above options when underlying asset price process is given as in Heston model, where the volatility of asset price is itself stochastic and follows Cox-Ingersoll-Ross model.
• SDP moment approach can be extended for average strike Asian options. (This project covers the average price Asian options).

• SDP moment approach can also be applied to the following generalizations of the barrier options:
  
  – Partial barrier options, where barrier applies for a limited time interval.
  – Parisian options where barriers must remain crossed from some pre-specified amount of time.
  – More general time dependent barriers. For example as shown in figure 6.1, the lower and upper Barriers change linearly with time.
Appendix A

Tables and Figures for Numerical Results

A.1 Semi-parametric Approach for Pricing European Options

In the table A.1, we present the numerical values for European call options obtained by following semi-parametric approach presented in chapter 3. In the table, upper and lower bounds are denoted by abbreviations UB and LB respectively and Black-Scholes value is denoted by BS. The values of other parameters are listed on the top of table for reference where the symbols have the usual meaning: R = relaxations, S_0 = initial stock price, K = strike price, T = time, r = risk-free interest rate and σ = volatility.

A.2 SDP Moment Approach for Pricing Exotic Options

Finally, we present the numerical values for different types of call options obtained by following SDP moment approach presented in chapter 4. In the tables A.2 to A.13, upper bound, lower bounds and Black-Scholes value are denoted by abbreviations UB, LB and BS respectively. The values of other parameters are listed on the top of table for reference where the symbols have the usual meaning: R = relaxations, S_0 = initial stock price, K = strike price, T = time, r = risk-free interest rate, γ and θ = parameters for Ornstein-Uhlenbeck and Cox-Ingersoll-Ross diffusions, σ = volatility, H_L and H_U = lower and upper barriers, respectively, for Barrier options.
### Table A.1: European Call Option Values: Geometric Brownian Motion

Parameters: $R = [3,...,5]$, $S_0 = 40$, $K = [28,...,52]$, $T = 1$ week, $r = 0.06$; $\sigma = [0.20,...,0.80]$

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>2-moment</th>
<th>3-moment</th>
<th>4-moment</th>
<th>BS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma = 0.20$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>UB</td>
<td>LB</td>
<td>UB</td>
<td>LB</td>
</tr>
<tr>
<td>37</td>
<td>3.166005</td>
<td>3.056615</td>
<td>3.286707</td>
<td>3.047553</td>
</tr>
<tr>
<td>40</td>
<td>0.590438</td>
<td>0.068655</td>
<td>0.610815</td>
<td>0.069774</td>
</tr>
<tr>
<td>43</td>
<td>0.086801</td>
<td>0.000000</td>
<td>0.128157</td>
<td>0.000021</td>
</tr>
<tr>
<td>46</td>
<td>0.060769</td>
<td>0.000000</td>
<td>0.155394</td>
<td>0.000017</td>
</tr>
<tr>
<td>49</td>
<td>0.026938</td>
<td>0.000000</td>
<td>0.199215</td>
<td>0.000003</td>
</tr>
<tr>
<td>52</td>
<td>0.014670</td>
<td>0.000000</td>
<td>0.116712</td>
<td>0.000005</td>
</tr>
<tr>
<td>$\sigma = 0.40$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>UB</td>
<td>LB</td>
<td>UB</td>
<td>LB</td>
</tr>
<tr>
<td>37</td>
<td>3.477811</td>
<td>3.596074</td>
<td>3.599848</td>
<td>3.115097</td>
</tr>
<tr>
<td>40</td>
<td>1.171434</td>
<td>1.192773</td>
<td>1.204293</td>
<td>0.906530</td>
</tr>
<tr>
<td>43</td>
<td>0.020712</td>
<td>0.000002</td>
<td>0.155394</td>
<td>0.000017</td>
</tr>
<tr>
<td>46</td>
<td>0.010424</td>
<td>0.000000</td>
<td>0.186409</td>
<td>0.000003</td>
</tr>
<tr>
<td>49</td>
<td>0.006674</td>
<td>0.000000</td>
<td>0.185894</td>
<td>0.000009</td>
</tr>
<tr>
<td>52</td>
<td>0.002993</td>
<td>0.000000</td>
<td>0.100424</td>
<td>0.000000</td>
</tr>
<tr>
<td>$\sigma = 0.60$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>UB</td>
<td>LB</td>
<td>UB</td>
<td>LB</td>
</tr>
<tr>
<td>37</td>
<td>3.177670</td>
<td>3.170443</td>
<td>3.170443</td>
<td>3.333736</td>
</tr>
<tr>
<td>40</td>
<td>1.781318</td>
<td>1.873277</td>
<td>1.873277</td>
<td>1.347924</td>
</tr>
<tr>
<td>43</td>
<td>0.717434</td>
<td>0.649091</td>
<td>0.649091</td>
<td>0.371427</td>
</tr>
<tr>
<td>46</td>
<td>0.324220</td>
<td>0.289494</td>
<td>0.289494</td>
<td>0.070322</td>
</tr>
<tr>
<td>49</td>
<td>0.224220</td>
<td>0.155781</td>
<td>0.155781</td>
<td>0.09168</td>
</tr>
<tr>
<td>52</td>
<td>0.156868</td>
<td>0.100692</td>
<td>0.100692</td>
<td>0.000863</td>
</tr>
<tr>
<td>$\sigma = 0.80$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>UB</td>
<td>LB</td>
<td>UB</td>
<td>LB</td>
</tr>
<tr>
<td>37</td>
<td>4.422705</td>
<td>4.422705</td>
<td>4.422705</td>
<td>3.636261</td>
</tr>
<tr>
<td>40</td>
<td>2.395239</td>
<td>2.395239</td>
<td>2.395239</td>
<td>3.636261</td>
</tr>
<tr>
<td>43</td>
<td>1.193794</td>
<td>1.152693</td>
<td>1.152693</td>
<td>0.721913</td>
</tr>
<tr>
<td>46</td>
<td>0.652730</td>
<td>0.553779</td>
<td>0.553779</td>
<td>0.239442</td>
</tr>
<tr>
<td>49</td>
<td>0.414507</td>
<td>0.235775</td>
<td>0.235775</td>
<td>0.066217</td>
</tr>
<tr>
<td>52</td>
<td>0.284298</td>
<td>0.194895</td>
<td>0.194895</td>
<td>0.015570</td>
</tr>
</tbody>
</table>
Figure A.1: Semi-parametric Approach, European Call Option: Geometric Brownian Motion

(a) Volatility=0.20

(b) Volatility=0.40

(c) Volatility=0.60
Table A.2: European Call Option Values: Geometric Brownian Motion

Parameters: $R = [1, ..., 8], S_0 = 1, K = 0.80, T = 3, r = 0.06; \sigma = [0.05, ..., 0.25]$  

<table>
<thead>
<tr>
<th>$\sigma = 0.05$</th>
<th>$\sigma = 0.10$</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>$R$</th>
<th>UB</th>
<th>LB</th>
<th>UB−LB</th>
<th>Error</th>
<th>$R$</th>
<th>UB</th>
<th>LB</th>
<th>UB−LB</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.337363</td>
<td>0.331784</td>
<td>0.005579</td>
<td>1.67</td>
<td>1</td>
<td>0.353332</td>
<td>0.331784</td>
<td>0.021548</td>
<td>6.29</td>
</tr>
<tr>
<td>2</td>
<td>0.332087</td>
<td>0.331784</td>
<td>0.000304</td>
<td>0.09</td>
<td>2</td>
<td>0.335764</td>
<td>0.331784</td>
<td>0.003980</td>
<td>1.19</td>
</tr>
<tr>
<td>3</td>
<td>0.331822</td>
<td>0.331784</td>
<td>0.000038</td>
<td>0.01</td>
<td>3</td>
<td>0.333431</td>
<td>0.331784</td>
<td>0.001647</td>
<td>0.50</td>
</tr>
<tr>
<td>4</td>
<td>0.331791</td>
<td>0.331784</td>
<td>0.000007</td>
<td>0.00</td>
<td>4</td>
<td>0.332841</td>
<td>0.331784</td>
<td>0.001057</td>
<td>0.32</td>
</tr>
<tr>
<td>5</td>
<td>0.331788</td>
<td>0.331784</td>
<td>0.000004</td>
<td>0.00</td>
<td>5</td>
<td>0.332681</td>
<td>0.331784</td>
<td>0.000897</td>
<td>0.27</td>
</tr>
<tr>
<td>6</td>
<td>0.331787</td>
<td>0.331784</td>
<td>0.000003</td>
<td>0.00</td>
<td>6</td>
<td>0.332663</td>
<td>0.331784</td>
<td>0.000879</td>
<td>0.26</td>
</tr>
<tr>
<td>7</td>
<td>0.331787</td>
<td>0.331784</td>
<td>0.000003</td>
<td>0.00</td>
<td>7</td>
<td>0.332656</td>
<td>0.331785</td>
<td>0.000871</td>
<td>0.26</td>
</tr>
<tr>
<td>8</td>
<td>0.331786</td>
<td>0.331784</td>
<td>0.000002</td>
<td>0.00</td>
<td>8</td>
<td>0.332656</td>
<td>0.331788</td>
<td>0.000871</td>
<td>0.26</td>
</tr>
</tbody>
</table>

Black Scholes Value: 0.331784

<table>
<thead>
<tr>
<th>$\sigma = 0.15$</th>
<th>$\sigma = 0.20$</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>$R$</th>
<th>UB</th>
<th>LB</th>
<th>UB−LB</th>
<th>Error</th>
<th>$R$</th>
<th>UB</th>
<th>LB</th>
<th>UB−LB</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.377971</td>
<td>0.331784</td>
<td>0.046188</td>
<td>13.02</td>
<td>1</td>
<td>0.409601</td>
<td>0.331784</td>
<td>0.077818</td>
<td>20.99</td>
</tr>
<tr>
<td>2</td>
<td>0.346751</td>
<td>0.331784</td>
<td>0.014967</td>
<td>4.41</td>
<td>2</td>
<td>0.365454</td>
<td>0.331784</td>
<td>0.033670</td>
<td>9.66</td>
</tr>
<tr>
<td>3</td>
<td>0.341610</td>
<td>0.331784</td>
<td>0.009827</td>
<td>2.92</td>
<td>3</td>
<td>0.357842</td>
<td>0.331784</td>
<td>0.026058</td>
<td>7.56</td>
</tr>
<tr>
<td>4</td>
<td>0.340538</td>
<td>0.331784</td>
<td>0.008754</td>
<td>2.60</td>
<td>4</td>
<td>0.356432</td>
<td>0.332160</td>
<td>0.024272</td>
<td>7.05</td>
</tr>
<tr>
<td>5</td>
<td>0.340468</td>
<td>0.332119</td>
<td>0.008349</td>
<td>2.48</td>
<td>5</td>
<td>0.356369</td>
<td>0.334713</td>
<td>0.021657</td>
<td>6.27</td>
</tr>
<tr>
<td>6</td>
<td>0.340397</td>
<td>0.333126</td>
<td>0.007272</td>
<td>2.16</td>
<td>6</td>
<td>0.356261</td>
<td>0.337490</td>
<td>0.018771</td>
<td>5.41</td>
</tr>
<tr>
<td>7</td>
<td>(0.3400)</td>
<td>(0.3337)</td>
<td>0.006280</td>
<td>(1.86)</td>
<td>7</td>
<td>(0.3558)</td>
<td>(0.3392)</td>
<td>0.016549</td>
<td>(4.76)</td>
</tr>
<tr>
<td>8</td>
<td>(0.3401)</td>
<td>(0.3331)</td>
<td>0.006996</td>
<td>(2.08)</td>
<td>8</td>
<td>(0.3557)</td>
<td>(0.3359)</td>
<td>0.019852</td>
<td>(5.74)</td>
</tr>
</tbody>
</table>

Black Scholes Value: 0.337276

<table>
<thead>
<tr>
<th>$\sigma = 0.25$</th>
<th>Average CPU Time(sec)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>$R$</th>
<th>UB</th>
<th>LB</th>
<th>UB−LB</th>
<th>Error</th>
<th>$R$</th>
<th>UB</th>
<th>LB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.447099</td>
<td>0.331784</td>
<td>0.115316</td>
<td>29.61</td>
<td>1</td>
<td>0.504265</td>
<td>0.504265</td>
</tr>
<tr>
<td>2</td>
<td>0.390416</td>
<td>0.331784</td>
<td>0.058633</td>
<td>16.24</td>
<td>2</td>
<td>0.259718</td>
<td>0.259718</td>
</tr>
<tr>
<td>3</td>
<td>0.379904</td>
<td>0.331784</td>
<td>0.048119</td>
<td>13.52</td>
<td>3</td>
<td>0.320804</td>
<td>0.320804</td>
</tr>
<tr>
<td>4</td>
<td>0.377561</td>
<td>0.332166</td>
<td>0.044345</td>
<td>12.48</td>
<td>4</td>
<td>0.356671</td>
<td>0.356671</td>
</tr>
<tr>
<td>5</td>
<td>0.377217</td>
<td>0.337466</td>
<td>0.039751</td>
<td>11.12</td>
<td>5</td>
<td>0.445374</td>
<td>0.445374</td>
</tr>
<tr>
<td>6</td>
<td>0.377398</td>
<td>0.341600</td>
<td>0.035799</td>
<td>9.96</td>
<td>6</td>
<td>0.688186</td>
<td>0.688186</td>
</tr>
<tr>
<td>7</td>
<td>(0.3875)</td>
<td>(0.3412)</td>
<td>0.046306</td>
<td>(12.71)</td>
<td>7</td>
<td>0.950886</td>
<td>0.950886</td>
</tr>
<tr>
<td>8</td>
<td>(0.4302)</td>
<td>(0.2531)</td>
<td>0.177068</td>
<td>(51.82)</td>
<td>8</td>
<td>1.253114</td>
<td>1.253114</td>
</tr>
</tbody>
</table>

Black Scholes Value: 0.364873

Black Scholes Timing: 0.32
Figure A.2: European Call Option: Geometric Brownian Motion
### Table A.3: European Call Option Values: Ornstein-Uhlenbeck Process

Parameters: \( R = [1,...,8] \), \( S_0 = 1 \), \( K = 0.95 \), \( T = 2 \), \( \gamma = 1 \), \( \theta = 1.1 \), \( \sigma = [0.05,...,0.25] \)

<table>
<thead>
<tr>
<th>( \sigma = 0.05 )</th>
<th>( \sigma = 0.10 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R )</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.138679</td>
</tr>
<tr>
<td>2</td>
<td>0.136591</td>
</tr>
<tr>
<td>3</td>
<td>0.136501</td>
</tr>
<tr>
<td>4</td>
<td>0.136490</td>
</tr>
<tr>
<td>5</td>
<td>0.136489</td>
</tr>
<tr>
<td>6</td>
<td>0.136489</td>
</tr>
<tr>
<td>7</td>
<td>0.136485</td>
</tr>
<tr>
<td>8</td>
<td>0.136481</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.136502
Variance: 0.001227

<table>
<thead>
<tr>
<th>( \sigma = 0.15 )</th>
<th>( \sigma = 0.20 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R )</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.154354</td>
</tr>
<tr>
<td>2</td>
<td>0.144355</td>
</tr>
<tr>
<td>3</td>
<td>0.144251</td>
</tr>
<tr>
<td>4</td>
<td>0.143402</td>
</tr>
<tr>
<td>5</td>
<td>0.143391</td>
</tr>
<tr>
<td>6</td>
<td>0.143272</td>
</tr>
<tr>
<td>7</td>
<td>0.142890</td>
</tr>
<tr>
<td>8</td>
<td>0.142833</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.141308
Variance: 0.009304

<table>
<thead>
<tr>
<th>( \sigma = 0.25 )</th>
<th>Average CPU Time(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R )</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.179252</td>
</tr>
<tr>
<td>2</td>
<td>0.168256</td>
</tr>
<tr>
<td>3</td>
<td>0.166440</td>
</tr>
<tr>
<td>4</td>
<td>0.163008</td>
</tr>
<tr>
<td>5</td>
<td>0.163005</td>
</tr>
<tr>
<td>6</td>
<td>0.162671</td>
</tr>
<tr>
<td>7</td>
<td>0.162103</td>
</tr>
<tr>
<td>8</td>
<td>0.162089</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.159202
Variance: 0.020498

Monte Carlo Simulation: 0.137101
Variance: 0.004720

Monte Carlo Simulation: 0.148310
Variance: 0.014493

Monte Carlo Timing: 332.93938
Figure A.3: European Call Option: Ornstein-Uhlenbeck Process
Table A.4: European Call Option: Cox Ingersoll Ross Process

Parameters: $R = [1,...,8]$, $S_0 = 1$, $K = 0.95$, $T = 3$, $\gamma = 0.9$, $\theta = 1.1$, $\sigma = [0.05,...,0.25]$

<table>
<thead>
<tr>
<th>$\sigma = 0.05$</th>
<th>$\sigma = 0.10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.145856</td>
</tr>
<tr>
<td>2</td>
<td>0.143437</td>
</tr>
<tr>
<td>3</td>
<td>0.143307</td>
</tr>
<tr>
<td>4</td>
<td>0.143306</td>
</tr>
<tr>
<td>5</td>
<td>0.143304</td>
</tr>
<tr>
<td>6</td>
<td>0.143298</td>
</tr>
<tr>
<td>7</td>
<td>0.143289</td>
</tr>
<tr>
<td>8</td>
<td>0.143288</td>
</tr>
<tr>
<td>Monte Carlo Simulation: 0.143251</td>
<td>Variance: 0.001487</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\sigma = 0.15$</th>
<th>$\sigma = 0.20$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.163917</td>
</tr>
<tr>
<td>2</td>
<td>0.151933</td>
</tr>
<tr>
<td>3</td>
<td>0.151432</td>
</tr>
<tr>
<td>4</td>
<td>0.150935</td>
</tr>
<tr>
<td>5</td>
<td>0.150799</td>
</tr>
<tr>
<td>6</td>
<td>0.150271</td>
</tr>
<tr>
<td>7</td>
<td>0.150224</td>
</tr>
<tr>
<td>8</td>
<td>0.150223</td>
</tr>
<tr>
<td>Monte Carlo Simulation: 0.148184</td>
<td>Variance: 0.011580</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\sigma = 0.25$</th>
<th>Average CPU Time(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.192176</td>
</tr>
<tr>
<td>2</td>
<td>0.175931</td>
</tr>
<tr>
<td>3</td>
<td>0.175822</td>
</tr>
<tr>
<td>4</td>
<td>0.172344</td>
</tr>
<tr>
<td>5</td>
<td>0.170701</td>
</tr>
<tr>
<td>6</td>
<td>0.170705</td>
</tr>
<tr>
<td>7</td>
<td>0.170536</td>
</tr>
<tr>
<td>8</td>
<td>0.170252</td>
</tr>
<tr>
<td>Monte Carlo Simulation: 0.166492</td>
<td>Variance: 0.026874</td>
</tr>
</tbody>
</table>
Figure A.4: European Call Option: Cox Ingersoll Ross Process

(a) Volatility = 0.05

(b) Volatility = 0.15

(c) Volatility = 0.25
### Table A.5: Asian Call Option: Geometric Brownian Motion

Parameters: \( R = [1, \ldots, 5] \), \( S_0 = 1 \), \( K = 1.05 \), \( T = 4 \), \( r = 0.16 \), \( \sigma = [0.05, \ldots, 0.25] \)

<table>
<thead>
<tr>
<th>( \sigma = 0.05 )</th>
<th>( \sigma = 0.10 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R )</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.356118</td>
</tr>
<tr>
<td>2</td>
<td>0.351023</td>
</tr>
<tr>
<td>3</td>
<td>0.350785</td>
</tr>
<tr>
<td>4</td>
<td>0.350783</td>
</tr>
<tr>
<td>5</td>
<td>0.350770</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.350965
Confidence Interval(95%): [0.351507,0.350423]
Variance: 0.007638

<table>
<thead>
<tr>
<th>( \sigma = 0.15 )</th>
<th>( \sigma = 0.20 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R )</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.395240</td>
</tr>
<tr>
<td>2</td>
<td>0.364725</td>
</tr>
<tr>
<td>3</td>
<td>0.360085</td>
</tr>
<tr>
<td>4</td>
<td>0.359348</td>
</tr>
<tr>
<td>5</td>
<td>0.359373</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.350850
Confidence Interval(95%): [0.352492,0.349208]
Variance: 0.070191

<table>
<thead>
<tr>
<th>( \sigma = 0.25 )</th>
<th>Average CPU Time(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R )</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.461458</td>
</tr>
<tr>
<td>2</td>
<td>0.406932</td>
</tr>
<tr>
<td>3</td>
<td>0.398564</td>
</tr>
<tr>
<td>4</td>
<td>0.397564</td>
</tr>
<tr>
<td>5</td>
<td>0.397570</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.352483
Confidence Interval(95%): [0.355295,0.349670]
Variance: 0.205888

Monte Carlo Timing: 107.49
APPENDIX A

(a) Volatility=0.05

(b) Volatility=0.15

(c) Volatility=0.25

Figure A.5: Asian Call Option: Geometric Brownian Motion
### Table A.6: Asian Call Option: Ornstein-Uhlenbeck Process

Parameters: $R = [1, \ldots, 5]$, $S_0 = 1$, $K = 0.90$, $T = 3$, $\gamma = 1.1$, $\theta = 1.2$, $\sigma = [0.05, \ldots, 0.25]$  

<table>
<thead>
<tr>
<th>$\sigma = 0.05$</th>
<th>$\sigma = 0.10$</th>
<th>$\sigma = 0.15$</th>
<th>$\sigma = 0.20$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.242136</td>
<td>0.241629</td>
<td>0.000507</td>
</tr>
<tr>
<td>2</td>
<td>0.241632</td>
<td>0.241629</td>
<td>0.000002</td>
</tr>
<tr>
<td>3</td>
<td>0.241632</td>
<td>0.241629</td>
<td>0.000003</td>
</tr>
<tr>
<td>4</td>
<td>0.241630</td>
<td>0.241629</td>
<td>0.000000</td>
</tr>
<tr>
<td>5</td>
<td>0.241630</td>
<td>0.241629</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.241664  
Variance: 0.000394  

Monte Carlo Simulation: 0.241713  
Variance: 0.001567  

Monte Carlo Simulation: 0.241641  
Variance: 0.003494  

Monte Carlo Simulation: 0.241593  
Variance: 0.006256  

Monte Carlo Simulation: 0.241750  
Variance: 0.009756  

<table>
<thead>
<tr>
<th>Average CPU Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

Monte Carlo Timing: 36.41
Figure A.6: Asian Call Option: Ornstein Uhlenbeck Process
### Table A.7: Asian Call Option: Cox Ingersoll Ross Process

Parameters: $R = [1, \ldots, 5]$, $S_0 = 1$, $K = 0.90$, $T = 3$, $\gamma = 0.5$, $\theta = 1.2$, $\sigma = [0.05, \ldots, 0.25]$

<table>
<thead>
<tr>
<th>$\sigma = 0.05$</th>
<th>$\sigma = 0.10$</th>
<th>$\sigma = 0.15$</th>
<th>$\sigma = 0.20$</th>
<th>$\sigma = 0.25$</th>
<th>Average CPU Time(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.197681</td>
<td>0.196417</td>
<td>0.001263</td>
<td>0.64</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0.196442</td>
<td>0.196417</td>
<td>0.000024</td>
<td>0.01</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0.196437</td>
<td>0.196417</td>
<td>0.000020</td>
<td>0.01</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>0.196421</td>
<td>0.196420</td>
<td>0.000001</td>
<td>0.00</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>0.196436</td>
<td>0.196425</td>
<td>0.000011</td>
<td>0.01</td>
<td>5</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.196257
Variance: 0.001023

Monte Carlo Simulation: 0.196382
Variance: 0.003933

Monte Carlo Simulation: 0.196211
Variance: 0.008888

Monte Carlo Simulation: 0.196398
Variance: 0.024753

Monte Carlo Timing: 49.00

Average CPU Time: 0.015876

Monte Carlo Timing: 49.00
Figure A.7: Asian Call Option: Cox Ingersoll Ross Process

(a) Volatility=0.05

(b) Volatility=0.15

(c) Volatility=0.25
### Table A.8: Down and Out Barrier Call Option: Ornstein Uhlenbeck Process

Parameters: \( R = [1, \ldots, 6], S_0 = 1, K = 1, H = 0.8, T = 2, \gamma = 1, \theta = 0.95, \sigma = [0.30, \ldots, 0.70] \)

<table>
<thead>
<tr>
<th>( \sigma = 0.30 )</th>
<th>( \sigma = 0.40 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R )</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.188337</td>
</tr>
<tr>
<td>2</td>
<td>0.074764</td>
</tr>
<tr>
<td>3</td>
<td>0.044174</td>
</tr>
<tr>
<td>4</td>
<td>0.040415</td>
</tr>
<tr>
<td>5</td>
<td>0.038177</td>
</tr>
<tr>
<td>6</td>
<td>0.036523</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.032096
Variance: 0.007781

<table>
<thead>
<tr>
<th>( \sigma = 0.50 )</th>
<th>( \sigma = 0.60 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R )</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.297984</td>
</tr>
<tr>
<td>2</td>
<td>0.056150</td>
</tr>
<tr>
<td>3</td>
<td>0.040689</td>
</tr>
<tr>
<td>4</td>
<td>0.038588</td>
</tr>
<tr>
<td>5</td>
<td>0.037278</td>
</tr>
<tr>
<td>6</td>
<td>0.036229</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.042197
Variance: 0.019996

<table>
<thead>
<tr>
<th>( \sigma = 0.70 )</th>
<th>Average CPU Time(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R )</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.405649</td>
</tr>
<tr>
<td>2</td>
<td>0.048219</td>
</tr>
<tr>
<td>3</td>
<td>0.037854</td>
</tr>
<tr>
<td>4</td>
<td>0.036226</td>
</tr>
<tr>
<td>5</td>
<td>0.034785</td>
</tr>
<tr>
<td>6</td>
<td>0.033853</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.026950
Variance: 0.015285

<table>
<thead>
<tr>
<th>Average CPU Time(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R )</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
</tbody>
</table>

Monte Carlo Timing: 106.68
Figure A.8: Down and Out Barrier Call Option: Ornstein Uhlenbeck Process
### Table A.9: Down and Out Barrier Call Option: Cox Ingersoll Ross Process

Parameters: $R = [1,...,6]$, $S_0 = 1$, $K = 1$, $T = 3$, $H=0.8$, $\gamma = 1$, $\theta = 0.95$, $\sigma = [0.30,...,0.70]$

<table>
<thead>
<tr>
<th>$\sigma = 0.30$</th>
<th>$\sigma = 0.40$</th>
<th>$\sigma = 0.50$</th>
<th>$\sigma = 0.60$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.244159</td>
<td>0.000000</td>
<td>0.244159</td>
</tr>
<tr>
<td>2</td>
<td>0.095773</td>
<td>0.000001</td>
<td>0.095772</td>
</tr>
<tr>
<td>3</td>
<td>0.031514</td>
<td>0.012284</td>
<td>0.019230</td>
</tr>
<tr>
<td>4</td>
<td>0.026075</td>
<td>0.017175</td>
<td>0.008899</td>
</tr>
<tr>
<td>5</td>
<td>0.022336</td>
<td>0.020791</td>
<td>0.001546</td>
</tr>
<tr>
<td>6</td>
<td>0.021982</td>
<td>0.021785</td>
<td>0.000004</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.024312
Variance: 0.006167

Monte Carlo Simulation: 0.026120
Variance: 0.014680

Monte Carlo Simulation: 0.024593
Variance: 0.017088

Monte Carlo Simulation: 0.020986
Variance: 0.020768

Monte Carlo Timing: 82.94
Figure A.9: Down and Out Barrier Call Option: Cox Ingersoll Ross Process
### Table A.10: Up and Out Barrier Call Option: Ornstein Uhlenbeck Process

Parameters: \( R = [1,...,6], S_0 = 1.2, K = 0.40, T = 3, H = 1.4, \gamma = 1, \theta = 1.1, \sigma = [0.35,...,0.55] \)

<table>
<thead>
<tr>
<th>( \sigma = 0.35 )</th>
<th>( \sigma = 0.40 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R )</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.997048</td>
</tr>
<tr>
<td>2</td>
<td>0.701871</td>
</tr>
<tr>
<td>3</td>
<td>0.226189</td>
</tr>
<tr>
<td>4</td>
<td>0.183810</td>
</tr>
<tr>
<td>5</td>
<td>0.173877</td>
</tr>
<tr>
<td>6</td>
<td>0.171944</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.168891
Variance: 0.080599

<table>
<thead>
<tr>
<th>( \sigma = 0.45 )</th>
<th>( \sigma = 0.50 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R )</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.997436</td>
</tr>
<tr>
<td>2</td>
<td>0.357609</td>
</tr>
<tr>
<td>3</td>
<td>0.142087</td>
</tr>
<tr>
<td>4</td>
<td>0.100376</td>
</tr>
<tr>
<td>5</td>
<td>0.093404</td>
</tr>
<tr>
<td>6</td>
<td>0.091922</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.088128
Variance: 0.045467

<table>
<thead>
<tr>
<th>( \sigma = 0.55 )</th>
<th>( \text{Average CPU Time(sec)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R )</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.998498</td>
</tr>
<tr>
<td>2</td>
<td>0.195550</td>
</tr>
<tr>
<td>3</td>
<td>0.094259</td>
</tr>
<tr>
<td>4</td>
<td>0.061623</td>
</tr>
<tr>
<td>5</td>
<td>0.055988</td>
</tr>
<tr>
<td>6</td>
<td>0.053777</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.053949
Variance: 0.028945

Monte Carlo Timing: 117.19
Figure A.10: Up and Out Barrier Call Option: Ornstein Uhlenbeck Process

(a) Volatility=0.35

(b) Volatility=0.45

(c) Volatility=0.55
Table A.11: Up and Out Barrier Call Option: Cox Ingersoll Ross Process

Parameters: $R = [1,...,5]$, $S_0 = 1.15$, $K = 0.50$, $T = 2$, $H = 1.50$, $\gamma = 1$, $\theta = 1.05$, $\sigma = [0.30,...,0.50]$

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\sigma = 0.30$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>UB</td>
<td>LB</td>
<td>UB−LB</td>
<td>Error</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.834965</td>
<td>0.002297</td>
<td>0.832668</td>
<td>198.90</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.550264</td>
<td>0.184246</td>
<td>0.366018</td>
<td>99.66</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.405373</td>
<td>0.313339</td>
<td>0.092034</td>
<td>25.61</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.358777</td>
<td>0.346820</td>
<td>0.011956</td>
<td>3.39</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.352167</td>
<td>0.351903</td>
<td>0.000265</td>
<td>0.08</td>
<td></td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.369265
Variance: 0.079611

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\sigma = 0.35$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>UB</td>
<td>LB</td>
<td>UB−LB</td>
<td>Error</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.877167</td>
<td>0.000002</td>
<td>0.877164</td>
<td>200.00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.559766</td>
<td>0.127303</td>
<td>0.432373</td>
<td>125.88</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.364631</td>
<td>0.258106</td>
<td>0.106525</td>
<td>34.21</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.277586</td>
<td>0.275032</td>
<td>0.002555</td>
<td>0.92</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.273758</td>
<td>0.273392</td>
<td>0.000366</td>
<td>0.13</td>
<td></td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.305175
Variance: 0.078088

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\sigma = 0.40$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>UB</td>
<td>LB</td>
<td>UB−LB</td>
<td>Error</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.912693</td>
<td>0.000000</td>
<td>0.912693</td>
<td>200.00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.554639</td>
<td>0.081742</td>
<td>0.472897</td>
<td>148.62</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.239578</td>
<td>0.206799</td>
<td>0.032780</td>
<td>14.69</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.215333</td>
<td>0.213127</td>
<td>0.002206</td>
<td>1.03</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.213191</td>
<td>0.212760</td>
<td>0.000432</td>
<td>0.20</td>
<td></td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.235876
Variance: 0.076771

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\sigma = 0.45$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>UB</td>
<td>LB</td>
<td>UB−LB</td>
<td>Error</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.956570</td>
<td>0.000000</td>
<td>0.956569</td>
<td>200.00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.419046</td>
<td>0.046410</td>
<td>0.372636</td>
<td>160.12</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.194704</td>
<td>0.162883</td>
<td>0.031821</td>
<td>17.80</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.170186</td>
<td>0.167916</td>
<td>0.002270</td>
<td>1.34</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.165420</td>
<td>0.165835</td>
<td>-0.000415</td>
<td>-0.25</td>
<td></td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.182868
Variance: 0.065750

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\sigma = 0.50$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>UB</td>
<td>LB</td>
<td>UB−LB</td>
<td>Error</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.995744</td>
<td>0.000000</td>
<td>0.995743</td>
<td>200.00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.322455</td>
<td>0.020030</td>
<td>0.302425</td>
<td>176.61</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.160630</td>
<td>0.127846</td>
<td>0.032785</td>
<td>22.73</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.136965</td>
<td>0.134405</td>
<td>0.002560</td>
<td>1.89</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.132063</td>
<td>0.132405</td>
<td>-0.000342</td>
<td>-0.26</td>
<td></td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.161365
Variance: 0.063197

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average CPU Time(sec)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>UB</td>
<td>LB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.432276</td>
<td>0.432276</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.277166</td>
<td>0.277166</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.606049</td>
<td>0.606049</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.787318</td>
<td>1.787318</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5.574232</td>
<td>5.574232</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Monte Carlo Timing: 81.99
Figure A.11: Up and Out Barrier Call Option: Cox-Ingersoll-Ross Process

(a) Volatility=0.30

(b) Volatility=0.40

(c) Volatility=0.50
### A.2 SDP Moment Approach for Pricing Exotic Options

**Table A.12:** Double Barrier Call Option: Ornstein Uhlenbeck Process

Parameters: $R = [1,...,8]$, $S_0 = 1.25$, $K = 1$, $T = 2$, $H_l = 0.75$, $H_u = 1.50$, $\gamma = 1$, $\theta = 0.95$, $\sigma = [0.10,...,0.45]$ 

<table>
<thead>
<tr>
<th>$\sigma = 0.10$</th>
<th>$\sigma = 0.20$</th>
<th>$\sigma = 0.30$</th>
<th>$\sigma = 0.40$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$</td>
<td>UB</td>
<td>LB</td>
<td>UB−LB</td>
</tr>
<tr>
<td>1</td>
<td>0.105297</td>
<td>0.000001</td>
<td>0.105296</td>
</tr>
<tr>
<td>2</td>
<td>0.041657</td>
<td>0.000002</td>
<td>0.041655</td>
</tr>
<tr>
<td>3</td>
<td>0.031195</td>
<td>0.001756</td>
<td>0.029439</td>
</tr>
<tr>
<td>4</td>
<td>0.026624</td>
<td>0.005442</td>
<td>0.021182</td>
</tr>
<tr>
<td>5</td>
<td>(0.0214)</td>
<td>(0.0159)</td>
<td>0.005524</td>
</tr>
<tr>
<td>6</td>
<td>(0.0217)</td>
<td>(0.0199)</td>
<td>0.001792</td>
</tr>
<tr>
<td>7</td>
<td>(0.0218)</td>
<td>(0.0214)</td>
<td>0.000416</td>
</tr>
<tr>
<td>8</td>
<td>(0.0235)</td>
<td>(0.0232)</td>
<td>0.000262</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.023363

Variance: 0.001419

Monte Carlo Simulation: 0.045102

Variance: 0.008038

<table>
<thead>
<tr>
<th>$\sigma = 0.45$</th>
<th>Average CPU Time(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.310923</td>
</tr>
<tr>
<td>2</td>
<td>0.116218</td>
</tr>
<tr>
<td>3</td>
<td>0.042420</td>
</tr>
<tr>
<td>4</td>
<td>0.022578</td>
</tr>
<tr>
<td>5</td>
<td>(0.0172)</td>
</tr>
<tr>
<td>6</td>
<td>(0.0145)</td>
</tr>
<tr>
<td>7</td>
<td>(0.0128)</td>
</tr>
<tr>
<td>8</td>
<td>(0.0135)</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.005613

Variance: 0.001439

Monte Carlo Simulation: 0.005613

Variance: 0.001439
Figure A.12: Double Barrier Call Option: Ornstein Uhlenbeck Process
A.2 SDP Moment Approach for Pricing Exotic Options

Table A.13: Double Barrier Call Option: Cox Ingersoll Ross Process

| Parameters: R = [1,...,6], S_0 = 1.1, K = 1, T = 2, H_l = 0.75, H_u = 1.5, γ = 1, θ = 1.2, σ = [0.30,...,0.50] |

<table>
<thead>
<tr>
<th>σ = 0.30</th>
<th>σ = 0.35</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.475269</td>
</tr>
<tr>
<td>2</td>
<td>0.211354</td>
</tr>
<tr>
<td>3</td>
<td>0.041026</td>
</tr>
<tr>
<td>4</td>
<td>(0.0402)</td>
</tr>
<tr>
<td>5</td>
<td>(0.0345)</td>
</tr>
<tr>
<td>6</td>
<td>(0.0359)</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.054776
Variance: 0.012251

Monte Carlo Simulation: 0.030664
Variance: 0.007377

<table>
<thead>
<tr>
<th>σ = 0.40</th>
<th>σ = 0.45</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.497406</td>
</tr>
<tr>
<td>2</td>
<td>0.127103</td>
</tr>
<tr>
<td>3</td>
<td>0.020803</td>
</tr>
<tr>
<td>4</td>
<td>(0.0081)</td>
</tr>
<tr>
<td>5</td>
<td>(0.0075)</td>
</tr>
<tr>
<td>6</td>
<td>(0.0077)</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.013827
Variance: 0.003399

Monte Carlo Simulation: 0.006815
Variance: 0.002119

<table>
<thead>
<tr>
<th>σ = 0.50</th>
<th>Average CPU Time(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>UB</td>
</tr>
<tr>
<td>1</td>
<td>0.490928</td>
</tr>
<tr>
<td>2</td>
<td>0.073091</td>
</tr>
<tr>
<td>3</td>
<td>0.015973</td>
</tr>
<tr>
<td>4</td>
<td>(0.0031)</td>
</tr>
<tr>
<td>5</td>
<td>(0.0015)</td>
</tr>
<tr>
<td>6</td>
<td>(0.0012)</td>
</tr>
</tbody>
</table>

Monte Carlo Simulation: 0.002845
Variance: 0.000839

Monte Carlo Timing: 82.66
Figure A.13: Double Barrier Call Option: Cox Ingersoll Ross Process
Bibliography


