

PRICING AMERICAN OPTIONS WITH JUMP-DIFFUSION BY MONTE CARLO
SIMULATION

by

BRADLEY WARREN FOUSE

B.S., Kansas State University, 2009

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Industrial and Manufacturing Systems Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2009

Approved by:

Major Professor
Dr. Chih-Hang Wu

Abstract

In recent years the stock markets have shown tremendous volatility with significant spikes and drops in the stock prices. Within the past decade, there have been numerous jumps in the market; one key example was on September 17, 2001 when the Dow industrial average dropped 684 points following the 9-11 attacks on the United States. These evident jumps in the markets show the inaccuracy of the Black-Scholes model for pricing options. Merton provided the first research to appease this problem in 1976 when he extended the Black-Scholes model to include jumps in the market. In recent years, Kou has shown that the distribution of the jump sizes used in Merton's model does not efficiently model the actual movements of the markets. Consequently, Kou modified Merton's model changing the jump size distribution from a normal distribution to the double exponential distribution.

Kou's research utilizes mathematical equations to estimate the value of an American put option where the underlying stocks follow a jump-diffusion process. The research contained within this thesis extends on Kou's research using Monte Carlo simulation (MCS) coupled with least-squares regression to price this type of American option. Utilizing MCS provides a continuous exercise and pricing region which is a distinct difference, and advantage, between MCS and other analytical techniques. The aim of this research is to investigate whether or not MCS is an efficient means to pricing American put options where the underlying stock undergoes a jump-diffusion process. This thesis also extends the simulation to utilize copulas in the pricing of baskets, which contains several of the aforementioned type of American options. The use of copulas creates a joint distribution from two independent distributions and provides an efficient means of modeling multiple options and the correlation between them.

The research contained within this thesis shows that MCS provides a means of accurately pricing American put options where the underlying stock follows a jump-diffusion. It also shows that it can be extended to use copulas to price baskets of options with jump-diffusion. Numerical examples are presented for both portions to exemplify the excellent results obtained by using MCS for pricing options in both single dimension problems as well as multidimensional problems.

Table of Contents

List of Figures	v
List of Tables	vi
Acknowledgements	vii
Dedication	viii
CHAPTER 1 - INTRODUCTION.....	1
1.1 Introduction.....	1
1.2 Research Motivations	5
1.3 Research Objectives & Contributions.....	6
1.4 Outline	7
CHAPTER 2 - LITERATURE REVIEW	9
2.1 Evaluating American Options.....	9
2.1.1 Partial-Differential Equations	9
2.1.2 Lattice Methods	10
2.1.3 Stochastic Meshes	12
2.1.4 Monte Carlo Simulation.....	16
2.2 Modeling of the Jump Diffusion Processes	17
2.2.1 Merton's Model	19
2.2.2 Kou's Model	20
2.2.2.1 Quadratic Approximation Algorithm.....	23
2.2.2.2 Piecewise Exponential Approximation.....	24
2.3 Copulas	25
CHAPTER 3 - SIMULATING AMERICAN OPTIONS WITH JUMP DIFFUSION	30
3.1 Background on Least-Squares Regression	30
3.1.1 Notation & Algorithm.....	31
3.2 A Jump-Diffusion Simulation Model	34
3.2.1 Notation.....	35
3.2.2 Updated Algorithm	36
3.2.3 Characteristics of Simulation Parameters	38

3.3 Chapter Conclusions	41
CHAPTER 4 - GENERATING SIMULATION DATA	42
4.1 Uniform Random Number Generator	42
4.2 Normal Random Number Generator.....	47
4.3 Copula Generator	49
CHAPTER 5 - COMPUTATIONAL RESULTS	50
5.1 Design of Experiment	50
5.2 Results.....	51
CHAPTER 6 - EXTENSION TO BASKETS	57
6.1 A Jump-Diffusion Simulation Model with Copulas	57
6.1.1 Cholesky Decomposition	57
6.1.2 Notation.....	58
6.1.3 Updated Algorithm	58
CHAPTER 7 - COMPUTATIONAL RESULTS FOR CORRELATED BASKETS	62
7.1 Copula Design of Experiment.....	62
7.2 Copula Dynamics.....	63
7.3 Copula Results	64
CHAPTER 8 - MEMORY REQUIREMENTS	67
8.1 Memory Requirements for Simulation Models	67
CHAPTER 9 - CONCLUSIONS	68
9.1 Conclusion on Monte Carlo Simulation	68
9.2 Future Work.....	71
CHAPTER 10 - BIBLIOGRAPHY	73
Appendix A - Derivation of the Roots of $G(x)$	77

List of Figures

Figure 2-1: Binomial Tree	11
Figure 2-2: Stochastic Mesh Grid	12
Figure 2-3: Implicit (left) and Explicit (right) Methods	13
Figure 2-4: Random Variate Generation.....	26
Figure 3-1: Simulated Path with Jumps	34
Figure 3-2: Simulated Paths 2.....	35
Figure 3-3: Input 1	40
Figure 3-4: Input 2	40
Figure 3-5: Input 3	40
Figure 3-6: Input 4	40
Figure 3-7: Input 5	40
Figure 3-8: Input 6	40
Figure 4-1: Fitted Histogram of $\mathcal{N}(0,1)$	48
Figure 4-2: Normality Test	48
Figure 7-1: Stock 1 High Correlation	65
Figure 7-2: Stock 1 Low Correlation	65
Figure 7-3: Stock 2 High Correlation	65
Figure 7-4: Stock 2 Low Correlation	65
Figure 7-5: Stock 3 High Correlation	65
Figure 7-6: Stock 3 Low Correlation	65

List of Tables

Table 1: Parameter Descriptions.....	38
Table 2: U(0,1) Test Results	47
Table 3: Design of Experiment Levels	51
Table 4: Monte Carlo simulation comparison for American put option with $t = 0.25$ Year	53
Table 5: λ Significance Testing 1	55
Table 6: λ Significance Testing 2	55
Table 7: Monte Carlo simulation comparison for American put option with $t = 1.0$ Year	56
Table 8: Copula Design of Experiment	63
Table 9: Copula Results	66

Acknowledgements

I would like to take a moment to thank my major professor Dr. John Wu for all of his help in getting my thesis completed. Dr. Wu was an invaluable source with the insight and recommendations he provided throughout my research. I would also like to sincerely thank Dr. Wu and the rest of the Kansas State University Industrial and Manufacturing Systems Engineering faculty and staff for the kindness and support they have provided over the past five years. I will always remember the relationships I built with these individuals while I was a student in the IMSE department.

Dedication

I would like to dedicate this thesis to my parents (Dennis & Donna Fouse), brother (Jeremy Fouse), grandmas (Cathy Brungardt & Jo Fouse), my late grandpa (Warren Fouse), and the rest of my family and friends. Without their continuous love and support I would not be the person I am today. For all that they have taught me I am forever grateful.

CHAPTER 1 - INTRODUCTION

1.1 Introduction

As the financial markets have evolved during the recent years, so has the desire to be able to accurately calculate the expected future worth of the various financial assets and derivatives on the market. To satisfy this desire a new technical field, known as Financial Engineering or Quant Mathematics, has emerged that integrates the methods and tools utilized by financial analysts, mathematicians, and engineers to calculate the expected worth of financial assets and derivatives.

Financial Engineering

Financial Engineering focuses on a multitude of different financial instruments (e.g. assets and/or their derivatives). Originally these included asset management, portfolio optimization, risk assessment, and hedging, but have more recently focused on Collateralized Debt Obligations (CDOs) and the various types of stock options. Many of the CDOs only work well under certain market settings, however many of them can cause catastrophic impacts when the markets deviate from such settings. Options and futures on the other hand are still one of the most viable financial instruments regardless of the general market trends.

Options

The research contained within this thesis focuses on the stock options portion of Financial Engineering. An option is a contract, purchased for a premium, between the buyer and the seller that gives the purchaser the *right* – but not the *obligation* – to buy or sell the underlying asset at a future date. There is a vast variety of options available in the financial markets today with the two most common types being the European and American options. Before explaining the difference between the two types, it is important to note the key factors involved with stock options. The basic elements of an option are (1) the expiration date, which is the point at which the option can no longer be exercised, (2) the strike price, which is the amount for which the underlying stock will be purchased or sold, and (3) whether the option type is a call or put. A call option gives the purchaser the right to buy the underlying stock at the strike price and a put

option gives the purchaser the right to sell the underlying stock. Every style of option requires these three factors.

The most distinct difference between the two aforementioned types of options, the European and American option, is when the option can be exercised. A European option gives the buyer the right to exercise the option *on* the expiration date whereas an American option gives the buyer the right to exercise at *any* point up to, and including, the expiration date. Besides the ability to exercise at any point, the two options are identical in that each can be purchased as a call or a put and the gains for each are calculated by one of the following equations depending on the type:

- Put Option: $\text{Max}\{ \text{Strike Price} - \text{Market Price}, 0 \}$
- Call Option: $\text{Max}\{ \text{Market Price} - \text{Strike Price}, 0 \}$

American Options

In general, American options are more flexible (in terms of when an option can be exercised) than the European option, and therefore, the price of an American option is usually higher than that of a European option with the same key elements described above. The added flexibility of American options, however, makes the analysis of such options much more complicated. Since American options have a continuous or quasi-continuous exercise region, determining the optimal exercise point and the expected worth of the option is computationally challenging. As a result, this type of option has been extensively studied by financial market practitioners and within the academic communities. Through these studies, a variety of different approaches have been developed to estimate, or approximate, the expected value of the American option: (a) Dual based and approximate dynamic programming methods to find upper and lower bounds, as seen in Haugh (2004); (b) least-squares approach as seen in Longstaff and Schwartz (2001); (c) stochastic meshes as seen in Broadie and Glasserman (2004), Achdou and Pironneau (2005), and Zhang (2005). Each of these methods is effective in pricing American options, however, one common problem with several of the methods is when modifying the pricing technique to incorporate options where the underlying stock follows a jump-diffusion process.

Jump-Diffusion

In 1976, R. C. Merton began addressing the phenomena of “price jumps” in the stock markets, which has been termed the jump-diffusion process. When a stock price follows a jump-diffusion process, it means that the randomness can be split into two separate types of processes; these processes include the jump process and the diffusion process. Prior to Merton’s findings, it was assumed that the stock markets only followed the geometric Brownian motion process (a type of diffusion process) in that over a short enough period of time the stock price can only change by a small amount. This is one of the key assumptions of the Black-Scholes formula, the very first mathematical model of the stock market developed by Fischer Black and Myron Scholes in 1973, and is the basis for which each of the methods listed above approximate the expected worth of the American option.

Merton, however, observed that the changes in the stock prices were not always in small and continuous steps. In many instances, either when there is an announcement of market events or some type of human intervention, the stock prices have shown immediate and substantial spikes or dips in the prices. Stated in a more scientific sense, it is quite evident that there are short term instances in the market that the Black-Scholes model cannot explain (e.g. outliers). In recent years these dips are evident after events like the 9-11 attacks on the United States when the DOW industrial average dropped 684 points on the first day trading resumed (September 17, 2001) and the wars on terrorism that followed (Kauffman 2001). A second obvious instance is the multiple bailout plans that were established during President Bush’s tenure for the automakers and financial industries when the DOW decreased 14.1% in October 2008 (Steverman 2008). Each of these events caused an immediate, and significant, “price jump” in the stock market that would not normally have been possible if the markets wholly followed a diffusion process.

Since Merton (1976) introduced the idea of the jump-diffusion process, additional research activity has focused on the possible “jumps” in the market in order to find methods that incorporate these jumps into the pricing of options. Kou (2002, 2004, and 2008) have some significant contributions to this area. Throughout Kou’s research he presented a model very similar to the model proposed by Merton (1976), with the major difference between the two models being how the jump sizes are modeled. Merton believed that the size of the jumps follow a normal distribution and Kou on the other hand, after noticing a few distinct key problems with

the normal distribution, believed that the jump process can be better explained using a double exponential distribution. Kou 2008 (Figure 2) showed how the normal distribution does not accurately model the market behaviors by presenting the leptokurtic nature of the market jumps and the dissimilarity between the historical data and the normal distribution. The historical data shows the inaccuracy of the normal distribution by comparing it to the jump size distribution and presenting the fact that the jump size distribution has a much higher peak and fatter tails than that of a normal distribution.

Kou (2008) provided two analytical approximations for pricing American options without dividends. Namely, he extends the Barone-Adesi and Whaley (1987) quadratic approximation and the piecewise exponential approximation presented by Ju (1998). Kou found that the piecewise exponential approximation provided better results at the cost of programmability and the time required to solve the problem. The quadratic approximation itself has three major deficiencies: (1) it is an approximation algorithm that bases the worth of an American option off of the worth of an equivalent European option; (2) the approximation algorithm contains a discrete pricing region so that the entire range of prices is not possible, and (3) the approximation algorithm does not indicate an optimal exercise point. Kou's approximation algorithm can only approximate the price of the option based on discrete pricing values determined by the value of a European option, but cannot indicate when the option should be optimally exercised.

These deficiencies, however, are common in several other numerical methods. Many of the methods used to evaluate American options are deficient in that they evaluate the option based on discrete pricing regions (e.g. m -nomial trees) or are extremely complicated due to the use of complex integral equations (e.g. the use of piecewise exponential approximation or Laplace Transformations). These issues can be addressed, and significantly streamlined, using Monte Carlo simulation techniques. By coupling Longstaff and Schwartz's (2001) least-squares regression with Monte Carlo simulation, researchers and market practitioners can accurately calculate the fair value (also referred to as the expected worth or the price) of an American option with jump-diffusion with a simple algorithm that provides a continuous pricing region and the optimal exercise point. Least-squares regression with Monte Carlo simulation provides a means to accurately approximate the two pieces of information that are required to analyze an American option: the fair value and the exercise point.

The research effort within this thesis extends on Kou's jump-diffusion model and utilizes Monte Carlo simulation to calculate the price and the optimal exercise time of an American option where the underlying asset follows a jump-diffusion process. Additionally, the research is extended to incorporate copulas into the pricing of baskets, which are small portfolios (usually less than five options) of any type of option.

1.2 Research Motivations

Generally, American options cannot be solved using closed-form mathematical formulas, even when all of the required distributions (e.g. distributions for the jump size and the frequency of jumps) are known. As a result, it is commonly recognized that American options can only be evaluated using numerical procedures and not analytical procedures. The addition of jump-diffusion processes further complicates this matter. In most cases when jumps are involved the exact distributions are unknown making it even more difficult to create a pricing formula. Also, in cases where the closed form approximated solution includes renewal integral equations and there is a presence of two-directional jumps – either up or down – unique solutions may not exist because of the difficulty of determining enough boundary conditions based on the renewal arguments alone. This fact motivates our research for using stochastic sampling to estimate the worth of American options. Since both stochastic sampling and complex integral equations are approximations, this leads to a very important question comparing the two types of numerical approximation methods (stochastic sampling via Monte Carlo simulation vs. complex integral equations like Laplace transformations): *Which method is better in regards to computational effort and modeling flexibility?*

Monte Carlo simulation is a versatile method for pricing options. As this research will demonstrate, modeling the addition of the jump-diffusion process only requires a few minor extensions to the simulation models proposed by DeHaven (2007). DeHaven (2007) presented a Monte Carlo simulation approach using the discrete event simulation program, Rockwell Software's Arena 10.0. However, her research did not extend the pricing to include either jumps in the markets or correlated multi-option baskets. The research performed within this thesis is based on the simulation model of DeHaven (2007) and extends it to include the jump-diffusion processes. One of the main reasons that the modifications are relatively straight forward is because the exact distributions of the jumps are not required to model American options with

jump-diffusion. In this thesis, we have shown that with the adequate use of stochastic sampling and simulation procedures the proposed method can provide very accurate results for pricing complex American options under various settings.

Another significant benefit of using Monte Carlo simulation is that it allows for a continuous pricing region, which in turn, can price American options with extreme accuracy. As previously mentioned this lack of continuous pricing region is a deficiency of many numerical approximation algorithms and is a problem that can be addressed using Monte Carlo simulation. DeHaven (2007) has proven the extreme accuracy of pricing options with Monte Carlo simulation through her comparisons of stochastic mesh pricing methods and least-squares pricing using Monte Carlo simulation. Her comparisons detail how the continuous pricing region is beneficial in improving the accuracy of pricing American options.

Kou (2004) presents two approximation algorithms for pricing American options with jump-diffusion, both of which are numerical approximations using either complex integral equations or approximation equations based on the value of a European option. Of the two heuristics presented, the quadratic approximation is the easiest to implement but provides less accurate pricing of American options. The piecewise exponential approximation, however, is much harder to implement due to the integrals associated with the process, but provides more accurate estimations. Motivated by the results presented by Kou (2004), this thesis will investigate the applicability of using Monte Carlo simulation to price American options with jump-diffusion. The research within this thesis builds on the findings of Kou and DeHaven to create a simulation model to evaluate American options where the underlying stock undergoes a jump-diffusion process and further extends the simulation model to a multidimensional domain.

1.3 Research Objectives and Contributions

The purpose of this study is to explore the effectiveness of pricing American options with jump-diffusion through the Monte Carlo simulation approach. This study shows the flexibility Monte Carlo simulation provides and relates the option price to the following input parameters: (1) initial stock price, (2) strike price, (3) option life span or expiration date, (4) risk-free interest rate, (5) stock volatility, (6) mean jump size, and (7) average number of jumps per year. The independent variables are the input parameters and the dependent variable is the price (i.e., the fair value) of the option.

Namely, this thesis seeks to explore if there is a significant difference between using Monte Carlo simulation and other numerical techniques to price American options with jump-diffusion. To answer this question, this thesis will compare the accuracy and effectiveness of a Monte Carlo simulation to that of the approximation algorithm Kou (2004) presents for finite-horizon American options. The Monte Carlo simulation will combine Kou's jump-diffusion model with the least-squares regression model presented by Longstaff and Schwartz (2001) and is programmed using the C++ programming language.

Additionally, the simulation model is extended to incorporate copulas in the pricing of baskets containing American options with jump-diffusion. A copula is a mathematical tool that combines several univariate distributions to create a joint distribution. The use of copulas allows for the correlation between options to be modeled so that industry-wide jumps can be accounted for.

From the research described above, the main contributions of this thesis are listed as follows:

- Explore the use of stochastic sampling techniques via a Monte Carlo simulation model to ascertain whether it provides a quick and accurate way of pricing American options where the underlying stock undergoes a jump-diffusion process.
- Extend the simulation model to incorporate copulas so that baskets, where their underlying stocks undergo a jump-diffusion process, can also be analyzed.

1.4 Outline

The remainder of the thesis is organized as follows. Chapter 2 presents a literature review of the current research efforts to price American options with and without jump-diffusion and Chapter 3 details one of the two main methods and research tasks performed within this thesis which includes the simulation of American options with jump diffusion. This chapter provides detailed information regarding the methods followed, algorithms proposed, and the general characteristics of the jump-diffusion process. Chapter 4 summarizes the validation of the various random variates used in the computational experiments. Chapter 5 presents the results of the research and Chapter 6 extends the methods explained in Chapter 3 to incorporate copulas to price baskets of American options. Chapter 7 presents the results of this extension, Chapter 8

discusses memory requirements of the least-squares algorithms and Chapter 9 summarizes this thesis and provides directives for future research.

CHAPTER 2 - LITERATURE REVIEW

This chapter reviews the existing literature that is related to the area of study for this thesis. Namely, this chapter looks at the state-of-the-art methods for pricing American options with and without jump-diffusion. Section 2.1 introduces several of the most commonly used methods for American options without jump-diffusion and Section 2.2 discusses the methods presented by S.G Kou and R.C. Merton for calculating American options where the underlying stock price follows a jump-diffusion process. In Section 2.3 the existing works for modeling correlated behaviors on equities or assets using the concept of copulas are presented. This modeling technique is applied to price a basket of correlated American options in Chapter 6.

2.1 Evaluating American Options

Of the two common types of options on the markets today, American and European, the American option is much more complex to evaluate. Not only does the purchaser need to determine the optimal exercise policy (i.e. when the option should be exercised), but the option price must also be determined. The Black and Scholes model is an explicit closed-form pricing formula for European options without dividends. Unfortunately, unlike in the European case, explicit closed-form solutions for American option pricing problems are not generally attainable. As a result when exact formulations cannot be obtained or are too difficult to implement, numerical evaluation methods are frequently the preferred choice to price such options. The existing numerical evaluation methods can be divided into four main categories including the Partial-Differential Equation (PDE) based methods, lattice methods, stochastic mesh methods, and simulation-based methods. Each of the subsequent subsections review the four methods and discuss the advantages and disadvantages of each method.

2.1.1 Partial-Differential Equations

The most significant advancements for the PDE based methods in recent literature have been in the applications of domain transformation and asymptotic expansion techniques. In particular, Fourier, Laplace, and generalized transformation methods have been applied to stochastic volatility models and many other pricing models (Broadie and Detemple, 2004).

These techniques can also be applied for stock pricing models involving jump-diffusion processes as evident in Kou (2005). Due to the fact that PDE based methods use various approximation techniques, they are usually considered as less precise and in many instances more complicated in the implementation and calculation of the stock prices, and are therefore not nearly as popular as other pricing methods. The PDE approaches are only briefly mentioned in this section. A more extensive review of PDE approaches involved in the jump-diffusion domain will be presented and explained in Section 2.2.

2.1.2 Lattice Methods

Lattice methods use discrete-time and discrete-state approximations of differential equations to price American options. These methods are more commonly referred to as m -nomial trees; e.g. $m=2$ for the binomial tree method, $m=3$ for the trinomial tree method, etc. In general, lattice methods are easy to implement for simple models but become much less accurate as the complexity of the model increases and are therefore not commonly used.

Figure 2-3, below shows the structure of the lattice method with $m=2$, a binomial tree. Each point represents a possible stock price level. The tree starts at period 0 with a single point (the initial stock price). There is then a probability p_1 that the stock price increases and a probability p_2 that the stock price decreases in the next period. This procedure continues for each subsequent period, creating a binomial tree that continuously spreads out at a rate of $2t$ where t is the number of periods within the time horizon. In the case of a binomial tree, there are a total of two possible outcomes for each consecutive step, a trinomial tree has three possible outcomes, and an m -nomial tree has m possible outcomes. As the number of m increases, so do the computational requirements at each period. As Broadie and Detemple (2004) indicated, the improvement in pricing accuracy using $m>2$ does not outweigh the increased computational costs via increasing the number of pricing periods. Broadie and Detemple (2004) also mentioned that values of $m>2$ have not resulted in better overall convergence when the additional computational efforts are considered.

Lattice methods were first proposed for financial engineering applications by Cox et al. (1979). There have been numerous research proposals for lattice methods, however the four most widely used lattice approximations are those described in Cox et al. (1979), Jarrow and Rudd (1982), Boyle (1986), and Amin (1991). Each article focuses on a binomial tree and each

provides a different means for calculating the probability of the stock price increasing and decreasing. For example, Cox et al (1979) used $p_1 = \frac{e^{(r-\delta)h} - e^{-\sigma\sqrt{h}}}{e^{\sigma\sqrt{h}} - e^{-\sigma\sqrt{h}}}$, $p_2 = 1 - p_1$ and Amin (1991) used that $p_1=0.5$, $p_2=0.5$ at each period.

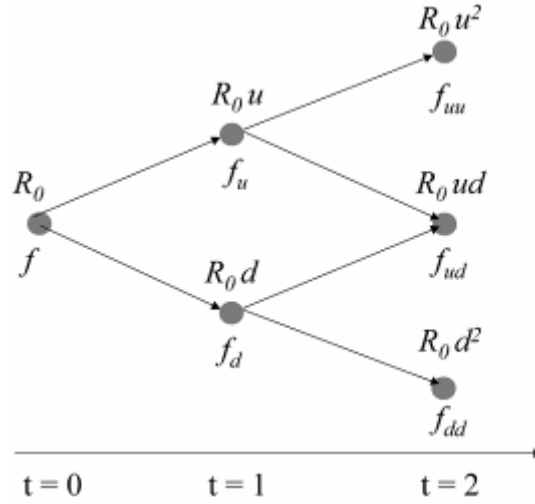


Figure 2-1: Binomial Tree

The popularity of lattice methods is due to its conceptual simplicity and ease of implementation. However, a major problem with this method is that the number of possible stock prices is very limited in the initial time periods. When looking at the first two periods, with $m=2$, there are a total of six possible stock prices – this number increases to 11 possible prices with $m=3$. Comparing this to stochastic meshes (which is covered in the subsequent section), it is quite evident that there is a significant difference in the number of possible stock prices in the initial stages which can lead to inaccurate results when using lattice methods. Additionally, when comparing lattice methods to Monte Carlo simulation techniques, there is a distinct advantage of using Monte Carlo simulation when computational effort versus accuracy is considered. For example, if a simulation of 200 paths is generated and is compared to a binomial tree, the simulation has a significant advantage since there are, in essence, 200 possible pricing nodes in the first period (compared to two nodes in the first period of the binomial tree). To obtain 200 possible price nodes within the first period using the binomial tree method, the first period (Δt) must be split into 100 separate segments. In other words, it requires a tree in which the first period is split up to contain $t=100$ periods ($t=2$ is shown in Figure 2-3 above) in order to obtain the same number of possible prices as a Monte Carlo simulation with 200 generated paths.

It is obvious that Monte Carlo simulation has a distinct advantage when looking at the computational effort versus accuracy comparison.

2.1.3 Stochastic Meshes

Stochastic meshes are used to price American options by using discrete-time and discrete-state approximations and are utilized when closed-form solutions are not obtainable. Stochastic meshes can be split up into two distinct methods: finite difference method and finite element method. In both approaches, a mesh must be created to represent the discretization of the time vs. stock price space, as shown in Figure 2.1. A stochastic mesh is a grid in which each point represents a stock price in a discrete time period. In general, the points within the mesh are equally spaced so that the change in time is $\Delta t = T/N$ and the change in stock price is $\Delta S = S_{max}/Q$. As evident in Figure 2.1, below, there is a total of $(N+1)(Q+1)$ points because there are a total of $N+1$ time periods and $Q+1$ stock prices (DeHaven 2007).

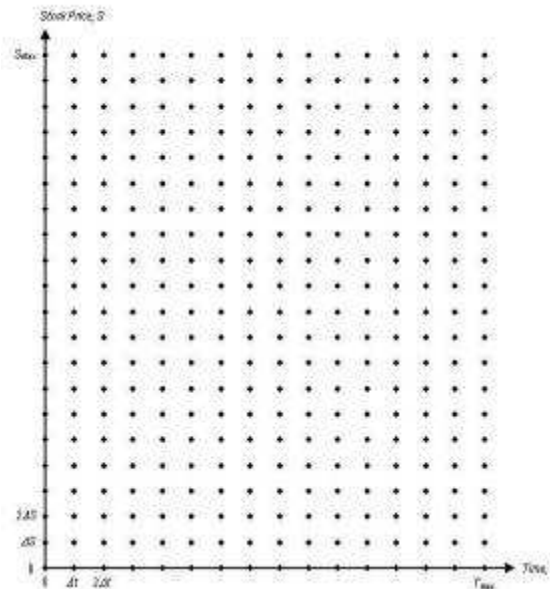


Figure 2-2: Stochastic Mesh Grid

As previously mentioned, there are two methods for solving American options using a stochastic mesh. Each of these methods is explained next.

Finite Differences Method

The finite differences method was first presented by Brennan and Schwartz (1977) and was later presented in the area of financial engineering by Hull and White (1990), Wilmott

(1998), Achdou and Pironneau (2005), and Hull (2006). Each of these authors use either explicit or implicit finite differences methods to provide numerical solutions to PDEs and show that in general, the finite differences method is one of the simplest ways to approximate a differential equation. For this reason, the finite differences method is widely used for models and securities that are more complex. This method is based on the expression $f(x+b) - f(x+a)$ which implies that the next point is derived from its predecessor using either backward or forward recursion – which are two possible ways to solve American options using finite differences methods. Figure 2-2 shows the difference in the way these two procedures are followed. The explicit method (or backward recursion) relates the value at time t to the three alternative values at time $t+\Delta t$, or as shown in the graph time i to time $i+1$. The implicit method (or forward recursion) relates the value at time $t+\Delta t$ to three alternative values at time t . The explicit method is equal to a trinomial lattice approach and the implicit method is equivalent to a multinomial lattice (Hull and White 1990).

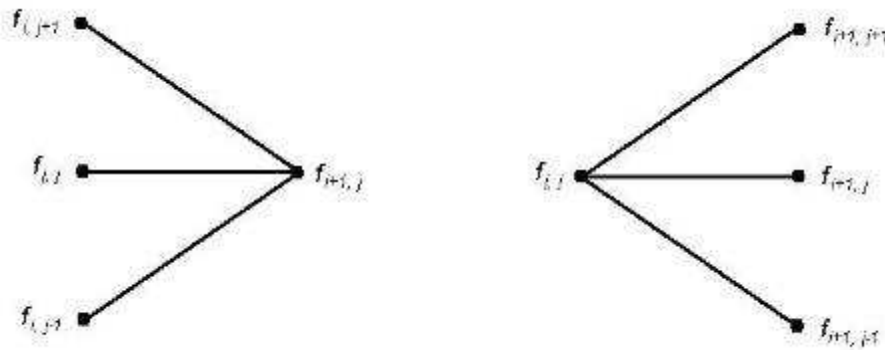


Figure 2-3: Implicit (left) and Explicit (right) Methods

As DeHaven (2007) and Hull and White (1990) explain, for the finite difference method the American put option must satisfy the equation

$$\frac{\partial f}{\partial t} + rS \frac{\partial f}{\partial S} + 0.5\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = rf \quad (2.1)$$

where the standard notation for the Black and Scholes model is followed; e.g. r is the risk-free interest rate, S is the stock price, and σ is the stock volatility. Equation (2.2) is the partial derivative with respect to the stock price, equation (2.3) is the second derivative with respect to the stock price, and equation (2.4) is the partial derivative with respect to time.

$$\frac{\partial f}{\partial S} = \frac{f_{i,j+1} - f_{i,j-1}}{2\Delta S}. \quad (2.2)$$

$$\frac{\partial^2 f}{\partial S^2} = \frac{f_{i,j+1} + f_{i,j-1} - 2f_{i,j}}{\Delta S^2}. \quad (2.3)$$

$$\frac{\partial f}{\partial t} = \frac{f_{i+1,j} - f_{i,j}}{\Delta t}. \quad (2.4)$$

By substituting each of these 3 equations back into equation (2.1), an equation can be created that defines all the interior points for the stochastic mesh using the implicit method. This equation is presented as follows:

$$a_j f_{i,j-1} + b_j f_{i,j} + c_j f_{i,j+1} = f_{i+1,j} \quad (2.5)$$

where

$$a_j = 0.5rj\Delta t - 0.5\sigma^2 j^2 \Delta t$$

$$b_j = 1 + \sigma^2 j^2 \Delta t + r\Delta t$$

$$c_j = -0.5rj\Delta t - 0.5\sigma^2 j^2 \Delta t.$$

The values for the explicit method are obtained in a similar way where the following equation is substituted for equation (2.4):

$$\frac{\partial f}{\partial t} = \frac{f_{i,j} - f_{i-1,j}}{\Delta t}.$$

From this, equation (2.5) is modified so that

$$a_j^* f_{i+1,j-1} + b_j^* f_{i+1,j} + c_j^* f_{i+1,j+1} = f_{i,j} \quad (2.6)$$

where

$$a_j^* = \frac{1}{1+r\Delta t} \left[-0.5rj\Delta t + 0.5\sigma^2 j^2 \Delta t \right],$$

$$b_j^* = \frac{1}{1+r\Delta t} \left[1 - \sigma^2 j^2 \Delta t \right], \text{ and}$$

$$c_j^* = \frac{1}{1+r\Delta t} \left[0.5rj\Delta t + 0.5\sigma^2 j^2 \Delta t \right].$$

The stochastic mesh created by (2.5) and (2.6) is bounded by the following boundary conditions that are required to retain feasibility:

1. The value at expiration T is equal to $f_{N,j} = \max(K - j\Delta S, 0)$

2. The value when $S=0$ is equal to $f_{i,0} = K$
3. The put value is equal to zero when $S=S_{\max}$

Comparing the explicit to the implicit method, the explicit method is conceptually simpler and is easier to implement. Additionally, Hull and White (1990) indicate that the explicit finite difference method uses 40 to 70 percent as much CPU computation time as the implicit method to obtain the same level of accuracy.

As DeHaven (2007) showed in her research results, the Monte Carlo simulation provides a better means of pricing options in regards to price and both computational time requirements and memory requirements compared to the finite differences method. She found that the least-squares regression method (which will be discussed in detail in Chapter 3 and is a large portion of this research) consistently resulted in higher returns than that of the finite differences method. DeHaven's research also shows that the finite differences method has a running time of $O(Q^{N^3} + N^2M)$, where N is the number of time periods, M is the number of paths simulated, and Q represents the number of stock price intervals. The running time for Monte Carlo simulation coupled with least-squares regression is $O(NM)$ where N and M are the same parameters as previously stated. It is obvious that the running time of the finite differences method grows much quicker than that of the least-squares regression, indicating a major advantage of Monte Carlo simulation coupled with least-squares regression.

Additionally, DeHaven's research shows a significant difference between the two methods in terms of memory requirements. She shows that the memory required for least-squares regression grows in the order of $O(M)$, where M is the number of paths generated, whereas the finite differences method grows in the order of $O((N+1)(Q+1))$, where N is the number of time periods and Q is the number of stock price intervals. Within her discussion, DeHaven provides an example of a simulation with 200 paths and 400 time periods in which she shows that using least-squares regression requires about 0.74% of the memory required for the finite differences method.

Finite Element Method

As previously stated, both of these methods are used to approximate a PDE model when a closed form solution is not available. The difference between these two methods lies in what the procedure approximates. As opposed to the finite differences method, which approximates the

actual differential equation, the finite elements method approximates the *solution* of the differential equation. This method will attempt to either eliminate the PDE or convert it to a standard differential equation that can be solved using standard techniques. This thesis will not go into great detail to explain this method due to its unpopularity within the field of financial engineering. This method is not widely used since it is more complicated mathematically, requires substantially more memory than other methods and the results obtained do not warrant the extra work that is required.

2.1.4 Monte Carlo Simulation

The last category of methods for evaluating American options to be presented is using Monte Carlo simulation to price American options. This method replaces the continuous exercises region of an American option with discrete time periods, usually denoted by N , which is very similar to the previously discussed methods. Though discrete time periods are created, a major difference between Monte Carlo simulation and the other methods presented thus far is that the pricing region remains continuous. This is an enormous advantage over other pricing methods and in turn produces very accurate results. Additionally, Monte Carlo simulation is useful in that a closed-form evaluation of the stock prices is not required. Therefore, Monte Carlo methods tend to be used when it is infeasible, if not computationally impossible, to compute an exact result with a mathematical formula. The ability to model systems – which in the scope of this research, the systems are the underlying stock prices – stemmed from the use of random variates to simulate the stochastic nature of the stock prices. This method is typically used when a model is extremely complex, nonlinear, or involves more than just a few controlling parameters.

As Charnes (2000) explains, these complex models tend to contain high-dimensional integrals. Monte Carlo simulation becomes attractive in these cases due to its flexibility, which will be shown within this thesis with the addition of modeling a jump-diffusion process and multi-option baskets. Its ease of implementation and modification and the fact that the error convergence rate is independent of the dimension of the problem make the Monte Carlo simulation approach a well suited tool for the complex option pricing problems. Namely, the error rate is of the magnitude $1/\sqrt{C}$, where C is the number of paths generated. However, the

error rate can also be viewed as a detriment to Monte Carlo simulation because of the fact that more replications (paths) must be performed in order to reduce the amount of error in the results.

Monte Carlo simulation was first applied to the financial markets by Boyle (1977). Recently this area of study has exploded within the research community, especially in applying Monte Carlo simulation to the pricing of American options. Due to the complexity of determining both the exercise point and the option price, only a handful of closed-form equations have been created. Most of these, however, only work in limited circumstances or under certain unrealistic assumptions. The first to price American options using Monte Carlo simulation was credited to Tilley (1993). However, his methodology was memory intensive and grows in the order of $O(MN)$ where M is the number of paths and N is the number of periods. Chan et al. (2003) provide a backward-path generation method to reduce the large amount of storage required in Tilley's model. Their solution is able to reduce the memory storage to $O(M)$ by generating the paths backwards and not storing all of the intermediate stock prices as is the case with Tilley's simulation model. The research contained within this thesis uses Chan et al.'s (2003) algorithm as a framework to price American options. Please refer to Chapter 3 for detailed information regarding Chan et al.'s algorithm.

In summary, Monte Carlo simulation works by generating M pricing paths of an underlying stock, calculates the gains of that path using the traditional valuing system as presented in the introduction (dependent on whether the option is a call or a put option), and then finds the expected option value discounted to the initial time period. This discounted present value is therefore the estimated price (i.e. the premium) associated with the option. Generally, numerical methods contain an expected value term within the equations. By generating M paths and finding the average option worth of these paths, Monte Carlo simulation is essentially creating this expected value through a stochastic sampling technique. Overall, Monte Carlo simulation has grown drastically in popularity due to its ease of implementation and modification as well as its accuracy.

2.2 Modeling of the Jump Diffusion Processes

As previously mentioned in Chapter 1, the idea of a market where the underlying stocks' prices have the possibility of jumps was introduced by Merton in 1976. Some researchers have proposed jump-diffusion models (Merton, 1976 and Kou, 2002) whereas other researchers have

used stochastic volatility models that are beyond the scope of this research. The stochastic volatility models are either too complicated to obtain practical algorithms that are easy to implement and provide comparable results or are too simplistic and cannot capture the important leptokurtic features of the markets (leptokurtic features will be explained further in a subsequent section). Jump-diffusion models on the other hand are easier to implement, better in capturing the market's phenomena, and are comparative to stochastic volatility models in terms of pricing accuracy (Zhu 2005).

In recent years a multitude of research has been conducted within jump-diffusion models. The use of the Poisson distribution for modeling the timing of the jumps is fairly consistent among the researchers due to the unique features of the market jumps and because the jumps are frequently memoryless and *rare* (*rare* is used loosely and means that a jump will not occur frequently within a small enough time horizon). However, the issue of modeling the size of the jumps has been continuously debated over the past three decades. Merton (1976) chose to use a log-normally distributed process, Kou (2004, 2008) chose a log-double exponentially distributed process, Hanson and Westman (2002) propose a log-uniform process, and yet another research group in Zhu and Hanson (2005) propose a log-double-uniformly distributed model. In each case, there are advantages and disadvantages of the underlying distribution used for modeling jump-diffusion processes under different circumstances.

Every jump-diffusion model, despite which underlying distribution is chosen, has two distinct disadvantages: the amount of time that can be modeled (the planning horizon) and increased calculation times due to the additional processes being added to the Black-Scholes model. Jump-diffusion models are not good at modeling long term behaviors of the financial markets due to the unpredictable nature of the markets in the long run. Over a short amount of time, a researcher/practitioner can predict the likelihood of a jump. However, this ability to predict jumps decreases as the planning horizon increases. Therefore, jump-diffusion models have a short-term domain and are much more accurate, and valid, within this domain. The second disadvantage is the amount of time required to model the jump-diffusion process. As the next two subsections will explain, jump-diffusion models require additional mathematical terms be added to the Black and Scholes model. The result of these additional terms is increased computational time and effort.

The research contained within this thesis focuses on Merton's and Kou's models due to their popular acceptances among researchers and their relatively close modeling to the practical jump-diffusion behaviors visible within the markets. The following two sections will explain the jump-diffusion models presented by these two researchers.

2.2.1 Merton's Model

As Merton (1976) presented, the change in the stock price can be attributed to two key factors: the “normal” variations in the price caused by supply and demand and the “abnormal” variations in the price caused by the arrival of important new information or events that directly affect the markets. The “normal” variations explain the day-to-day changes in the underlying stock price and the “abnormal” variations explain the large jumps visible in the markets. In order to model these “abnormal” jumps in the stock prices, he modified the Black and Scholes model to include a term that would account for the jumps, as shown in the equations below. Equation (2.7) is the original Black and Scholes model and equation (2.8) is the model Merton formulated where $J(t)$ is a Poisson process with mean λ and Y_i is a sequence of independent identically distributed random numbers that follows a standard normal distribution.

The added term constitutes the jump-diffusion of the market. The Poisson process, $J(t)$, is a counting process for the number of jumps in a year and Y_i is the size of the price jump. As in the original Black and Scholes model, μ is the drift parameter, σ is the stock volatility, and $W(t)$ follows a standard Brownian motion.

$$\frac{dS(t)}{S(t)} = \mu dt + \sigma dW(t) \quad (2.7)$$

$$\frac{dS(t)}{S(t)} = \mu dt + \sigma dW(t) + d\left(\sum_{i=1}^{J(t)} Y_i - 1\right) \quad (2.8)$$

In Merton's model, Y_i follows a normal distribution and has a normal density of

$$f_Y(y) \sim \frac{1}{\sigma' \sqrt{2\pi}} \exp\left\{-\frac{(y - \mu')^2}{2\sigma'^2}\right\}, \quad (2.9)$$

where μ' and σ' is the mean and standard deviation of Y (Kou 2008).

The stochastic differential equations (SDE) in equation (2.7) and equation (2.8) can be solved to formulate an equation to calculate the stock price at a given time t . Equations (2.10)

and (2.11) show the solutions to these two SDEs where S_0 is the initial stock price and $S(t)$ is the price of the stock in time period t .

$$S(t) = S_0 \exp([\mu - \frac{1}{2}\sigma^2]t + \sigma W(t)) \quad (2.10)$$

$$S(t) = S_0 \exp([\mu - \frac{1}{2}\sigma^2]t + \sigma W(t)) * \prod_{i=1}^{J(t)} \exp(Y_i) \quad (2.11)$$

Merton's model is beneficial in that it provides a means for explaining a portion of the extreme jumps visible in the stock market that the Black and Scholes model does not explain. However, one of the key disadvantages of this model can be seen when analyzing the jump size term, Y_i . According to Merton (1976) the jump size, Y_i , follows a standard normal distribution which is symmetric and bell-shaped; however, Kou (2008) showed that this distribution does not always realistically represent the jump sizes seen in the stock markets. As explained in the subsequent section, Kou shows that Y_i is better explained by the double exponential distribution.

Since Merton proposed the idea of jumps within the markets, various researchers have studied different methods of pricing options with a jump-diffusion process. Some of these researchers include Amin (1993) who extended the binomial method to handle jumps, Zhang (1997) who developed extensions for the PDE finite difference method using variational inequalities to handle jump-diffusion models, Laprise et al (2006) provides an approximation that prices an American option based on the price of a European call option, and Feng and Linetsky (2008) who proposed a new high-order time discretization scheme for the partial integrodifferential equation (PIDE) based on the extrapolation approach to the solution of ordinary differential equations (ODEs). These are just a few examples of research that extends on Merton's model to price various types of options.

2.2.2 Kou's Model

The underlying model proposed by Kou (2002) is identical in notation to Merton's model. However, Kou (2002) makes one major modification to the pricing formula: instead of Y_i following a standard normal distribution, Kou believes that Y_i actually follows a double exponential distribution. Under this model, the asset price $S(t)$ is given by

$$\frac{dS(t)}{S(t)} = \mu dt + \sigma dW(t) + d\left(\sum_{i=1}^{J(t)} V_i - 1\right), \quad (2.12)$$

where $\{V_i\}$ is a sequence of independently identically distributed nonnegative random variables such that $Y=\log(V)$ has an asymmetric double exponential distribution with density of

$$f_Y(y) \sim p\eta_1 \exp(-\eta_1 y)1_{\{y \geq 0\}} + q\eta_2 \exp(\eta_2 y)1_{\{y < 0\}} \text{ and} \quad (2.13)$$

$$E[V] = E[e^Y] = p \frac{\eta_1}{\eta_1 - 1} + q \frac{\eta_2}{\eta_2 + 1}. \quad (2.14)$$

Here, p and q represent the probability of positive and negative jumps, respectively, $p \geq 0$, $q \geq 0$, and $p + q = 1$. The parameters $1/\eta_1$ and $1/\eta_2$ are the means of the two exponential distributions and as in Merton's model, all random numbers, $J(t)$, $W(t)$, and Y_i are assumed to be independent. Additionally, it is assumed that $\eta_1 > 1$ to ensure that $E[V] > \infty$ and $\eta_2 > 0$.

Kou (2008) explained his rationale behind the distribution change in his paper published in 2008. By looking at a histogram of the normalized daily returns of the S&P 500 index from January 2, 1980 to December 31, 2005, it is evident that the histogram contains a high peak near the mean value and two heavy tails when compared to the normal distribution. These two features combined are known as the leptokurtic feature, which means that the kurtosis of the distribution is very large (Kou 2008). The double exponential distribution (and others such as the Laplace distribution and Logistic distribution) is a member of the leptokurtic distribution family and produces the leptokurtic features. In a financial sense, the kurtosis is reviewed when looking at the historical returns of a stock and the variance of the dataset. Leptokurtic distributions usually have low variance because the returns are usually close to the mean. Many investors will structure their portfolios to produce the leptokurtic feature in order to reduce the possibility of having large, irregular swings in stock prices.

Kou (2008) calculated the estimated kurtosis and skewness of the sample data from the S&P 500 and found that the kurtosis (K) is approximately 42.23 and the skewness (S) of the distribution is -1.73 . Any value for K larger than 3 (a value of 3 indicates a normal distribution) is considered a leptokurtic distribution. This dataset, and many other index price histories, is obviously a leptokurtic distribution with $K = 42.23$. The skewness indicates that the dataset has asymmetric tails since it is not equal to zero. A normal distribution has a skewness equal to zero. The negative skewness signifies that the S&P 500 dataset has a heavier left tail than right.

The fact that the skewness is not zero and the kurtosis is larger than 3 shows two things: (a) the returns do not follow a normal distribution as Merton suggested, and (b) the benefit of

using the double exponential distribution since this distribution can model the asymmetric attributes of the data. Because of these facts, Kou believes the double exponential distribution is suitable for modeling the returns, and therefore the prices, of options. This is due to the double exponential distribution being a leptokurtic distribution that can be configured to contain skewness since it is made of two different exponential distributions with the ability to have different means.

The ability to have different means allows the positive and negative jump sizes to be asymmetric. Thinking about this in a non-mathematical sense when an announcement is made on Wall Street, investors' perspectives can be split into two groups: (a) those who feel that the announcement is good, causing the stock prices to jump upward, and (b) those who feel that the announcement is bad, causing the stock prices to jump downward. However, the number of investors in each group will not always be equally divided and therefore a symmetrical distribution may not be an ideal description of the resulting market jumps. Under such circumstances, the use of the double exponential distribution may be a better choice to model the jump sizes. If more investors feel that the announcement of new market information is good, the double exponential distribution can model this (with the parameters of $\eta_1=25$ and $\eta_2=50$ for example) by making the size of the upward jumps larger than that of the downward jumps. If the majority thinks the announcement is bad, then the parameters can be switched, making the downward jump sizes larger. Using a normal distribution, as in the case of Merton's model, assumes that the two aforementioned groups are equal in size and that the announcement has the same effect on those feeling the market information was good and those that feel the announcement was bad. If this fact is true, the double exponential distribution can model the symmetry; however, if it is not true the double exponential distribution can also model the asymmetrical features.

However, the main problems associated with the double exponential distribution lie in the amount of time required and the number of mathematical functions that must be evaluated to obtain a solution as well as the fact that the double exponential distribution contains exponentially small tails and does not have bounded jump amplitudes, which according to Zhu, does not accurately model real market data (Zhu 2005). Contrary to this thought, the fit of the tails is extremely dependent on the data that is being fitted. So in one instance, the double exponential distribution will not be the best fit distribution and in other cases it will.

Within Kou's (2004, 2008) research, he provides methods for pricing four different types of options with the double exponential jump-diffusion model. In this paper, he provides formulae and approximation algorithms to price finite-time horizon American options, loopback options, barrier options, and finally perpetual American options. For the finite-time horizon American options, Kou (2004, 2008) formulated two different approximation algorithms to price these options. The following subsections present his approximation algorithms.

2.2.2.1 Quadratic Approximation Algorithm

The first of Kou's (2004, 2008) approximation algorithms is the quadratic approximation algorithm. This approximation algorithm uses the price of a European option with two additional terms to approximate the price of an American option. In particular, the algorithm is as follows:

Quadratic Approximation:

The price of an American put option with maturity t and strike price K can be approximated by the following function:

$$\psi(v, t) = \begin{cases} \text{EuP}(v, t) + Av^{-\beta_3} + Bv^{-\beta_4}, & \text{if } v \geq v_0 \\ K - v & \text{if } v \leq v_0 \end{cases} \quad (2.15)$$

where $v_0 \in (0, K)$ is the unique root of equation

$$C_\beta K - D_\beta [v_0 + \text{EuP}(v_0, t)] = (C_\beta - D_\beta)Ke^{-rt} \bullet P^{v_0}[S(t) \leq K]$$

where $P^{v_0}[S(t) \leq K]$ is the probability that the stock price at t is below the strike price, K , with an initial stock price of v ,

$$\begin{aligned} C_\beta &= \beta_3 \beta_4 (1 + \eta_2), \\ D_\beta &= \eta_2 (1 + \beta_3)(1 + \beta_4) \end{aligned}$$

and the two constants of equation (2.15) are calculated by

$$A = \frac{v_0^{\beta_3}}{\beta_4 - \beta_3} \{ \beta_4 K - (1 + \beta_4)[v_0 + \text{EuP}(v_0, t)] + Ke^{-rt} P^{v_0}[S(t) \leq K] \} > 0 \quad (2.16)$$

$$B = \frac{v_0^{\beta_4}}{\beta_3 - \beta_4} \{ \beta_3 K - (1 + \beta_3)[v_0 + \text{EuP}(v_0, t)] + Ke^{-rt} P^{v_0}[S(t) \leq K] \} > 0. \quad (2.17)$$

Please refer to Appendix A for the mathematical derivation of $\beta_1, \beta_2, \beta_3, \beta_4$. These betas are the four roots of the equation $G(x) = \alpha$ where $G(x)$ is part of the moment generating function. Namely,

$$G(x) := x \left(r - \delta - \frac{1}{2} \sigma^2 - \lambda \zeta \right) + \frac{1}{2} x^2 \sigma^2 + \lambda \left(\frac{p \eta_1}{\eta_1 - x} + \frac{(1-p) \eta_2}{\eta_2 + x} - 1 \right) \quad (2.18)$$

where ζ is equal to equation (2.14) and δ is equal to the dividend rate.

2.2.2.2 Piecewise Exponential Approximation

The second of his approximation algorithms is the called the piecewise exponential approximation algorithm, extending the research performed by Ju (1998), Carr et al. (1992), Gukhal (2001) and Pham (1997). This approximation algorithm also uses the price of a European put option and a few additional terms to approximate the price of an American put option. This approximation algorithm is given by:

$$\begin{aligned} P_A(S_t, t, T) = & P_E(S_t, t, T) + \int_t^T e^{-r(s-t)} rKE * [1_{\{S_s \leq S_s^*\}} | S_t] ds - \\ & \delta \int_t^T e^{-r(s-t)} rKE * [S_s * 1_{\{S_s \leq S_s^*\}} | S_t] ds - \\ & \lambda \int_t^T e^{-r(s-t)} rKE * [\{P_A(VS_{s-}, s, T) - (K - VS_{s-})\} 1_{\{S_{s-} \leq S_{s-}^*\}} 1_{\{VS_{s-} > S_{s-}^*\}} | S_t] ds \end{aligned} \quad (2.19)$$

where $P_E(S_t, t, T)$ is the price of the European put option, S_s^* is the early exercise boundary at time s . If the stock price falls below S_s^* at time s , then it is optimal to exercise immediately. The integrals represent the present value of the interest accrued on the strike price in the exercise region, the present value of the dividends lost in the exercise region, and finally the rebalancing costs due to the jumps from the early exercise region to the continuation region. Kou (2008) goes into further detail regarding the effects this term has on the overall option price. Since the research contained within this thesis focuses more on the first approximation algorithm presented, this additional detail is not presented.

Other researchers have utilized or extended on Kou's double exponential model for pricing options. A few examples include Quittard-Pinon and Randrianarivony (2007) and Feng and Linetsky (2008).

2.3 Copulas

One of the key aspects of analyzing a small group of stocks, known as a basket, is the correlation between the stocks in question. The use of copulas to model this correlation has grown in recent years. Mathematically, a copula is a function that allows the combination of several univariate distributions to obtain a joint distribution with a certain dependence structure from the correlation (Dorey et al.). Sklar's theorem, which is based on a two-dimensional copula but can be extended to an n -dimensional copula, is the foundation for copulas. Sklar's theorem is as follows (Dorey et al.):

Sklar's Theorem

Let F_{XY} be a joint distribution with marginals F_X and F_Y . Then there exists a function $C:[0,1]^2 \rightarrow [0,1]$ such that

$$F_{XY}(x,y) = C(F_X(x), F_Y(y)) \quad (2.20)$$

If X and Y are continuous, then C is unique; otherwise, C is uniquely determined on the (range of X)(range of Y). Conversely, if C is a copula and F_X and F_Y are distribution functions, then the function F_{XY} is a joint distribution with marginals F_X and F_Y .*

Sklar's Theorem can be extended to show that the n marginal distributions and the dependence structure can be separated and the copula function will completely describe the dependence between each of the n variables. Copulas are used as a way to link n univariate distributions to form a multivariate distribution.

In layman's terms, a copula works in the same manner as creating a random variate using the CDF to obtain a sample from the PDF of a distribution. For example, starting with a random value obtained from a uniform $U(0,1)$ distribution one can obtain a random normal variate by inverting the CDF of the normal distribution at that $U(0,1)$ value. The value of the CDF will then be the randomly generated number. This process will work for any distribution and is exemplified using a normal distribution for this example. Figure 2-4 shows how the process works. The horizontal line is the $U(0,1)$ number that is used to evaluate the CDF of the normal distribution. The random variate is then the number that corresponds to this point on the x-axis; in this case, the $U(0,1)$ is approximately 0.7 and the random variate that is generated is approximately 0.5. Please note that this is the theoretical process of generating a random variate. The computer implementation of this process can vary depending on the approximation algorithm that is chosen.

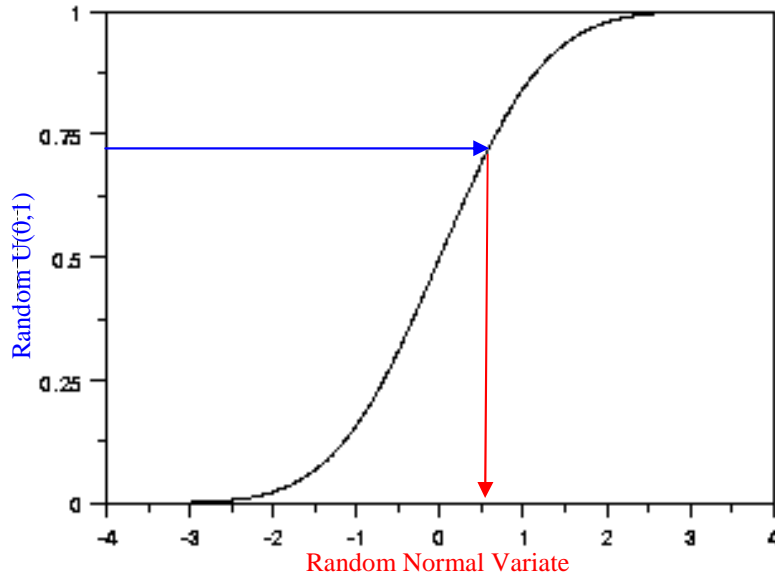


Figure 2-4: Random Variate Generation

Copulas operate in a similar manner, extending the procedure to include more than one distribution. However, they differ in that the CDF and PDF are probabilistically linked and are not necessarily linked in a straight-line manner. In the previous example the $U(0,1)$ and the random variate are linked in a linear manner. However, in the case of a copula – a multivariate normal distribution, for example – the two values are not linearly related due to the multidimensional aspect of the copula (Dorey et al.).

As a formal definition:

Definition

A copula is a function $C:[0,1]^2 \rightarrow [0,1]$ which satisfies:

- (a) For every u, v in $[0,1]$, $C(u,0) = C(0,v) = 0$ and $C(u,1) = u$ and $C(1,v) = v$;
- (b) For every u_1, u_2, v_1, v_2 in $[0,1]$ such that $u_1 \leq u_2$ and $v_1 \leq v_2$,
 $C(u_2, v_2) - C(u_2, v_1) - C(u_1, v_2) + C(u_1, v_1) \geq 0$.

Some of the most important copulas being utilized within financial modeling are as follows (Bluhm 2007):

Gaussian Copula

The n -variate Gaussian copula with linear correlation matrix Γ is

$$C(u_1, u_2, \dots, u_m) = N_m[N^{-1}[u_1], \dots, N^{-1}[u_m]; \Gamma]$$

where $N[\bullet]$ is the standard normal function, $N^{-1}[\bullet]$ is its inverse, and $N[\dots;\Gamma]$ is the multivariate Gaussian distribution function with correlation matrix $\Gamma = (\rho_{ij})_{1 \leq i, j \leq m}$. Positive definite matrices of Γ can be decomposed using Cholesky decomposition (explained further in section 6.1.1) to obtain an $n \times n$ matrix A where $\Gamma = AA^T$. Clearly A is a lower triangular matrix where the values represent the covariance matrix between the n variates. With this, define $\mathbf{Z} = [Z_1, \dots, Z_n]$ where each $Z_i \cong \mathcal{N}(0,1)$ and are independent, the Gaussian copula will be of the form

$$\mu + A\mathbf{Z} \sim \mathcal{N}(\mu, \Gamma).$$

Figure 2-5 below provides two examples of a Gaussian copula created from code written in C++. In each case $n = 2$, meaning there are two univariate distributions in each chart. The chart on the left is a Gaussian copula with Gaussian marginals, and the plot on the right is a Gaussian copula with Student-T marginals. In each case, the correlation between the two distributions is equal to 0.5 (which is why the data points are rotated approximately 45 degrees).

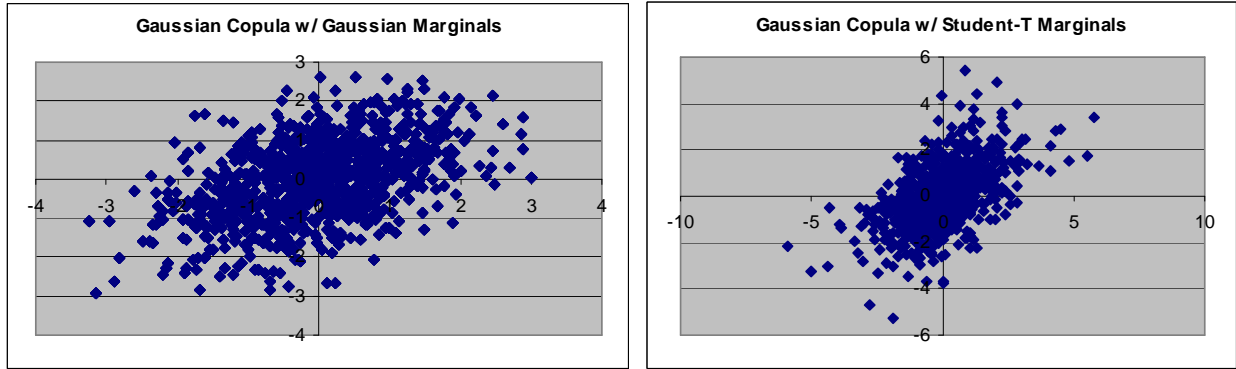


Figure 2-5: Gaussian Copulas

When looking at the plots above, imagine the distributions on both the x- and y-axis. Due to the shape of the Gaussian (normal) and Student-T distributions, the resulting copula is extremely dense near the origin and then fades out. In the case of the Student-T marginals, it is evident that the tails spread out to form a *bowtie* shaped copula. This spread is due to the fatter tails that are evident in the Student-T distribution as compared to the Gaussian distribution. These two examples show how the copula inherits the properties of the marginal distributions and combines them to produce a joint-distribution.

Student-T Copula

Another very popular type of copula is the Student-T copula. The n -variate Student-T copula with linear correlation matrix Γ is

$$C(u_1, u_2, \dots, u_m) = \Theta(\Theta_d^{-1}[u_1], \dots, \Theta_d^{-1}[u_m]; \Gamma)$$

where Θ is the multi-variate Student -T distribution function with d degrees of freedom and Θ^{-1} is the inverse of the function and Γ is the correlation matrix as before. Again, the correlation matrix can be decomposed using Cholesky decomposition to obtain a lower triangular covariance matrix, A . A Student-T variate with mean μ can be represented as

$$\mathbf{X} = \mu + \frac{\sqrt{v}}{\sqrt{S}} A \mathbf{Z}$$

where $S \sim \chi^2$ and $\mathbf{Z} = [Z_1, \dots, Z_n]$ where each $Z_i \equiv \mathcal{N}(0,1)$ and are independent (independence is also assumed between S and \mathbf{Z}). Student-T copulas are beneficial in that they are very similar to the Gaussian copula but provide thicker tails than that of the Gaussian.

The following figure provides two examples of the Student-T copula. Again, each chart contains two univariate distributions in each chart. The chart on the left contains Gaussian marginals and the chart on the right contains Student-T marginals. Again, the marginals have a correlation of 0.5.

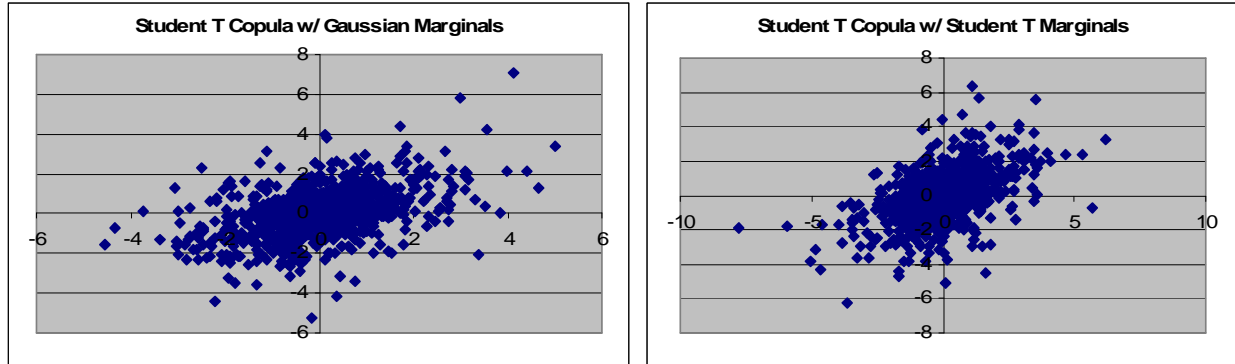


Figure 2-6: Student-T copula

Each of these copulas can be used with different marginals. For example, the most common copulas are the following: Gaussian copula with Gaussian marginals, Gaussian copula with Student-T marginals, Student-T copula with Student-T marginals, and finally the Student-T copula with Gaussian marginals, all of which were shown in Figures 2.5 and 2.6. Copula functions have some explicit advantages that make them a preferable choice for multivariate simulation. Namely, as Srinivas et. al (2006) explains, copulas are beneficial in the following ways:

1. Copulas provide a means to simulate multivariate distributions from dependent random univariates.
2. Copulas provide a means of separating marginals from the dependence structure. This reduces the study of multivariate distributions to a study of multivariate dependence structure.
3. Copulas remove the problems associated with linear correlation coefficients in conventional simulations.

The research contained within this thesis will utilize copulas to price baskets where the options' underlying assets undergo a jump-diffusion process.

CHAPTER 3 - SIMULATING AMERICAN OPTIONS WITH JUMP DIFFUSION

This chapter introduces the main objective of this thesis: pricing American options with Monte Carlo simulation. This chapter begins by presenting background information regarding the use of least-squares regression to price American options. The subsequent sections extend on this background knowledge to include the modeling of the jump-diffusion process and explain the relationship each input parameter (independent variable) has with the simulated stock prices.

3.1 Background on Least-Squares Regression

The use of the least-squares regression to estimate the expected value of continuation for an American option was first introduced by Longstaff and Schwartz (2001). Their proposed method uses backward analysis to determine whether or not the option would be exercised in the given period by comparing the immediate profit upon exercising and the expected profit of continuing to hold the option. The method begins by simulating numerous sample paths from the initial time period until the final period. At this point, each of the sample paths are analyzed to determine if the option should be exercised, noting the expected value of each path if it is continued to be held, and proceeding on to analyze the previous time period to determine if the path is in-the-money. In each instance that a path could be exercised – or in other words, for each path that is in-the-money – a quadratic regression, shown in equation (3.1), is performed that relates the continuation value with the current value of the option if exercised immediately. The method

$$Continuing = \beta_0 + \beta_1 * exercise + \beta_2 * exercise^2. \quad (3.1)$$

continues by noting the maximum value for each path, determining whether that value is from exercising immediately or continuing and proceeds to the next time period, repeating the process until the initial time period is reached. Once all periods are analyzed, the maximum value for each path is analyzed to determine the average value of the option, indicating the expected worth (or fair market price) of that option. The method proposed by Longstaff and Schwartz (2001) is

a very efficient algorithm in that the regression is only performed when the options are in-the-money. This prevents any unnecessary calculations and ultimately improves the running times.

Chan, et al. (2003) proposed a Monte Carlo simulation method for pricing American options. The algorithm they proposed is very similar to that of Longstaff and Schwartz (2001); however, instead of forwardly generating each of the paths and then working backwards, Chan et al generates the paths backwards. Namely, equation (2.10) is modified and is defined as:

$$\begin{aligned}
S_1 &= S_0 \exp([r - \frac{1}{2}\sigma^2]\Delta t + \sigma\sqrt{\Delta t}\varepsilon_N) \\
&\vdots \\
S_i &= S_0 \exp(i[r - \frac{1}{2}\sigma^2]\Delta t + \sigma\sqrt{\Delta t}(\varepsilon_N + \varepsilon_{N-1} + \dots + \varepsilon_{N-i+1})). \\
&\vdots \\
S_N &= S_0 \exp(N[r - \frac{1}{2}\sigma^2]\Delta t + \sigma\sqrt{\Delta t}(\varepsilon_N + \varepsilon_{N-1} + \dots + \varepsilon_1))
\end{aligned} \tag{3.2}$$

where $\varepsilon \sim \mathcal{N}(0, 1)$.

A major benefit of this algorithm is that each of the random numbers required during the simulation can be obtained from the initial starting seed value. This allows the random number set to be regenerated from the initial seed value as the algorithm progresses backwards. In this case each random number is generated twice, but significantly reduces the amount of storage required to perform the simulation since each random number (and resulting stock prices) need not be stored. The simulations used within this thesis utilize Chan, et al's algorithm. The following section provides the notation of the algorithm and the algorithm itself.

3.1.1 Notation & Algorithm

The following notation, in addition to a few variables to be introduced in a subsequent section, will be used for the remaining portions of the thesis:

- $S_0 \equiv$ Initial stock price
- $r \equiv$ Risk-free interest rate
- $K \equiv$ Strike price
- $\sigma \equiv$ Stock volatility
- $N \equiv$ Number of time periods
- $\Delta t \equiv$ Length of each time period (in years)
- $M \equiv$ Number of paths

$T \equiv$ Expiration time

$Z_i \equiv$ Independently identically distributed from $\mathcal{N}(0, 1)$ for $i = 1, 2, \dots, N$

$$\omega_i = Z_N + Z_{N-1} + \dots + Z_{N-i+1}$$

The algorithm given in Chan et al. (2003) is now explained in a step-by-step procedure as explained by DeHaven (2007). The research contained within this thesis utilizes the least-squares approach of Longstaff and Schwartz (2001) with the addition of backwardly generating paths of Chan et al (2003). DeHaven (2007) presents an algorithm as follows:

Step 0. System Inputs:

- (a) Initial stock price (S_0)
- (b) Risk-free interest rate (r)
- (c) Strike price (K)
- (d) Stock volatility (σ)
- (e) Number of time periods (N)
- (f) Number of paths (M)
- (g) Length of time horizon in years (T)
- (h) Call or put option

Step 1. Initialization:

- (a) Set $\Delta t = T/N$
- (b) Set the seed for the initial path to any positive integer.
- (c) Generate the random variate $Z_j \sim \mathcal{N}(0,1)$ for each path $j = 1, 2, \dots, M$
and compute their sum ω_N .

Step 2. Compute S_N for the expiration date T using:

$$S_i = S_0 \exp(i[r - \frac{1}{2}\sigma^2]\Delta t + \sigma\sqrt{\Delta t}\omega_i). \quad (3.3)$$

Step 3. Compute the cash flows for each path using one of the following:

$$P(j) = \begin{cases} \max\{K - S_j(t), 0\} & \text{put option} \\ \max\{S_j(t) - K, 0\} & \text{call option} \end{cases} \quad (3.4)$$

Step 4. Backup one time period; set $i = i - 1$.

(a) Using the same seed sequence, extract Z_{N-i+1} and compute

$$\omega_{i-1} = \omega_i - Z_{N-i+1}.$$

(b) Compute S_{i-1} by using equation (3.3).

(c) Extract the next seed value.

Step 5. Compute if the option is in the money for each path j . For each path:

(a) Let X be the vector containing asset prices S_i and Y be the vector containing the corresponding cash flows received at $i+1$ time period, which have been discounted backward to the i^{th} time period.

(b) Regress using least-squares approach to estimate the value of continuing using equation (3.1). This will result in the conditional expectation function $E[Y|X]$.

(c) Compute the value of continuing using $E[Y|X]$ and the value of immediately exercising using equation (3.4).

(d) Determine whether to exercise the option immediately or hold the option until the next time period, based on which gives the higher value. Establish the current cash flows conditional on not exercising prior to time period i using:

$$C_i(j) = \begin{cases} \text{cash flow} & \text{if cash flow} \geq E[Y|X] \\ 0 & \text{otherwise} \end{cases} \quad (3.5)$$

(e) Compute the present value of the cash flows $P_i(j)$ given by:

$$P_i(j) = C_i(j) + e^{-r\Delta t} P_i(j). \quad (3.6)$$

Step 6. If at time period zero stop, else go back to Step 4.

DeHaven (2007) provides four examples illustrating the correctness of the algorithm. In each instance the spread in the graphs expand as time progresses, confirming the fact that it

accurately approximates Brownian motion and the Black and Scholes model. Please refer to her thesis for the details of the examples and the graphs that correspond to her examples.

3.2 A Jump-Diffusion Simulation Model

Building on the least-squares Monte Carlo simulation, the next step is to include the jump-diffusion process into the algorithm. As previously mentioned in Chapter 2, when a stock follows a jump-diffusion process, the stock price has the potential to increase or decrease by an unusually large amount. Figure 3-1 shows a simulated stock price that contains two major jumps within the simulated time horizon.

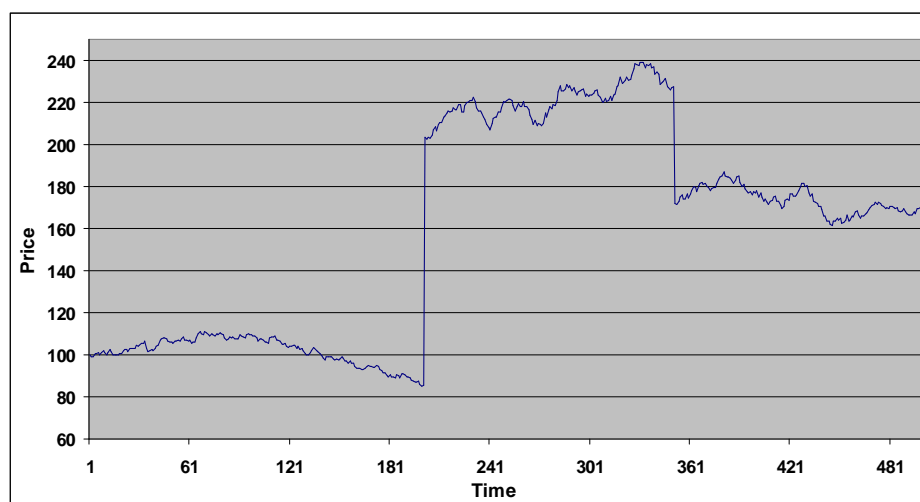


Figure 3-1: Simulated Path with Jumps

The jump frequency of a simulated path follows a Poisson distribution with a mean of λ . As one might expect, not all simulated paths contain jumps within the time horizon and each jump varies in magnitude. Figure 3-2 illustrates an example setting of 10 simulated paths in which only a few paths contain major or minor jumps in the stock price; the two major jumps within the paths (one upward and one downward) are labeled. As proposed in Kou (2002), the jump sizes follow a double exponential distribution with means of $1/\eta_1$ and $1/\eta_2$. Therefore, the likelihood and jump sizes are dependent on the parameters λ and η_1, η_2 , respectively. Larger values of λ result in more frequent jumps and larger values of η_1 or η_2 result in jumps of smaller magnitude.

The C++ code that is created to perform the Monte Carlo simulations is explained in the subsequent sections. The following sections detail the addition of jump-diffusion to the

algorithm presented in Chapter 3.1, as well as provide the results of the Monte Carlo simulation. This section begins by providing an overview of the additional parameters and variables used within the new C++ code.

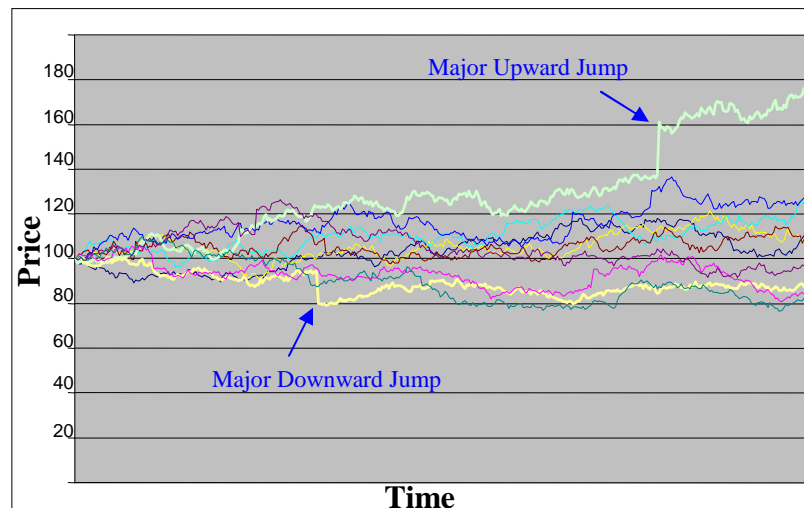


Figure 3-2: Simulated Paths 2

3.2.1 Notation

The notation used within the jump-diffusion model is identical to that of the least-squares regression algorithm presented in section 3.1. However, a few additional variables are added to this algorithm to incorporate the jump-diffusion into the pricing algorithm. The following variables are required for this algorithm:

$J_j(t) \equiv$ independent Poisson process with rate λ

$\eta_1 \equiv$ parameter of the double exponential distribution for the size of an upward jump

$\eta_2 \equiv$ parameter of the double exponential distribution for size of a downward jump

$V_i \equiv$ Independently identically distributed from double exponential distribution with means of $1/\eta_1$ and $1/\eta_2$

$p \equiv$ The probability of an upward jump

$1 - p \equiv$ The probability of a downward jump

3.2.2 Updated Algorithm

The updated algorithm is also very similar to that explained in the previous section. The following is a step-by-step process to price American options where the underlying stock undergoes a jump-diffusion process.

Step 0. System Inputs:

- (a) Initial stock price (S_0)
- (b) Risk-free interest rate (r)
- (c) Strike price (K)
- (d) Stock volatility (σ)
- (e) Number of time periods (N)
- (f) Number of paths (M)
- (g) Length of time horizon in years (T)
- (h) Call or put option
- (i) Exponential Distribution parameters (η_1 and η_2)
- (j) Probability of Upward Jump (p)
- (k) Rate of Poisson Process (λ)

Step 1. Initialization:

- (a) Set $\Delta t = T/N$
- (b) Set the seed for the initial path to any positive integer.
- (c) Generate the random numbers Z_j for each path $j = 1, 2, \dots, M$
- (d) Generate the random numbers $J_j(t)$ for each path $j = 1, 2, \dots, M$ and calculate the Jump Size for each period i using the equation:

$$V_i = \prod_{n=1}^{J(t)} V_n, \text{ where } V \text{ follows the double exponential distribution}$$

Step 2. Compute S_N for the expiration date T using:

$$S_i = S_{i-1} \exp([r - \frac{1}{2}\sigma^2]\Delta t + \sigma\sqrt{\Delta t}Z_i) * V_i. \quad (3.7)$$

*Note: If a jump is not present in period i , $V_i = 1$.

Step 3. Compute the cash flows for each path using one of the following:

$$P(j) = \begin{cases} \max\{K - S_j(t), 0\} & \text{put option} \\ \max\{S_j(t) - K, 0\} & \text{call option} \end{cases} \quad (3.8)$$

Step 4. Backup one time period; set $i = i - 1$.

- (a) Using the same seed sequence, extract Z_{N-i+1} .
- (b) Using the same seed sequence, extract $J(t)_{N-i+1}$ and compute its corresponding jump size V_{i-1} .
- (c) Compute S_{i-1} using:
$$S_{i-1} = S_i / \exp([r - \frac{1}{2}\sigma^2]\Delta t + \sigma\sqrt{\Delta t}Z_{i-1}) / V_{i-1}.$$
- (d) Extract the next seed value.

Step 5. For each path j , if the option is in-the-money, compute:

- (a) Let X be the vector containing asset prices S_i and Y be the vector containing the corresponding cash flows received at $i+1$ time period, which have been discounted backward to the i^{th} time period.
- (b) Regress using the least-squares method to estimate the value of continuing using equation (3.1). This results in the conditional expectation function $E[Y|X]$.
- (c) Compute the value of continuing using $E[Y|X]$ and the value of immediately exercising using equation (3.8).
- (d) Determine whether to exercise the option immediately or hold the option until the next time period, based on which gives the higher value. Establish the current cash flows conditional on not exercising prior to time period i using:

$$C_i(j) = \begin{cases} \text{cash flow} & \text{if cash flow} \geq E[Y|X] \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

- (e) Compute the present value of the cash flows $P_i(j)$ given by:

$$P_i(j) = C_i(j) + e^{-r\Delta t} P_i(j). \quad (3.10)$$

Step 6. If at time period zero stop, else go back to Step 4.

This algorithm uses Chan's (2003) memory reduction technique of not storing the intermediate prices. This method is extremely beneficial in that it requires far less memory than other algorithms that store each of the intermediate stock prices for each of the paths in the simulation. The only disadvantage of this process is the increase in computational requirements since each seed value must be calculated twice. However, as the number of paths or the number of periods being analyzed grow, the savings in memory outweigh the costs in computational power, especially since computers are continually advancing in their computational capabilities.

3.2.3 Characteristics of Simulation Parameters

Each of the input parameters associated with the least-squares model has its own effect on the pricing of options. This section gives a brief description of each of the parameters and shows how each parameter affects, if at all, the movements of the simulated stock prices.

Table 1: Parameter Descriptions

Parameter	Symbol	Description
Number of Periods	N	Number of segments the planning horizon is split into
Period Length	Δt	Amount of time (in years) for each of the N periods
Strike Price	K	Price in which the underlying stock is bought/sold at the time of exercise
Stock Volatility	σ	Natural variation of the underlying stock
Jump Frequency	λ	The average number of jumps in a given year
Jump Size	η	$1/\eta$ is the average jump size when a jump occurs

The following eight graphs demonstrate the effects of the input parameters. Each of the parameters is graphed two times, once with a high value and once with a low value. In each graph, paths are simulated for a planning horizon of either one half of a year (Figures 3-3 and 3-4) or one quarter of a year (remaining four graphs). The difference between the first two parameters, N and Δt , can be seen by comparing Figures 3-3 and 3-5. As the number of periods N increases, the number of possible exercise points evaluated increases. As a result, the exercise

boundary that is required for evaluating American options becomes increasingly more complete resulting in better accuracy of the estimated option value. Just like m -nomial trees (which was previously discussed in Chapter 2), Monte Carlo simulation uses discrete time periods to evaluate the worth of the option. The value N is the number of discrete time periods that the planning horizon is split into in which the option is evaluated. Therefore, the more times the option is evaluated the more accurate the pricing simulation should be. The second parameter, Δt , is directly related to the value of N . Namely, if T is the planning horizon (in years) then $\Delta t = T/N$. Intuitively, this means that the more periods in which the planning horizon is split into, the less amount of time within each period.

The third parameter (the option strike price) is important for the pricing of options; however, the value of K does not influence the stock price movements. The strike price merely determines how much profit is made given a certain stock price. Figures 3-3 and 3-4 show the effect σ has on the movements of the stock price. Naturally as time passes the volatility of the stock prices increase, which is why the spread of the prices increases as the periods, or time in other words, progress. However, the amount of spread is dependent on the magnitude of σ since the stock prices spread in the order of $\sigma\sqrt{\Delta t}$. Namely, stocks with larger volatilities create wider spreads than stocks with smaller volatilities, which is quite evident by comparing Figures 3-3 and 3-4.

The next set of graphs exemplifies the result of having larger values of λ , the average number of jumps in a year. As λ increases, the number of jumps becomes more frequent resulting in slightly more variation in the stock prices. Notice the difference between periods 1 and 3 in Figures 3-5 and 3-6. The increase in λ caused more jumps which in turn increased the volatility between these two periods. In other words, the increased volatility is caused by the additional number of outliers in the graphs. The increased number of jumps causes more outliers which in turn causes higher levels of volatility.

The last set of graphs, Figures 3-7 and 3-8, demonstrate how the average jump size affects the stock prices. Notice that since common random numbers were used to produce these graphs, jumps occur for the same paths at the same period for each of the two graphs. The difference between the two graphs is that Figure 3-8 contains much larger jump sizes (on average 250% larger) than that of Figure 3-7. The larger jumps cause the outliers to spread out by larger

amounts. This fact can be seen by looking at the amount of paths that are outside the dense, inner section of Figures 3-7 and 3-8. Since the jump sizes are larger in Figure 3-8, this corresponds to a wider spread of the paths and therefore a few outlying paths.

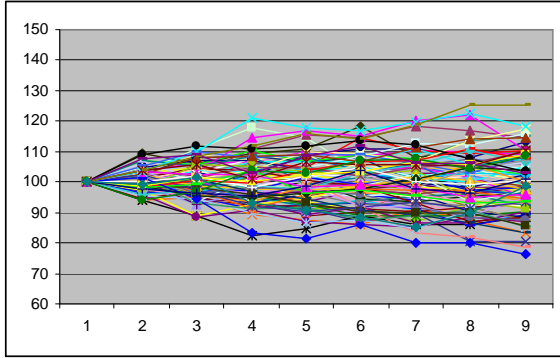


Figure 3-3: Input 1
 $(S_0 = 100, \sigma = 0.2, \lambda = 0, \eta_1 = 50, \eta_2 = 50)$

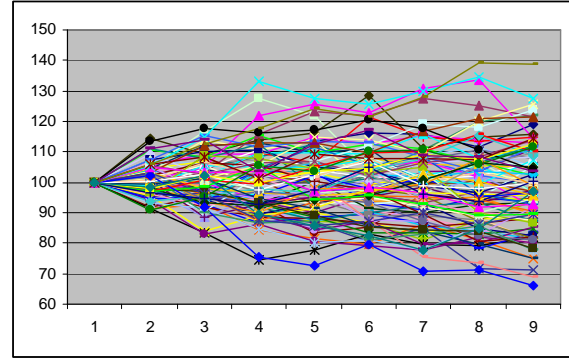


Figure 3-4: Input 2
 $(S_0 = 100, \sigma = 0.3, \lambda = 0, \eta_1 = 50, \eta_2 = 50)$

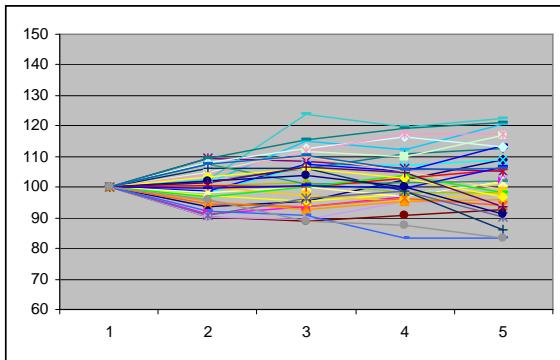


Figure 3-5: Input 3
 $(S_0 = 100, \sigma = 0.2, \lambda = 3, \eta_1 = 50, \eta_2 = 50)$

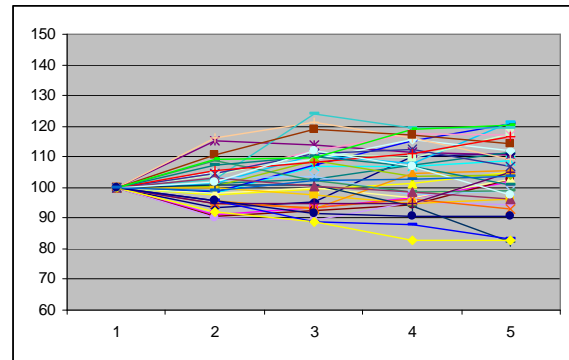


Figure 3-6: Input 4
 $(S_0 = 100, \sigma = 0.2, \lambda = 10, \eta_1 = 50, \eta_2 = 50)$

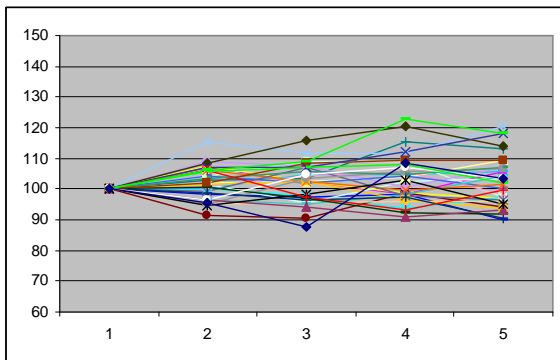


Figure 3-7: Input 5
 $(S_0 = 100, \sigma = 0.2, \lambda = 3, \eta_1 = 25, \eta_2 = 25)$

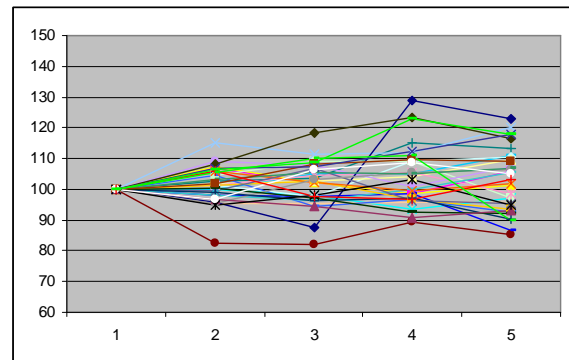


Figure 3-8: Input 6
 $(S_0 = 100, \sigma = 0.2, \lambda = 3, \eta_1 = 10, \eta_2 = 10)$

3.3 Chapter Conclusions

This chapter presented the least-squares algorithm that was used to simulate American options with jump-diffusion as well as the general characteristics of the simulation parameters. As previously highlighted, the Monte Carlo simulation for this portion of the thesis utilizes Chan's (2003) memory reduction technique. This chapter detailed the least-squares Monte Carlo simulation model proposed by DeHaven (2007), which included the memory reduction techniques presented by Chan (2003), and how this simulation model can be modified to incorporate the jump process into the pricing of American options. This chapter also presented how each of the input parameters has an impact on the general dynamics of the stock price movements. Figures 3-3 through 3-8 presented example stock paths that show the effects of the standard deviation (σ), the jump frequency (λ), and the average jump sizes (η_1 and η_2).

CHAPTER 4 - GENERATING SIMULATION DATA

The following sections tests the correctness of the random variate generators used within the simulation. In particular, this section presents the tests performed on each part of the simulation including the random variate generators for the uniform and normal distributions, as well as the copulas for the multi-option basket simulation. Chapter 6 extends the simulations presented in the previous chapters, however, the validation of the simulation inputs are summarized within this chapter. The rest of this chapter is divided into three sections; one section for each of the distributions and one section for copulas.

4.1 Uniform Random Number Generator

The uniform random number generator used within this thesis is called a linear congruential generator (LCG) that was first introduced by Lehmer (1951) and was programmed using the C++ coding language. The random numbers (Z_i) are defined by the recursive formula

$$Z_i = (aZ_{i-1} + c)(\text{mod } m) \quad (4.1)$$

where every variable except Z_i is a nonnegative integer (Law 2000). When selecting values for each of these variables, a few considerations must be made. In particular, the selection of values is very important to maximize the number of random numbers that can be generated before it cycles. This is a major problem with the linear congruential generators. The size of the period, which is the number of random variates created before cycling, is directly related by the value of m . Therefore, the value of m should be very large. According to Law (2000) the LCG reaches full period if the following are true:

1. The only positive integer that exactly divides both m and c is 1.
2. If q is a prime number (divisible by only itself and 1) that divides m , then q divides $a-1$.
3. If 4 divides m , then 4 divides $a-1$.

The values that are used throughout this research are $a = 100801$, $c = 103319$, and $m = 4294967295$. The three statements above are true for these values, indicating that the LCG reaches a full period before cycling.

To test the randomness of this generator (and to see if the values for a , c , and m were wisely chosen), ten thousand random variates were generated and tested using a variety of different tests. Namely, the random variates were tested using the chi-square test and four different tests that inspect the runs created (a run is merely another name for a subset of the random variates produced) by the LCG. The remaining parts of this section will explain each of these tests and provide the results of each.

Chi-Square Test

The chi-square test is used to check the uniformity of the random variates to see if they are uniformly distributed between 0 and 1. This process works by splitting the interval of $[0,1]$ into k subintervals where the size of k is determined by the number of samples (N) being tested; in these tests, $N = 10,000$. Once the value of k is determined, the next step is to create a histogram of the data to determine the number of random variates within each of the k intervals, which is assigned to f_j .

In order to calculate the chi-square statistic, the following equation is used:

$$\chi^2 = \frac{k}{N} \sum_{j=1}^k \left(f_j - \frac{N}{k} \right)^2 . \quad (4.2)$$

This will have an approximate chi-square distribution with $k-1$ degrees of freedom (d) under the null hypothesis that the random numbers are identically independently distributed. If the calculated value is greater than the value given in the chi-square table for the given degrees of freedom, then the null hypothesis is rejected. In a mathematical sense, if $\chi^2 > \chi_{k-1,1-\alpha}^2$ then the null hypothesis is rejected and the random variates are not identically independently distributed which indicates that the random number generator is not a sufficient generator. Otherwise, we fail to reject the null hypothesis that the random number generator produces identically independently distributed variates.

Runs Up and Down Test

The runs up and down test checks to see how many of the variates consecutively run up and down. In many cases, especially with ineffective generators, one sees major runs in the numbers where a large amount of consecutive values are increasing and then consecutively

decreasing. The number of runs up and down of a good random number generator should be normally distributed with a mean and variance of

$$\mu_a = \frac{2N-1}{3} \text{ and } \sigma_a^2 = \frac{16N-29}{90},$$

where a is the total number of runs (number of runs up and number of runs down). Since a should follow a normal distribution, the standard normal table and test statistics can be used to verify that a truly follows a normal distribution. The test statistic is calculated using the standard test statistic equation with the values of μ_a and σ_a above. Namely, the test statistic is calculated using the equation

$$Z_0 = \frac{a - \mu_a}{\sigma_a}. \quad (4.3)$$

Once Z_0 is calculated, the null hypothesis can be examined. In this case, the null hypothesis is that a is statistically normal. Failure to reject this hypothesis occurs when $-z_{\alpha/2} \leq Z_0 \leq z_{\alpha/2}$, where α is the level of significance. In effect, this test uses a two-sided test with an $\alpha/2$ level of significance in each tail. The critical values used in this research are -1.96 and 1.96 . If the test statistic is between these values, then it fails to reject the null hypothesis that the numbers are normally distributed.

Runs Above and Below the Mean Test

This test is also known as the sign test because it checks the sequence of numbers above (+) and below (−) the mean. This test is essential because the runs up and down test may not adequately assess the independence of the random numbers. The runs above and below the mean test utilizes the normal distribution test statistics to determine if the numbers are independent. In the equations below n_1 and n_2 are the number of runs either above or below the mean and b is the total number of runs in the sample size. Given these values, the mean and variance of b for a truly independent sequence is given by

$$\mu_b = \frac{2n_1n_2}{N} + \frac{1}{2} \text{ and } \sigma_b^2 = \frac{2n_1n_2(2n_1n_2 - N)}{N^2(N-1)}.$$

If either n_1 or n_2 is greater than 20, b is approximately normally distributed. The test statistic can be calculated using the equation

$$Z_0 = \frac{b - \left(\frac{2n_1n_2}{N}\right) - \frac{1}{2}}{\sqrt{\frac{2n_1n_2(2n_1n_2 - N)}{N^2(N-1)}}}. \quad (4.4)$$

Again the test statistic must be between $-z_{\alpha/2} \leq Z_0 \leq z_{\alpha/2}$, which again corresponds to -1.96 and 1.96 , in order to fail to reject the null hypothesis. Therefore, if Z_0 is within this range, then we fail to reject the null hypothesis that the LCG produces random variates that are statistically independent.

Run Length Test

The last type of test that is required is the run length test. This test is broken down into two separate types of tests. These tests look at the number of continuous sequences above and below the mean. For example, the test would fail if there are continuously two numbers above the mean and then two numbers below the mean because the numbers generated are obviously not independent. The test statistics for these tests follow the chi-squared distribution.

The first portion of this test searches for runs up and down and inspects the number of runs of a certain size, i , to see if it is truly random. Therefore, in the following equations, let Y_i be the number of runs of length i in the set of random numbers. The expected number of runs for a sample size of N is found using the equations:

$$E(Y_i) = \frac{2}{(i+3)!} [N(i^2 + 3i + 1) - (i^3 + 3i^2 - i - 4)] \text{ for } i \leq N - 2$$

and (4.5)

$$E(Y_i) = \frac{2}{(N)!} \text{ for } i = N - 1.$$

Using the equations above, the next step is to calculate the chi-squared statistic using the equation

$$\chi^2 = \sum_{i=1}^L \frac{[O_i - E(Y_i)]^2}{E(Y_i)}, \quad (4.6)$$

where $L = N-1$ and O_i is the observed number of runs of length i . In order to be statistically significant the calculated χ^2 must be less than the chi-squared critical value determined with $L-1$

degrees of freedom. If the calculated value is less than the critical value, then it fails to reject the null hypothesis that the LCG produces statistically significant independent variates.

The second portion of this test searches for runs above and below the mean. The same procedure is followed for this portion as in the first. A test statistic is calculated and compared to the chi-squared test statistic. The main differences between the portions lie in how the expected value is calculated and what the value of L is equal to. In this portion of the test, the expected value is determined by the equation

$$E(Y_i) = \frac{Nw_i}{E(I)} \text{ for } N > 20, \quad (4.7)$$

where w_i , the approximate probability that a run has a length of i , and $E(I)$, the approximated expected run length, is found by

$$w_i = \left(\frac{n_1}{N}\right)^i \left(\frac{n_2}{N}\right) + \left(\frac{n_1}{N}\right) \left(\frac{n_2}{N}\right)^i \text{ and } E(I) = \frac{n_1}{n_2} + \frac{n_2}{n_1}.$$

In the previous equation, n_1 and n_2 are the number of runs above and below the mean, as was the case in the runs above and below the mean test.

The next step to this portion is to determine the approximate expected total number of runs in the sequence of all lengths. This is determined by

$$E(A) = \frac{N}{E(I)}.$$

As in the previous portion of the test, the final step is to calculate the chi-squared test statistic using equation (4.6) and compare it to the critical value with $L-1$ degrees of freedom. However, in this case, $L = N$ instead of $L = N-1$ as used in the previous portion.

To summarize this section, Table 2 provides the results for each of the aforementioned tests. Since the LCG used within this research passes all of the required tests, it is safe to assume that the values chosen for a , c , and m produce an efficient uniform random number generator that creates random independently identically distributed variates.

Table 2: U(0,1) Test Results

<u>Test Name</u>	<u>Test Statistic</u>	<u>Critical Value</u>	<u>Required Parameters</u>	<u>Result</u>
Chi-Squared	27.26	30.144	$k=20, d=19, \alpha=0.05$	Pass
Up and Down	0.372	$-1.96 \leq Z \leq 1.96$		Pass
Above and Below the Mean	0.733	$-1.96 \leq Z \leq 1.96$		Pass
Run Length Portion 1	2.602	9.488	$L-1=5, \alpha=0.05$	Pass
Run Length Portion 2	3.801	16.91	$L=N=9, \alpha=0.05$	Pass

4.2 Normal Random Number Generator

Two different tests were performed to determine the correctness of the normal random variate generator being used within this research. For these tests, ten thousand random numbers were created using the random number generator created in C++. These numbers were then tested using the chi-square test that was explained in the previous section. The parameters used within this test were as follows: $k = 30$, $d = 29$, and $\alpha = 0.05$. With these parameters the calculated test statistic is 25.3 and the critical value is 42.557 with a p-value of 0.662. Since the p-value is larger than the value of α , this indicates that the chi-square test fails to reject the null hypothesis that the random numbers generated are identically independently distributed. To confirm these results, Rockwell Software's Input Analyzer was used to fit the data to the normal distribution. This software fit the data with a $\mathcal{N}(0.00447, 0.996)$ distribution – a output histogram is provided in Figure 4-1 below – and performed the chi-square test on the data. Rockwell's output indicated a test statistic of 25.3 which again corresponds to a p-value of 0.662 indicating that it fails to reject the null hypothesis that the random numbers generated are identically independently distributed.

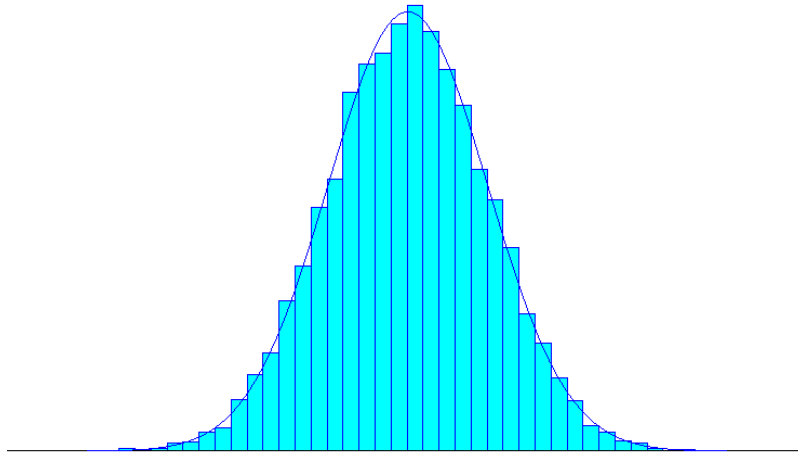


Figure 4-1: Fitted Histogram of $\mathcal{N}(0,1)$

The second test on the normal random number generator was performed with the statistical software, Minitab 15.1. Using the *Normality Test* function within Minitab, a probability plot was created which is shown in the figure below. This test also indicates that the data follows a normal distribution since the data closely resembles a straight line on the lognormal scale. This test also produces a p -value of 0.478, which again is larger than the associated α value of 0.05 indicating that it fails to reject the null hypothesis that the data follows a normal distribution.

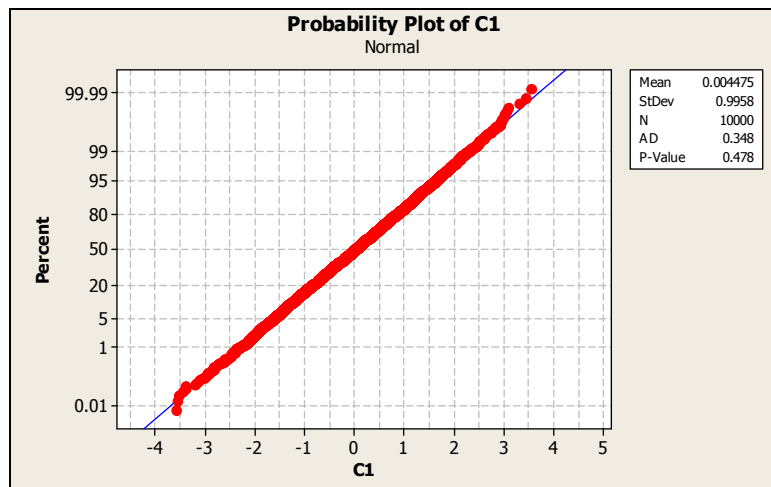


Figure 4-2: Normality Test

4.3 Copula Generator

The last main step that needs to be verified is our copula generator. To test this portion of the C++ program, 10,000 variates were simulated and plotted for each of the major copulas used within this research. Namely, the Gaussian and Student-T copulas were created using both Gaussian and Student-T marginals. The results of the copulas have been previously presented in Figures 2-5 and 2-6. In order to test the validity of the copula generator, the plots were compared to the published results presented in Bluhm (2007). Please refer to Bluhm's publication to observe the similarities between the plots he presents and the plots created through the copula generator used within this research.

CHAPTER 5 - COMPUTATIONAL RESULTS

The results of computational experiments for the proposed least-squares Monte Carlo simulation with the addition of jump-diffusion will be presented in this chapter. This chapter begins with a description of the experimental design that is used to test the effectiveness of using simulation to price American options. The subsequent section presents a comparison between the simulation model created for this thesis, the quadratic approximation algorithm presented in Kou (2004), as well as the binomial tree algorithm that was presented in Amin (1991) with 1,600 steps.

5.1 Design of Experiment

The purpose of this section is to explain how the experiments were conducted within this thesis. Section 5.1 will outline the design of experiment as well as explain the simulation parameters that were used within the experiments.

To test the accuracy and effectiveness of using Monte Carlo simulation, the least-squares algorithm presented in Chapter 3 was tested using various sets of input parameters. These simulations were then compared to the results of the quadratic approximation algorithm and Amin's (1991) binomial tree method presented in Kou (2004). To determine the relationship between the input parameters and the option price, a factorial design was created. Seven parameters were tested where σ , λ , η_1 , η_2 , and Δt have two levels, N has one level, and K has three levels. These levels are summarized in Table 3, below. A total of 96 different simulations were run with different combinations of the parameters. The two time horizons chosen for these experiments consisted of one quarter of a year and a full year. These two time horizons were chosen since they are the most commonly found life cycles in the financial markets for American options.

Each of the parameter combinations is tested using 30 replications of the least-squares Monte Carlo simulation model with 200 paths and a risk-free interest rate of 5%. The simulations were performed using common random numbers so that each can be compared in similar terms. This allows the simulation runs to be compared so that the relationship between each parameter and the resulting stock prices (and therefore option prices) can be investigated.

Table 3: Design of Experiment Levels

Parameter	Symbol	Levels
Period Length (yrs)	Δt	0.0625, 0.25
Number of Periods	N	4
Strike Price	K	90, 100, 110
Stock Volatility	σ	0.2, 0.3
Jump Frequency	λ	3, 7
Jump Size (upward)	η_1	25, 50
Jump Size (downward)	η_2	25, 50

As with any statistical test, more than one replication should be performed to prevent the possibility of outliers. To prevent this possibility, each simulation was run for 30 replications so that the possibility of the simulation having a bad starting seed did not affect the results. Additionally, since simulation uses the mean of the option values from each replication as the final output, more accurate results can be obtained by performing more replications. There is, however, a tradeoff associated with the number of replications performed – accuracy vs. computational time. As one may expect, more replications creates better accuracy at the expense of longer running times. The following section presents the results of the 96 experiments (with 30 replications each) performed with the simulation model presented in Chapter 3. Each simulation is modeling a put option, so the return of a path is the present value of the value of

$$\text{Max}\{\text{Strike Price} - \text{Market Price}, 0\}.$$

5.2 Results

The first set of tests conducted was for a planning horizon of 0.25 years. This corresponds to $N = 4$ and $\Delta t = 0.0625$. The results of each of the 48 simulations are summarized in Table 4. This table also presents the value of the options calculated by Kou's (2004) quadratic approximation equation and Amin's (1991) binomial tree method; Amin's binomial tree method was performed using a total of 1,600 steps.

Looking at Table 4 one fact is evident. Pricing American options with jump-diffusion using Monte Carlo simulation provides consistently higher values than that of the quadratic approximation and the binomial tree methods. Of the 48 experiments, only three configurations (or 6.25%) resulted in values lower than the values calculated by both Kou and Amin. However

these values are insignificantly lower, since the maximum difference was \$0.23. This outcome parallels the results presented by DeHaven (2007) when she compared least-squares Monte Carlo simulation with the finite-differences method. Monte Carlo simulation consistently provides higher values for American options due to the continuous nature of the pricing region and the fact that simulation creates multiple test trials of the stock paths instead of using mathematical equations to estimate the prices. Monte Carlo simulation provides the average expected value of each of the simulated stock price paths which provides a much more robust methodology than other pricing schemes used by practitioners.

A second noticeable result is when the stock is deep in-the-money. When the strike price is at \$110, the option begins \$10 in-the-money. This leads to a very interesting question as to why the values presented by Amin (1991) and Kou (2004) are only \$0.26 to \$2.23 more than the initial \$10 profit. By looking at standard Brownian motion alone (without any possibilities of jump), the probability that the stock price increases by more than \$10 (to reach the \$110 strike price) within $\frac{1}{4}$ of a year is small. Refer to Figures 3-3 and 3-4 from periods zero to four (which would be $\frac{1}{4}$ of a year). In these graphs, the number of jumps is set to zero ($\lambda = 0$) which means that the standard Black and Scholes model is being used to create the simulated paths. These figures show the likelihood that the stock prices increase by more than \$10 in $\frac{1}{4}$ of a year. Even with a variation of 30%, the likelihood is small. It is more evident that the majority of the paths are below the \$110 strike price, indicating higher expected returns than presented by Kou and Amin.

This means that the profit for the deep in-the-money options should be higher, if not significantly higher when jumps are included, than \$10 which is the case when the options are priced using Monte Carlo simulation. Monte Carlo simulation evaluates the option ranging from \$4.30 to \$7.56, meaning that the average ending stock price ranged from \$105.70 to \$102.44, respectively. As previously stated, Monte Carlo simulation takes the mean of 6,000 simulated stock paths (200 paths X 30 replications), whereas the other two methods calculate the expected value of mathematical approximation equations that may not accurately represent that actual stock price behavior. Due to this fact, Kou's quadratic approximation equation provides very conservative option values when the option is deep in-the-money. When the option is out-of-the-money Kou's quadratic approximation equation and Amin's binomial tree method are not as conservative and provide comparable results as Monte Carlo simulation.

Table 4: Monte Carlo simulation comparison for American put option with $t = 0.25$ Year

K	σ	λ	η_1	η_2	Binomial Tree	Kou (2004)	MCS
90	0.2	3	25	25	0.75	0.76	0.82
90	0.2	3	25	50	0.65	0.66	0.65
90	0.2	3	50	25	0.68	0.69	0.85
90	0.2	3	50	50	0.59	0.60	0.68
90	0.2	7	25	25	1.03	1.04	0.97
90	0.2	7	25	50	0.82	0.83	0.60
90	0.2	7	50	25	0.87	0.88	1.07
90	0.2	7	50	50	0.66	0.67	0.68
90	0.3	3	25	25	1.92	1.93	2.24
90	0.3	3	25	50	1.85	1.86	2.06
90	0.3	3	50	25	1.84	1.85	2.33
90	0.3	3	50	50	1.77	1.78	2.15
90	0.3	7	25	25	2.19	2.20	2.34
90	0.3	7	25	50	2.03	2.03	1.92
90	0.3	7	50	25	2.01	2.02	2.18
90	0.3	7	50	50	1.84	1.85	2.11
100	0.2	3	25	25	3.78	3.78	4.83
100	0.2	3	25	50	3.66	3.66	4.55
100	0.2	3	50	25	3.62	3.62	4.90
100	0.2	3	50	50	3.50	3.50	4.65
100	0.2	7	25	25	4.26	4.27	5.10
100	0.2	7	25	50	4.01	4.02	4.35
100	0.2	7	50	25	3.91	3.91	5.35
100	0.2	7	50	50	3.64	3.64	4.65
100	0.3	3	25	25	5.63	5.62	7.16
100	0.3	3	25	50	5.55	5.54	6.95
100	0.3	3	50	25	5.50	5.50	7.29
100	0.3	3	50	50	5.42	5.41	7.04
100	0.3	7	25	25	5.99	5.99	7.31
100	0.3	7	25	50	5.81	5.81	6.67
100	0.3	7	50	25	5.71	5.71	7.59
100	0.3	7	50	50	5.52	5.51	7.03
110	0.2	3	25	25	10.48	10.43	14.79
110	0.2	3	25	50	10.42	10.38	14.50
110	0.2	3	50	25	10.36	10.31	14.87
110	0.2	3	50	50	10.31	10.26	14.61
110	0.2	7	25	25	10.81	10.79	15.05
110	0.2	7	25	50	10.68	10.64	14.30
110	0.2	7	50	25	10.51	10.47	15.31
110	0.2	7	50	50	10.39	10.34	14.61
110	0.3	3	25	25	11.90	11.86	17.12
110	0.3	3	25	50	11.84	11.79	16.91
110	0.3	3	50	25	11.78	11.73	17.25
110	0.3	3	50	50	11.72	11.67	17.01
110	0.3	7	25	25	12.23	12.19	17.26
110	0.3	7	25	50	12.09	12.05	16.62
110	0.3	7	50	25	11.94	11.90	17.56
110	0.3	7	50	50	11.80	11.75	16.98

The last, yet very important, concept that can be observed through the results is the enormous effect the standard deviation has on the value of the option. Looking at each of the strike prices independently, there is only a significant value increase when the standard deviation (σ) increases from 0.2 to 0.3. In fact, there is approximately a 43% difference between the

maximum and minimum values (for $K = 90$) for the group with $\sigma = 0.2$ and an 18% difference between the maximum and minimum values of the group with $\sigma = 0.3$. However there is a 64% increase, which is significantly higher, between the two groups. This trend is consistent with the other two valuation methods and is also applicable for each of the other strike prices.

This indicates that for relatively small jump sizes ($\eta_i \geq 20$), the standard deviation – or in other words the diffusion process modeled by the Black and Scholes model – has a larger impact than the jump processes on the value of the option. This is under the basic assumption that there is not an abnormally large amount of jumps within the planning horizon. Common sense says that a large number of jumps in the planning horizon will affect the value of the option. Logically thinking, the number of jumps and the standard deviation of a stock should be related. A large amount of jumps within a small amount of time would indicate the company is extremely risky, which would in turn increase the standard deviation of the stock since the amount of risk is the fundamental basis of the stock's standard deviation. If jumps in a stock are frequent and of a small magnitude then these jumps should not be considered jumps, but instead considered as increased variability that is accounted for in larger standard deviation values. As a result, the purpose of modeling the jump-diffusion process is to capture the infrequent and significant jumps in the stock price after market announcements have been made or major events occur that directly affect the markets.

Tables 5 and 6 below show the effects of a high frequency of jumps (each simulation has a simulated time horizon of one quarter of a year and an initial stock price of \$100 and included 30 replications). Namely, significantly higher values of λ create a larger spread in the stock prices (which was shown in Chapter 3 and can even be seen with small jumps sizes) making the value of the option increase by considerable amounts. The larger spreads, in essence, reflect larger amounts of stock variation. This fact is evident when inspecting the 95% confidence intervals for the option price. The half width of the confidence interval (which is shown in Tables 5 & 6) is larger for bigger values of λ . Namely, with all other parameters held constant the half width of $\lambda = 15$ is much larger than the half width of $\lambda = 1$. Since every other factor is held constant, the standard deviation must increase in order for the half width to increase.

Table 5: λ Significance Testing 1

K	σ	λ	η_1	η_2	Price	Half Width
90	0.2	15	10	10	4.34	0.1835
90	0.3	15	10	10	5.32	0.2057
100	0.2	15	10	10	9.17	0.2379
100	0.3	15	10	10	10.59	0.2677
110	0.2	15	10	10	19.12	0.2376
110	0.3	15	10	10	20.54	0.2680

Table 6: λ Significance Testing 2

K	σ	λ	η_1	η_2	Price	Half Width
90	0.2	1	10	10	0.95	0.0884
90	0.3	1	10	10	2.36	0.1396
100	0.2	1	10	10	5.00	0.1594
100	0.3	1	10	10	7.31	0.2124
110	0.2	1	10	10	14.96	0.1631
110	0.3	1	10	10	17.29	0.2141

The second set of tests conducted was for a planning horizon of one year. This corresponds to $N = 4$ and $\Delta t = 0.25$ and as one can notice, the same general trends are present in this test group. As in the previous test set, Monte Carlo simulation consistently values American options higher than that of Kou (2004) and Amin (1991). However, in this test group, there were more instances (five total) where the simulation provided lower values than the other two methods. Looking at these five instances, three of the five are the exact parameter combinations from the previous test group in which the simulation valued the option lower than the quadratic approximation or binomial tree methods. This fact shows that the least-squares method is consistent over small and large time horizons and consistently provides excellent results.

Table 7: Monte Carlo simulation comparison for American put option with $t = 1.0$ Year

K	σ	λ	η_1	η_2	Binomial Tree	Kou (2004)	MCS
90	0.2	3	25	25	2.91	2.96	3.38
90	0.2	3	25	50	2.70	2.75	2.72
90	0.2	3	50	25	2.66	2.72	3.80
90	0.2	3	50	50	2.46	2.51	3.12
90	0.2	7	25	25	3.68	3.75	3.67
90	0.2	7	25	50	3.24	3.29	2.27
90	0.2	7	50	25	3.12	3.20	4.66
90	0.2	7	50	50	2.66	2.72	3.08
90	0.3	3	25	25	5.79	5.85	7.05
90	0.3	3	25	50	5.65	5.70	6.37
90	0.3	3	50	25	5.58	5.64	7.61
90	0.3	3	50	50	5.43	5.49	6.93
90	0.3	7	25	25	6.42	6.49	7.03
90	0.3	7	25	50	6.09	6.15	5.48
90	0.3	7	50	25	5.92	6.00	8.28
90	0.3	7	50	50	5.59	5.65	6.74
100	0.2	3	25	25	6.60	6.62	8.36
100	0.2	3	25	50	6.36	6.37	7.35
100	0.2	3	50	25	6.26	6.29	9.11
100	0.2	3	50	50	6.01	6.03	8.12
100	0.2	7	25	25	7.75	7.62	8.44
100	0.2	7	25	50	7.07	7.09	6.26
100	0.2	7	50	25	6.83	6.88	10.09
100	0.2	7	50	50	6.28	6.31	7.91
100	0.3	3	25	25	10.10	10.13	12.85
100	0.3	3	25	50	9.94	9.96	11.97
100	0.3	3	50	25	9.83	9.87	13.61
100	0.3	3	50	50	9.67	9.70	12.73
100	0.3	7	25	25	10.81	10.86	12.63
100	0.3	7	25	50	10.46	10.49	10.54
100	0.3	7	50	25	10.22	10.29	14.32
100	0.3	7	50	50	9.85	9.89	12.44
110	0.2	3	25	25	12.37	12.32	18.15
110	0.2	3	25	50	12.17	12.11	17.16
110	0.2	3	50	25	12.04	12.00	18.91
110	0.2	3	50	50	11.84	11.79	17.92
110	0.2	7	25	25	13.29	13.27	18.26
110	0.2	7	25	50	12.85	12.79	16.1
110	0.2	7	50	25	12.54	12.54	19.95
110	0.2	7	50	50	12.08	12.03	17.72
110	0.3	3	25	25	15.79	15.76	22.64
110	0.3	3	25	50	15.63	15.59	21.77
110	0.3	3	50	25	15.51	15.49	23.41
110	0.3	3	50	50	15.36	15.32	22.56
110	0.3	7	25	25	16.51	16.51	22.42
110	0.3	7	25	50	16.17	16.14	20.36
110	0.3	7	50	25	15.89	15.91	24.19
110	0.3	7	50	50	15.53	15.52	22.26

CHAPTER 6 - EXTENSION TO BASKETS

The process of analyzing a single American option can be computationally challenging. Extending this to analyze multiple options that are correlated complicates matters even further. This chapter presents how copulas can be combined with least-squares Monte Carlo simulation to accurately evaluate American options where the underlying stocks follow a jump-diffusion process. The rest of this chapter begins by presenting how to incorporate the correlation between multiple options using the Cholesky Decomposition and concludes with the notation and algorithm used within the Monte Carlo simulation. The computational experiments are then presented in the next chapter to demonstrate the effectiveness of different independent parameters.

6.1 A Jump-Diffusion Simulation Model with Copulas

Utilizing copulas within the pricing formula is an effective way of modeling the correlation between options. In closed-form numerical approximations, it can be difficult to incorporate copulas into the formulas. However, by using Monte Carlo simulation this inclusion can be a fairly straightforward process. The simulation need only incorporate a subroutine to perform the Cholesky decomposition on the correlation matrix, which is explained in the following subsection, as well as another subroutine to generate the correlated random variates among the options within the same basket. The following sections explain the Cholesky decomposition in greater detail, present the additional notation required, and then propose the algorithm.

6.1.1 Cholesky Decomposition

The Cholesky decomposition is used in this research to generate multiple, correlated variates (in the Brownian motion and the correlated jump sizes). The purpose of the process is to decompose the correlation matrix between the random input variables in order to produce the correlated variates.

Formally, Cholesky decomposition is of the form $\Gamma = AA^T$, where Γ is the correlation matrix between the options (Γ must be a symmetric positive definite matrix) and A is the

covariance matrix that is lower triangular. The goal is to calculate A . The Cholesky decomposition follows that

$$\Gamma = AA^T = \begin{pmatrix} \alpha_{11} & 0 & 0 \\ \alpha_{21} & \alpha_{22} & 0 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{pmatrix} \begin{pmatrix} \alpha_{11} & \alpha_{21} & \alpha_{31} \\ 0 & \alpha_{22} & \alpha_{32} \\ 0 & 0 & \alpha_{33} \end{pmatrix} = \begin{pmatrix} \alpha_{11}^2 & \alpha_{21}\alpha_{11} & \alpha_{31}\alpha_{11} \\ \alpha_{21}\alpha_{11} & \alpha_{21}^2 + \alpha_{22}^2 & \alpha_{31}\alpha_{21} + \alpha_{32}\alpha_{22} \\ \alpha_{31}\alpha_{11} & \alpha_{31}\alpha_{21} + \alpha_{32}\alpha_{22} & \alpha_{31}^2 + \alpha_{32}^2 + \alpha_{33}^2 \end{pmatrix}.$$

From this, we obtain the following formula for each of the values of A :

$$A_{i,j} = \frac{1}{A_{j,j}} \left(\Gamma_{i,j} - \sum_{k=1}^{j-1} A_{i,k} A_{j,k} \right), \text{ for } i > j$$

and

$$A_{i,i} = \sqrt{\Gamma_{i,i} - \sum_{k=1}^{i-1} A_{i,k}^2}.$$

A is then multiplied by each of the random variates to incorporate the correlation between them. For example, given that x_1, x_2 , and x_3 are three independent random variates and y_1, y_2 , and y_3 are the random correlated variates, y_1, y_2 , and y_3 have the properties of the correlation matrix Γ and are calculated by

$$[y_1, y_2, y_3]^T = A[x_1, x_2, x_3]^T$$

where A is calculated through the Cholesky decomposition.

6.1.2 Notation

The majority of the notation is exactly the same as in Chapter 3. The only additional variables that must be added to utilize copulas are the following:

$O \equiv$ Number of options in the basket

$\Gamma \equiv O \times O$ correlation matrix

$A \equiv \Gamma$ decomposed using Cholesky Decomposition

6.1.3 Updated Algorithm

Step 0. System Inputs:

- (a) Initial stock price (S_0)
- (b) Risk-free interest rate (r)
- (c) Strike price (K)
- (d) Stock volatility (σ)

- (e) Number of time periods (N)
- (f) Number of paths (M)
- (g) Length of time horizon in years (T)
- (h) Call or put option
- (i) Exponential Distribution parameters (η_1 and η_2)
- (j) Probability of Upward Jump (p)
- (k) Rate of Poisson Process (λ)
- (l) Number of options (O)
- (m) $O \times O$ Correlation Matrix (Γ)

Step 1. Initialization:

- (a) Set $\Delta t = T/N$
- (b) Perform Cholesky decomposition on correlation matrix Γ to obtain matrix A .
- (c) Set the seed for the initial path to any positive integer.
- (d) Generate the random numbers Z_{ijk} for each path $j = 1, 2, \dots, M$, period $i = 1, 2, \dots, N$, and option $k = 1, 2, \dots, O$.
- (e) Generate the random number $J_{ijk}(t)$ for each path $j = 1, 2, \dots, M$, period $i = 1, 2, \dots, N$, and option $k = 1, 2, \dots, O$ and calculate the jump size for each period i , path j , and option k using the equation:

$$V_{ijk} = \prod_{n=1}^{J(t)} V_n, \text{ where } V \text{ follows the double exponential distribution.}$$

*Note: If a jump is not present, $V_{ijk} = 1$.

- (f) Calculate the correlated random variates Z_{ijk} and V_{ijk} using:

$$Z_{ijk} = Z_{ijk} * A \text{ for each } k = 1, 2, \dots, O.$$

$$V_{ijk} = (V_{ijk} - 1) * A \text{ for each } k = 1, 2, \dots, O.$$

*Note: The value of $(V_{ijk} - 1)$ is used to get the actual jump size. An upward jump will result in a positive value and a downward jump will result in a negative value.

- (g) Set $V_{ijk} = V_{ijk} + 1$, which is merely adding 1 back into the correlated values to make the jump values positive.

Step 2. Compute S_{Njk} for the expiration date T for each path of option O using:

$$S_{ijk} = S_{i-1,jk} \exp([r - \frac{1}{2}\sigma^2]\Delta t + \sigma\sqrt{\Delta t}Z_{ijk}) * V_{ijk}. \quad (6.1)$$

Step 3. Compute the cash flows for each path and option using one of the following:

$$P(j) = \begin{cases} \max\{K - S_{ijk}(t), 0\} & \text{put option} \\ \max\{S_{ijk}(t) - K, 0\} & \text{call option} \end{cases} \quad (6.2)$$

Step 4. Backup one time period; set $i = i - 1$.

- (a) Compute S_{i-1} using:

$$S_{ijk} = S_{i+1,jk} / \exp([r - \frac{1}{2}\sigma^2]\Delta t + \sigma\sqrt{\Delta t}Z_{ijk}) / V_{ijk}. \quad (6.1)$$

Step 5. Compute if the option is in the money for each path j . For each path:

- (a) Let X be the vector containing asset prices S_{ijk} and Y be the vector containing the corresponding cash flows received at $i+1$ time period, which have been discounted backward to the i^{th} time period.
- (b) Regress using least-squares approach to estimate the value of continuing using the equation. (3.5). This will result in the conditional expectation function $E[Y|X]$.
- (c) Compute the value of continuing using $E[Y|X]$ and the value of immediately exercising using equation (6.2).
- (d) Determine whether to exercise the option immediately or hold the option until the next time period, based on which gives the higher value. Establish the current cash flows conditional on not exercising prior to time period i using:

$$C_i(j) = \begin{cases} \text{cash flow} & \text{if cash flow} \geq E[Y|X] \\ 0 & \text{otherwise} \end{cases} \quad (6.3)$$

(e) Compute the present value of the cash flows $P_{ik}(j)$ given by:

$$P_{ik}(j) = C_{ik}(j) + e^{-r\Delta t} P_{ik}(j). \quad (6.4)$$

Step 6. If at time period zero proceed to Step 7, else go back to Step 4.

Step 7. Proceed to next option; set $k = k + 1$.

Step 8. If $k = O + 1$ stop, else go back to Step 2.

CHAPTER 7 - COMPUTATIONAL RESULTS FOR CORRELATED BASKETS

The results of the computational experiments for the algorithm proposed in the previous chapter are presented within this chapter. This chapter begins with a description of the experimental design that was used to test the effectiveness of extending least-squares Monte Carlo simulation to the pricing of baskets of American options, followed by a description of the basic dynamics of correlated stocks. The final section of the chapter discusses the results of the experiments performed.

7.1 Copula Design of Experiment

To determine the applicability of evaluating baskets of American options using Monte Carlo simulation, the algorithm in Chapter 6 was used to evaluate the baskets. Namely, each of the parameters was systematically changed to determine the effects of each parameter on the value of the option. The parameters and the number of factors tested within these experiments are identical to those used in Chapter 5 with one additional parameter. The additional parameter tested within this chapter is the correlation matrix, which contains two levels (highly correlated and moderately correlated). These two levels are shown in the matrices below.

$$\Gamma_{\text{low}} = \begin{bmatrix} 1 & 0.2 & 0.25 \\ 0.2 & 1 & 0.2 \\ 0.25 & 0.2 & 1 \end{bmatrix}, \quad \Gamma_{\text{high}} = \begin{bmatrix} 1 & 0.9 & 0.85 \\ 0.9 & 1 & 0.95 \\ 0.85 & 0.95 & 1 \end{bmatrix}$$

A quarter fractional factorial design of experiment was used to test the effectiveness of different treatments for each of the three strike prices (90, 100, and 110). A total of 24 experiments (treatments) were tested between the three strike prices using the parameter values given in the following table for each of the strike prices. As in the experiments performed in Chapter 3, each of the simulations contained 200 paths and was run for 30 replications to reduce the possibility of outliers affecting the results of the experiments. The subsequent section presents the simulated movement dynamics of the correlated stocks.

Table 8: Copula Design of Experiment

Γ	σ	λ	η_1	η_2
High	0.2	3	25	25
High	0.2	7	25	50
High	0.3	3	50	25
High	0.3	7	50	50
Low	0.2	3	50	50
Low	0.2	7	50	25
Low	0.3	3	25	50
Low	0.3	7	25	25

7.2 Copula Dynamics

The purpose of this section is to detail how correlation affects the movements of correlated stock prices. The first step of the algorithm performs the Cholesky decomposition of the correlation matrix. The resulting decomposed matrix is given below:

$$\Gamma_{\text{low}} = \begin{bmatrix} 1 & 0 & 0 \\ 0.2 & .979796 & 0 \\ 0.25 & 0.153093 & 0.956066 \end{bmatrix}, \quad \Gamma_{\text{high}} = \begin{bmatrix} 1 & 0 & 0 \\ 0.9 & 0.43589 & 0 \\ 0.85 & 0.424419 & 0.312039 \end{bmatrix}$$

These are the matrices that the uncorrelated Z and V matrices are multiplied by to obtain correlated values. These correlated values are then used within the calculations of the stock prices. For simplicity and comparison reasons, the initial stock prices are set to the same value of \$100 for all of the treatments; however, the simulation allows differing initial stock prices for the underlying stocks.

Figures 7-1 through 7-3 show the simulated paths of three separate stocks that are highly correlated (e.g. are calculated using Γ_{high}) with the following input parameters: $\sigma = 0.2$, $\lambda = 3$, $\eta_1 = 25$, and $\eta_2 = 25$. Notice how the general movements of the three figures are very similar. When the standard diffusion process increases in stock 1, the standard diffusion process in stocks 2 and 3 consequently increase. When the standard diffusion process decreases in stock 1, the same can be said for stocks 2 and 3. This is due to the high correlation between the three stocks. Each has its respective natural variation, however, stocks 2 and 3 are also dependent on the natural variation of the other stocks. The same principle can be applied to generate the correlated jump sizes, if present. If a jump is present in stock 1, this jump will in turn cause

stocks 2 and 3 to jump, even if there was not an actual simulated jump in stocks 2 or 3. Again, this is because of the correlation between the stocks.

Comparing Figures 7-1 through 7-3 to Figures 7-4 through 7-6, the effects of correlation are evident. Notice how in the moderately correlated stocks the general diffusion processes are quite different. Stocks 2 and 3 do not follow the same general trends as stock 1 as they did with highly correlated stocks. This extreme difference can be easily seen when looking at stock 2. Notice in Figure 7-5 the uppermost stock price decreases between periods 4 and 5. However, looking at this same path in Figure 7-2, the stock price actually increases significantly between periods 4 and 5. This is the effect of the correlation between the stocks and the difference between moderate and high correlation.

Overall, high correlation causes very balanced option prices since each of the simulated stock paths follow the same general trends. Therefore, if the initial stock prices for the O different options are the same and each are highly correlated, then the value of the basket will be approximately $O \cdot W_1$, where W_1 is the average value of the first option. However, if the O options are not strongly correlated, then the values of the options may not be as similar and therefore the value of each option in the basket can vary by significant amounts, especially if the average jump size is large.

Now that the general effects of correlation have been presented, the subsequent section provides the results of the experiments conducted within this portion of the thesis.

7.3 Copula Results

The parameter combinations described in the first section of Chapter 7 were performed for each of the three strike prices (\$90, \$100, and \$110) with an initial stock price of \$100. The design of experiment resulted in 24 total treatments being tested to determine the relationship between the option value and the input parameters, including the correlation.

Table 9, below, summarizes the results of the experiments for a planning horizon of 0.25 years, which corresponds to $N = 4$ and $\Delta t = 0.0625$. The tests performed and the results obtained from the use of copulas are very similar to the experiments and results found in Chapter 5. Namely, the standard deviation of the stock still has the largest affect on the value of the basket. For each of the three strike prices the largest valued baskets contain a higher standard deviation.

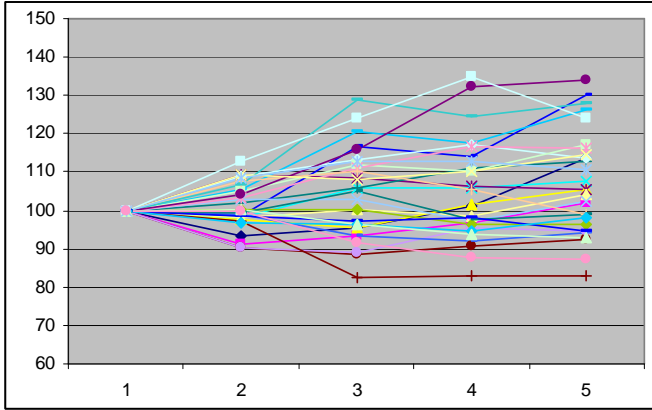


Figure 7-1: Stock 1 High Correlation

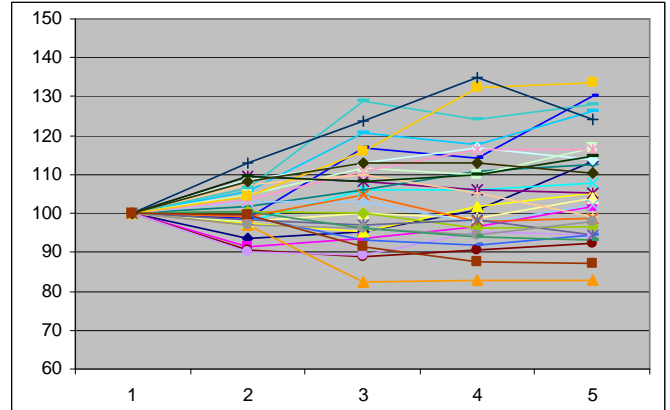


Figure 7-2: Stock 1 Low Correlation

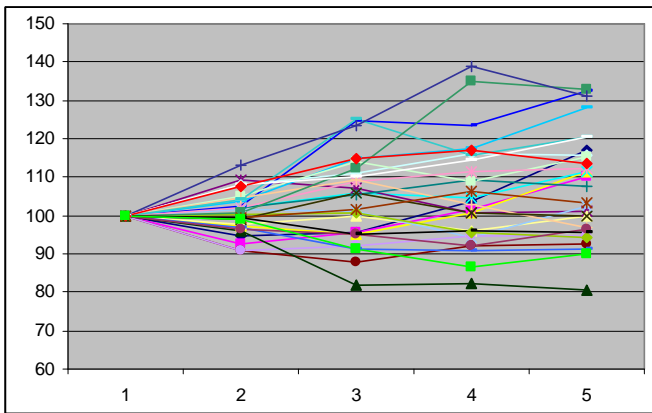


Figure 7-3: Stock 2 High Correlation

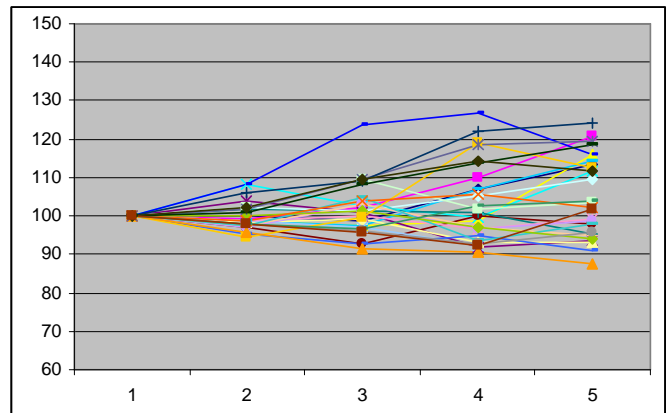


Figure 7-4: Stock 2 Low Correlation

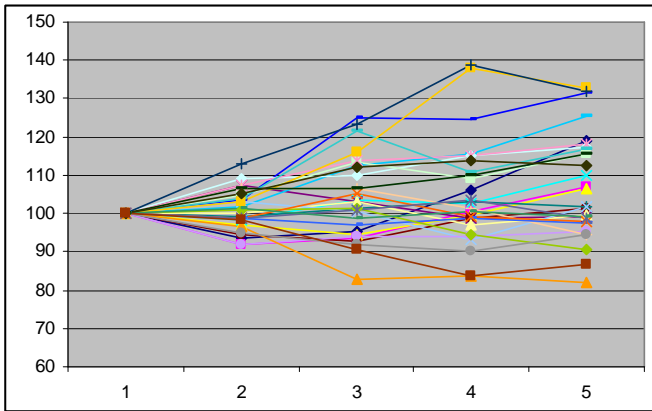


Figure 7-5: Stock 3 High Correlation

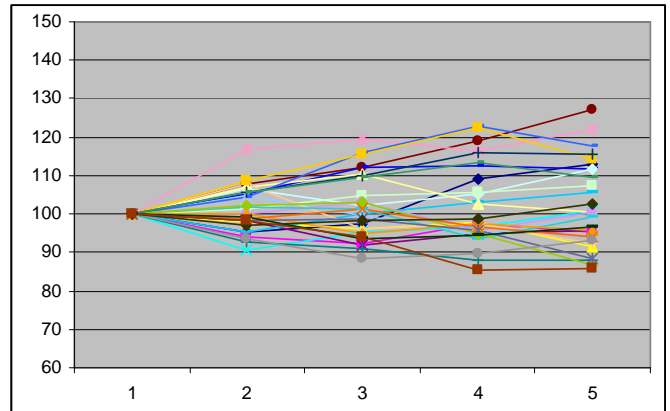


Figure 7-6: Stock 3 Low Correlation

Additionally, Table 9 shows that Monte Carlo simulation accurately prices baskets of American options in the same manner as individual options. This can be seen by the fact that the basket value is approximately 3 (since 3 options were modeled in this basket example) times the value of the first option – since the starting seeds are the same for both simulations, the first

option value is given in Table 4 in Chapter 5. This indicates that the pricing of correlated options using Monte Carlo simulation in a multi-dimensional domain provides accurate results that are similar to those in the single-dimensional domain. The fact that the multidimensional correlated values are closely related to the one-dimensional values indicates that Monte Carlo simulation using least-squares regression provides a very quick and accurate means of valuing baskets of American options with jump-diffusion.

Table 9: Copula Results

K	Γ	σ	λ	η_1	η_2	Price
90	High	0.2	3	25	25	2.36
90	High	0.2	7	25	50	1.61
90	High	0.3	3	50	25	7.00
90	High	0.3	7	50	50	6.22
90	Low	0.2	3	50	50	1.99
90	Low	0.2	7	50	25	3.26
90	Low	0.3	3	25	50	6.00
90	Low	0.3	7	25	25	6.87
100	High	0.2	3	25	25	14.35
100	High	0.2	7	25	50	12.23
100	High	0.3	3	50	25	21.98
100	High	0.3	7	50	50	20.95
100	Low	0.2	3	50	50	13.89
100	Low	0.2	7	50	25	16.13
100	Low	0.3	3	25	50	20.63
100	Low	0.3	7	25	25	21.72
110	High	0.2	3	25	25	44.23
110	High	0.2	7	25	50	42.09
110	High	0.3	3	50	25	51.86
110	High	0.3	7	50	50	50.81
110	Low	0.2	3	50	50	43.77
110	Low	0.2	7	50	25	46.03
110	Low	0.3	3	25	50	50.49
110	Low	0.3	7	25	25	51.59

CHAPTER 8 - MEMORY REQUIREMENTS

The memory requirements for each of the simulation algorithms are presented in Chapter 8. The purpose of this chapter is to overview and compare the memory requirements for each of the algorithms proposed within this thesis.

8.1 Memory Requirements for Simulation Models

Two of the three least-squares algorithms presented in this thesis utilize a memory reduction technique that was first presented by Chan (2003). As explained in Chapter 2, Chan presented a method that does not store the intermediate stock prices; instead the seed values are calculated two times, once to generate the forward paths and once to generate the backward paths. As a result computational requirements increase, but the memory requirements decrease significantly. This is an acceptable tradeoff since the speed and computational power of computers is continuously increasing, making the importance of decreased computational requirements decline. However, the size of the problems being solved by computers is continuously increasing making the storage requirements grow drastically. This drastic growth forces researchers/practitioners to have expensive computers that contain tremendous amounts of memory.

Chapter 3 describes two pricing algorithms that utilize Chan's (2003) memory reduction technique. Utilizing this technique allows each of these algorithms to have a memory requirement in the magnitude of $O(M)$, where M is the number of paths being simulated. Without this reduction, the memory requirement would be in the magnitude of $O(MN)$ where N is the number of periods simulated. Since the pricing accuracy is directly related to the number of periods being simulated (since the discretized time periods approach a continuous region as N approaches infinity), a higher number of periods must be simulated to obtain highly accurate prices. As a result, without Chan's memory reduction technique the memory requirements would be substantial. The algorithm presented in Chapter 6 does not utilize Chan's memory reduction techniques. As a result, the memory requirements for this algorithm is in the order of $O(MNO)$, where M is the number of paths, N is the number of periods, and O is the number of options in the basket.

CHAPTER 9 - CONCLUSIONS

The market activities over the last decade have shown that the Black and Scholes (1971) model does not entirely reflect all of the market behaviors. As a result, the Black and Scholes model should be enhanced to include both the jump and diffusion processes so that “jumps” in the market are captured. One of the most obvious jumps was the 684 point drop in the Dow industrial average on September 17, 2001, the next trading day after the 9-11 attacks on the United States (Kauffman 2001). The standard Brownian motion that is modeled within the Black and Scholes model does not have the capability of modeling such a significant and immediate change in stock price. There have been many instances just like this where the market has risen/fallen by tremendous amounts over a short period of time. This fact illustrates the need for an enhancement of the Black and Scholes model.

Merton (1973) was the first to present the idea of jumps in the markets. In recent years Kou (2004), among others, has continued the research started by Merton. Kou (2004) has shown that Merton’s use of the normal distribution to model the jump sizes is not as accurate as using the double exponential distribution. The purpose of this thesis was to explore the use of Kou’s (2004) jump-diffusion model to price American options using Monte Carlo simulation. Additionally, this thesis investigates the extension of this Monte Carlo simulation model to a multidimensional domain to explore the effectiveness of pricing baskets of American options.

The remaining sections of Chapter 9 are as follows. Section 9.1 begins with the conclusions obtained through the research presented within this thesis and Section 9.2 presents future work that could be investigated further.

9.1 Conclusion on Monte Carlo Simulation

This thesis has shown that Monte Carlo simulation provides an effective means for analyzing and pricing American options where the underlying stock follows a jump-diffusion process. It also demonstrated the flexibility of Monte Carlo simulation with the ability to make simple modifications to an existing simulation model to incorporate an additional, yet drastically different, stochastic process. The simulation presented within this thesis extended on the model presented by DeHaven (2007) to include the jump-diffusion process explained by Kou (2004).

The experiments conducted within this thesis have shown that, in general, Monte Carlo simulation values American options higher than that of other numerical techniques, including the techniques presented by Kou (2004) and Amin (1991). Monte Carlo simulation provides a means of accurately modeling the behaviors of the stock market in all aspects, including the natural stock variation (diffusion process) as well as the possible jumps that occur (jump process).

The experiments performed within this thesis have exemplified the ability to accurately model the natural stock price behavior as well as the jump processes. From these experiments, it was determined that when the jump sizes and the jump frequencies are small, the factor that has the largest effect on the option value is the standard deviation of the stock. The simulations performed show that values of η_i larger than 25 (which indicate average jump sizes of 4% or smaller) seem to have little effect on the option price. The major contributor to the options price, when the jump sizes and frequency is small, is the standard deviation. The results presented in Chapter 5 show how drastically the options price changes as the standard deviation increases from 0.2 to 0.3. Though the option's value is only moderately affected by the size and frequency of the jumps if these values are small, these factors begin to have a large affect on the option value as they increase. This is due to the added deviation that is created by the frequent, and possibly extreme, jumps in the stock's price. The added deviation increases the spread of the simulated paths and, in turn, increases the price of the option. The wider spreads are caused by an increased number of outlying stock paths that are present after large jumps occur.

The aforementioned experiments provided evidence of another fact when comparing the results of the Monte Carlo simulation to the results of Kou's (2004) approximation algorithm and Amin's (1991) binomial tree method. Namely the results indicate that when the option is out-of-the-money, all three methods accurately calculate the value of the option. However, as the option becomes at-the-money or in-the-money, Kou's approximation algorithm and Amin's binomial tree method become extremely conservative in the valuation process. Monte Carlo simulation, however, is less conservative and calculates a much higher price for the American option (e.g. \$14.79 versus \$10.43). Thinking about the prices logically, the value produced by the Monte Carlo simulation is more accurate since the option begins \$10 in-the-money. It only makes sense that the option should be valued higher than \$0.43 more than the initial profit, especially with the possibility of jumps. Overall, this shows the accuracy of Monte Carlo

simulation over the wide range of option values, from deep out-of-the-money to deep-in-the-money.

The second portion of this thesis shows that Monte Carlo simulation provides an accurate and effective means for calculating baskets of American options when the underlying stocks are correlated. Utilizing the benefits of copulas, a simulation model was presented that accurately models the correlation between the stocks. This was shown through multiple examples of highly and moderately correlated stocks with a variety of different input parameter combinations. When the underlying stocks are highly correlated, the general movements of the three stocks are remarkably similar. These similarities are expected (since the stocks are highly correlated) and show that the simulation accurately models the stocks, even when correlation is involved. When the stocks are moderately correlated, the general movements of the three stocks are fairly independent and are only moderately effected by the movements of the other two stock prices. Again, these movements are expected since the stocks are only moderately correlated, which also shows the correctness of the Monte Carlo simulation in pricing baskets of American options.

As with the pricing of single dimensional options, the major determining factor of the basket price when the average jump sizes are small is the volatility of the underlying stocks. Increasing the standard deviation from 0.2 to 0.3 has the same effects for the baskets as it did for the single option, drastically increased prices. Baskets also resemble the results of single options when the average jump size is large. When jump sizes are large, the price of the basket increases due to the added variability in the underlying stocks.

An additional concluding comment that must be made is in regards to the flexibility of Monte Carlo simulation. In order to extend the previous simulation model the only modification required is a few additional subroutines to incorporate the copulas (via the Cholesky decomposition) and the jump-diffusion process. Comparing this to other numerical methods that use complex integral equations, it is much more difficult to include the required factors. Instead of a few additional subroutines, the complex integral equations would require exact distributions (if they are available) and complex mathematical formulas to create the correlation between the jumps and stock prices. This is a significant advantage of Monte Carlo simulation which makes it an attractive method for pricing both single dimensional options and multidimensional baskets.

9.2 Future Work

This research uncovered some interesting questions that could be researched further. Three aspects of the Monte Carlo simulation involving copulas should be investigated further. These include utilizing Chan's (2004) memory reduction technique, investigating the process of calculating the correlation between the jumps, and modifying the input parameters so that each of the underlying stocks have separate input parameters (e.g. average jump size, average number of jumps, and standard deviation). This thesis held the input parameters constant among options to simplify the analysis of the Monte Carlo simulation.

Chapter 3, which implemented Chan's (2003) reduced memory algorithm, presented an algorithm for pricing American options with jump-diffusion that does not store all of the intermediate prices (and corresponding values to calculate the prices; e.g. Z_i and V_i). As a result, the memory requirements are far less expensive as other methods that store each of the prices. The only cost of this method is the computational requirements since each of the seed values must be calculated twice, once for the forward pricing and once for the backward pricing. This memory reduction technique has the potential to work when pricing multiple options in a multidimensional domain. The research contained within this thesis does not utilize Chan's (2003) reduced memory method and as a result requires a large amount of memory to perform the simulation. The required memory has the potential to be reduced significantly by implementing Chan's (2003) algorithm so that the memory requirements would grow in the order of $O(MO)$ instead of $O(MNO)$, where M is the number of paths simulated, O is the number of options, and N is the number of periods. Reducing the amount of memory required to run the simulation would be a significant enhancement, especially if the researcher/practitioner is evaluating a large number of options or periods.

The second factor that should be researched further is how the jump correlation is calculated. The process used within this research correlates the grouped jump size for a given period instead of correlating the individual jumps. In other words, within the algorithm the Poisson process determines the number of jumps within a given period ($J(t)$) and the double exponential is calculated and multiplied by itself $J(t)$ times. This process is grouping the $J(t)$ jumps during that period into a single, comprehensive jump size. In the real markets when a jump occurs for a stock, the correlation between the other stocks can be seen within a short amount of time. As a result, calculating the correlation between the individual jumps instead of

the grouped jumps may provide a more accurate representation of the correlation between the stocks and therefore accurately model the true nature of the markets.

The last enhancement regarding the simulation model presented within this thesis is to allow each of the stocks to have different jump and standard deviation parameters. Though companies may be in similar industries, their jump frequencies, jump amplitudes, and standard deviations may still be quite different. The model presented within this thesis assumes that these parameters are the same for each of the stocks being analyzed. Therefore, a possible area of future research is to add stock-specific input parameters.

CHAPTER 10 - BIBLIOGRAPHY

- AitSahlia, Farid. 2007. "A Canonical Optimal Stopping Problem for American Options Under a Double Exponential Jump-Diffusion Model." *Journal of Risk*. **10**: 85-100.
- Achdou, Yves, and Olivier Pironneau. 2005. *Computational Methods for Option Pricing*. Philadelphia: Society for Industrial and Applied Mathematics.
- Amin, Kaushik. 1991. "On the Computation of Continuous Time Option Prices Using Discrete Approximations." *Journal of Quantitative Analysis*. **26**(4) : 477-496.
- Amin, Kaushik. 1993. "Jump Diffusion Option Valuation in Discrete Time." *Journal of Finance*. **48**(5): 1833-1863.
- Barone-Adesi, G., and R.E. Whaley. 1987. "Efficient Analytic Approximation of American Option Values." *Journal of Finance*. **42**: 301-320.
- Bluhm, Christian, and Ludger Overbeck. 2007. *Structured Credit Portfolio Analysis, Baskets & CDOs*. Boca Raton, FL: Chapman & Hall.
- Brennan, Michael, and Eduardo Schwartz. 1977. "Finite Difference Methods and Jump Processes Arising in the Pricing of Contingent Claims: A Synthesis." *Journal of Financial and Quantitative Analysis* 13: 461-474.
- Broadie, Mark, and Jerome Detemple. 2004. "Option Pricing: Valuation Models and Applications." *Management Science*. **50**: 1145-1177.
- Broadie, Mark, and Paul Glasserman. 2004. "A Stochastic Mesh Method for Pricing High-Dimensional American Options." *Journal of Computational Finance*. 7(4): 35-72.
- Boyle, P. 1977. "Options: A Monte Carlo Approach." *Journal of Financial Economics*. **4**: 323-338.
- Carr, P., R. Jarrow, R. Myneni. 1992. "Alternative Characterizations of American Puts." *Mathematical Finance*. **2**: 87-106.
- Cox, J.C., S.A Ross, M. Rubinstein. 1979. "Option pricing: A simplified approach." *Journal of Financial Economics*. **7**: 229-263.
- Chan, Raymond, Chi-Yan Wong, and Kit-Ming. Yeung. 2002. "Pricing American-Style Options by Monte Carlo Methods without Storing all the Intermediate Asset Prices." Chinese University of Hong Kong Mathematics Department Research Report #2002-11.
- Chan, Raymond, Yong Chen, and Kit-Ming. Yeung. 2003. "A Memory Reduction Method in Pricing American Options." *Journal of Statistical Computation and Simulation* 74(7): 501-511.
- Chan, Raymond, Chi-Yan Wong, and Kit-Ming Yeung. 2006. "Pricing Multi-Asset American-Style Options by Memory Reduction Monte Carlo Methods." *Applied Mathematics and Computation* 179(2): 535-544.

- Charns, John M. 2000. "Using Simulation for Option Pricing." In *Proceedings of the 2000 Winter Simulation Conference*, ed. J.A. Joines, R.R. Barton, K. Kang, and P.A. Fishwick.
- Chen, Xinfu, and John Chadam. 2006. "A Mathematical Analysis of the Optimal Exercise Boundary for American Put Options." *SIAM Journal on Mathematical Analysis*. **38**: 1613-641.
- Chen, Xinfu, John Chadam, and Robert Stamcar. 2000. "The Optimal Exercise Boundary for American Put Options: Analytic and Numerical Approximations."
- DeHaven, Sarah. 2007. "The Optimal Exercising Problem from American Options: A Comparison of Solution Methods." MS Thesis, Kansas State University.
- Dorey, Martyn, and Phil Joubert. "Modeling Copulas: An Overview." *The Staple Inn Actuarial Society*.
- Feng, Liming and Vadim Linetsky. 2008. "Pricing Options in Jump-Diffusion Models: An Extrapolation Approach." *Operations Research*. **56**(2): 304-325.
- Glasserman, Paul and Bin Yu. "Simulation for American Options: Regression Now or Regression Later?"
- Gukhal, C.R. 2001. "Analytical Valuation of American Options on Jump-Diffusion Processes." *Mathematical Finance*. **11**: 97-115.
- Hanson, F. B., and J. J. Westman "Stochastic Analysis of Jump-Diffusions for Financial Log-Return Processes," in *Stochastic Theory and Control, Proceedings of a Workshop held in Lawrence, Kansas, October 18-20, 2001, Lecture Notes in Control and Information Sciences*, B.Pasik-Duncan (Editor), Springer-Verlag, New York, pp. 169-184, 24 July 2002.
- Haugh, Martin. "Approximate Dynamic Programming (ADP) and Dual Methods for Pricing American Options." Course Notes. Monte Carlo Simulation. Industrial Engineering and Operations Research, Columbia University. 6 January 2009.
- Huang, Z., Kou, S.G. 2006. "First passage times and analytical solutions for options on two assets with jump risk." *Preprint*, Columbia University.
- Hull, John and Alan White. 1990. "Valuing Derivative Securities Using the Explicit Finite Difference Method." *Journal of Financial and Quantitative Analysis* 25: 87-100.
- Hull, John. 2006. *Options, Futures, and Other Derivatives*. Upper Saddle River, NJ: Pearson Education, Inc. 6th Ed.
- Jarrow, Robert, and Andrew Rudd. 1982. "Approximate Option Valuation for Arbitrary Stochastic Processes." *Journal of Financial Economics*. **10**(3): 347-369.
- Ju, N. 1998. "Pricing an American Option by Approximating its Early Exercise Boundary as a Multipiece Exponential Function." *Review of Financial Studies*. **11**: 627-646.
- Kauffman, Matthew. 2001. "Dow Drops Record 684 Points as Jittery Markets Reopen." *Hartford Courant*, September 18. <http://www.courant.com/topic/hc-attack-a1dow-0918,0,5749796.story>.

- Kou, S.G. 2002. "A Jump-Diffusion Model for Option Pricing." *Management Science*. **48**: 1086-1101.
- Kou, S.G. 2004. "Option Pricing Under a Double Exponential Jump Diffusion Model." *Management Science*. **50**: 1178-1192.
- Kou, S.G, Gionvanni Petrella, and Hui Wang. 2005. "Pricing Path-Dependent Options with Jump Risk via Laplace Transforms." *The Kyoto Economic Review*. **74**(1): 1-23.
- Kou, S.G. 2008. "Jump-Diffusion Models for Asset Pricing in Financial Engineering." In *Handbooks in Operations Research and Management Science Vol. 15*, ed. J.R. Birge and V. Linetsky, 73-116. Oxford: North-Holland.
- Law, Averill and W. David Kelton. 2000. *Simulation Modeling and Analysis*. Boston: McGraw-Hill. 3rd Ed.
- Lehmer, D.H. (1951). "Mathematical methods in Large-Scale Computing Units." *Ann. Comput. Lab. Harvard Univ.* **26**: 141-146.
- Longstaff, Francis, and Eduardo Schwartz. 2001. "Valuing American Options by Simulation: A Simple Least-Squares Approach." *The Review of Financial Studies*. **14**(1): 113-147.
- Liu, Qiang. 2008. "Constructing the Optimal Exercise Boundary for American Options by Least-Squares Monte Carlo."
- Merton, R.C. 1976. "Option Pricing When Underlying Stock Returns are Discontinuous." *J. Financial Economics*. **3**: 125-144.
- Pham, H. 1997. "Optimal Stopping, Free Boundary and American Option in a Jump-Diffusion Model." *Applied Mathematics and Optimization*. **35**: 145-164.
- Quittard-Pinon, Francois and Rivo Randrianarivony. (2007). "How to Price Efficiently European Options in Some Geometric Lévy Processes Models?" *Working Paper*. <http://ssrn.com/abstract=998312>.
- Srinivas, S., D. Menon, A. Meher Prasad. 2006. "Multivariate Simulation and Multimodal Dependence Modeling of Vehicle Axle Weights with Copulas." *Journal of Transportation Engineering*. **132**(12): 945-955.
- Staum, Jeremy. 2001. "Simulation in Financial Engineering." In *Proceedings of the 2001 Winter Simulation Conference*, ed. B.A. Peters, J.S. Smith, D.J. Medeiros, and M.W. Rohrer, 123-133.
- Steverman, Ben. 2008. "Stocks: Rating the 2008 Meltdown." *Businessweek*, November 3. http://www.businessweek.com/investor/content/oct2008/pi20081031_960768.htm.
- Tilley, J. 1993. "Valuing American Options in a Path-Simulation Model." *Transactions of the Society of Actuaries*. **45**: 563-577.
- Wilmott, Paul. 1998. *Derivatives: The Theory and Pricing of Financial Engineering*. Chichester: Wiley.
- Zhang, Chensong. 2005. "Pricing American Options by Adaptive Finite Element Method." Paper presented at the Spotlight on Graduate Research Competition, December 15, at University of Maryland.

Zhu, Zongwu. 2005. "Option Pricing and Jump-Diffusion Models." Doctor of Philosophy in Mathematics Dissertation, University of Illinois at Chicago.

Zhu, Zongwu and F.B Hanson. 2005. "A Log-Double-Uniform Jump-Diffusion Model." *Preprint*, University of Illinois at Chicago.

Appendix A - Derivation of the Roots of $G(x)$

Further information regarding the following equations can be found in Kou (2005). The moment generating function $X(t)$ for a jump-diffusion process can be obtained as

$$E[e^{\theta X(t)}] = e^{G(\theta)t} \text{ where}$$

$$G(x) := x \left(r - \delta - \frac{1}{2} \sigma^2 - \lambda \zeta \right) + \frac{1}{2} x^2 \sigma^2 + \lambda \left(\frac{p \eta_1}{\eta_1 - x} + \frac{q \eta_2}{\eta_2 + x} - 1 \right).$$

The equation $G(x) = \alpha$ can be rewritten into the form of $ax^4 + bx^3 + cx^2 + dx + e = 0$, where

$$a = \sigma^2, \quad b = 2\mu - \sigma^2(\eta_1 - \eta_2), \quad c = -\sigma^2\eta_1\eta_2 - 2\mu(\eta_1 - \eta_2) - 2\lambda - 2\alpha,$$

$$d = -2\mu\eta_1\eta_2 - 2\lambda p(\eta_1 + \eta_2) + 2\lambda\eta_1 + 2\alpha(\eta_1 - \eta_2), \quad e = 2\alpha\eta_1\eta_2, \text{ and}$$

$$\mu = r - \delta - \frac{1}{2} \sigma^2 - \lambda \zeta. \text{ It can be shown that the four roots, } \beta_1, \beta_2, \beta_3, \text{ and } \beta_4, \text{ can be obtained by}$$

the following method which uses a combination of the Ferrari-Cardano derivation of the quartic equation and the Euler method of solving the cubic equation. This method first divides the equation by the leading coefficient to obtain a coefficient of 1 for the quartic term. Next, the quartic equation is reduced by removing the cubic term by applying the Tschirnhaus transformation and then reduces the equation once more to obtain a quadratic equation. From this point, the roots are solved and then substituted back to obtain the four roots of the original quartic equation. In a more formal manner, the process is as follows:

Solving Quartic Equations

Given $x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 = 0$ (after the leading coefficient has been divided), if $a_0=0$, then the quartic can be factored into

$$x(x^3 + a_3x^2 + a_2x + a_1) \tag{A1}$$

and the roots are then 0 and the roots of the cubic function. However, a_0 is not equal to zero, the

first step is to apply the Tschirnhaus transformation $x \mapsto y - \frac{a_3}{4}$ which yields

$$y^4 + py^2 + qy + r = 0 \tag{A2}$$

where

$$p = a_2 - \frac{3a_3^2}{8}, \quad q = a_1 - \frac{a_3a_2}{2} + \frac{a_3^3}{8}, \quad \text{and} \quad r = a_0 - \frac{a_3a_1}{4} + \frac{a_3^2a_2}{16} - \frac{3a_3^4}{256}.$$

At this point, there is a special case that greatly simplifies the process. If $r = 0$, then there is no absolute term and the equation can be factored into

$$y(y^3 + py + q). \quad (\text{A3})$$

The roots of the quartic are then $x = a_3/4$ and the roots of the cubic equation $y^3 + py + q$ with $a_3/4$ subtracted from each.

However, if r is not equal to zero, equation (A2) can be solved using a method discovered by Leonhard Euler. Euler determined that by finding the three roots of the related cubic equation

$$z^3 + \frac{p}{2}z^2 + \left(\frac{p^2 - 4r}{16}\right)z - \frac{q^2}{64} = 0$$

and setting p and q equal the square roots of two of the roots (it does not matter which two roots are chosen and the sign of the roots does not matter either) and setting $r = -\frac{f}{8pq}$, then the three roots of the cubic equation is p^2 , q^2 , and r^2 . However, the main fact that Euler discovered was that the four roots of the original quartic equation (stated before equation A1) can be calculated by the following equations:

$$x_1 = p + q + r - \frac{a_3}{4}, \quad x_2 = p - q - r - \frac{a_3}{4},$$

$$x_3 = -p + q - r - \frac{a_3}{4}, \quad \text{and} \quad x_4 = -p - q + r - \frac{a_3}{4}.$$

Each x_i for $i = 1, 2, 3$, and 4 are equal to the roots $\beta_1, \beta_2, \beta_3$, and β_4 .