Efficient Monte Carlo Simulation of Barrier Option Prices Under the Jump-Diffusion Framework

Chace Parrish Gordon
EFFICIENT MONTE CARLO SIMULATION OF BARRIER OPTION PRICES UNDER THE
JUMP-DIFFUSION FRAMEWORK

By

CHACE PARRISH GORDON

A Thesis submitted to the
Department of Mathematics
in partial fulfillment of the requirements for graduation with
Honors in the Major

Degree Awarded:
Spring, 2016
The members of the Defense Committee approve the thesis of Chace Parrish Gordon defended on April 19, 2016.

_________________________________
Dr. Giray Ökten
Thesis Director

______________________________
Dr. Alec Kercheval
Committee Member

______________________________
Dr. Fred Huffer
Outside Committee Member
Abstract

We present two efficient Monte Carlo algorithms for simulating the price of continuously monitored down-and-out barrier options when the underlying stock price follows a jump-diffusion process. Our algorithms are based on two variance reduction methods introduced by Ross & Ghamami [18] and Joshi & Leung [7]. We use numerical results to compare the efficiencies of these new algorithms with all of the Monte Carlo algorithms introduced in the literature. In addition, we present numerical results for implementing these algorithms when drawing from randomized quasi-Monte Carlo sequences and investigate their efficiencies compared to Monte Carlo approaches.
Contents

I  Introduction to Derivatives and Options  
   1  What is a derivative? 3
   2  Options 5
   3  Applications of Derivatives 8
      3.1  Risk-reducing Capabilities 8
      3.2  Investment Tools 9
      3.3  Regulatory Arbitrage 9
   4  Epilogue 9

II  The Black-Scholes Framework 9
   5  Black-Scholes Model 10
   6  Conclusion 12

III  The Jump-Diffusion Model 12
   7  Critiques of the Log-normal Model 12
   8  Merton’s Jump-Diffusion Model 13
   9  Monte Carlo and Randomized Quasi-Monte Carlo Methods 16

IV  Review of Related Publications 17

V  New Algorithms 25
   10 Numerical Results 27
      10.1  Monte Carlo Comparison 28
      10.2  Monte Carlo vs. Randomized Quasi-Monte Carlo Comparison 30

VI  Conclusion 31
Part I
Introduction to Derivatives and Options

Despite the numerous negative connotations associated with the trading of derivative securities, their introduction into exchanges worldwide has revolutionized today’s financial markets and allowed them to flourish into the prosperous markets that we know today. From their risk-reducing capabilities to their overall versatility as investment vehicles, derivatives serve as a practical instrument for many applications and problems in the financial world.

1 What is a derivative?

A derivative is a financial agreement or contract that derives its value – hence the name “derivative” – from the value of another underlying variable or asset, such as a stock or stock index [11].

Example 1.1

Suppose you work for Saudi Aramco, the largest oil producer in the world, and you want to have some form of guarantee that in one year, regardless of the future state of the oil market, you will be able to sell one million gallons of oil for the price of $100 per barrel. Derivative contracts serve their purpose precisely in these types of financial quandaries. One solution would be to enter into a forward contract, a type of derivative contract, with a willing buyer of oil that guarantees in one year you can sell one million gallons of oil at an agreed upon price – referred to as the forward price – of $100 per barrel. In the financial world, you are said to have the short position in the contract or commonly known as writing the contract, while the buyer of the contract has the long position. The specifics of these positions will be illustrated further in the next section. Since a forward is a “fair” contract in that neither party has a positive expected return on the agreement, there does not exist an initial cost to either party contractually tied to this agreement, which contrasts with a vast majority of derivatives on exchanges today.
As shown above, Figure 1 displays the payoff possibilities at expiry, one year from now, of the forward contract. Let’s suppose that the oil price decreases below its forward price of $100. What will be your resulting payoff? Since you have been guaranteed the right to sell your oil at $100, you possess the opportunity to sell your oil at a price higher than the “true” market price, resulting in a positive profit given by the difference between the forward price and the current stock price, which is referred to as the spot price. On the contrary, if the oil price instead increases beyond $100, then you are legally obligated to sell your oil at a price lower than the market price, which disparately results in a negative profit given by the difference between the forward and spot price. Thus, the payoff at expiry is:

\[
\text{Forward Contract Payoff} = F - S(T)
\]

where \( F \) denotes the forward price and \( S(T) \) denotes the oil price at time \( T \).

**Note:** As you may be able to infer at this point, the payoff structure of holding a long or short position in the forward contract closely resembles that of the payoff of owning the oil outright or short-selling the oil respectively. As a result, one might question the purpose of such a derivative. Oftentimes, institutional and retail investors desire the payoff possibilities of owning an asset without the physical ownership of the asset – possibly due to tax advantages – and the additional expenses arising from storing the asset. Hence, a forward contract or any number of the other derivatives serve as ideal solutions.
Regardless of the oil price at the option’s expiration, both you and the purchaser of oil are obligated to transact at the predetermined forward price. This forward contract protects you against the downside risk of the price dropping below $100, but also limits the profits to be made if the price instead increases beyond the forward price. However, there is another type of derivative instrument that provides protection against downside risks without limiting profit potential: the option.

2 Options

As we initially saw with forward contracts, the short and long position holders of the contract are obligated to abide by the guidelines of the contract (i.e. the short position holder is obligated to sell the barrels of oil at the forward price regardless if it will result in a disadvantageous transaction). However, this type of contract stems a few limitations in regards to its profit potential and versatility of payoff structure. As we discussed in our prior example, we want to construct a derivative security that protects us against risk, while not hindering our profitability. One viable solution is called an option.

Options are financial derivatives that give similar payoffs to forwards, but also grant protection against loss. The two most basic options are the call option and the put option. A call option is a contract that gives the buyer the option, but not the obligation to buy the underlying asset. On the other hand, a put option is a contract that gives the buyer the option, but not the obligation to sell the underlying asset [11]. Although options oftentimes have stock prices as their underlying assets, which are referred to as stock options, options can have any number of different underlying variables, such as options on the volatility of a stock’s price or the event of a terrorist attack occurring. In this paper, we will focus solely on stock options, specifically a type of exotic option referred to as a barrier option. At first glance, the concept of an option might seem abstract. Example 2.1 will illustrate the payoff possibilities of a call option on a stock.

Example 2.1

Suppose that you are an investor that speculates the price of Coca Cola’s stock, which we will abbreviate to KO for simplicity, to increase significantly in the next year. You have two choices. You could either purchase the stock outright and hold it until next year or you could buy a call option with an expiration of one year. As a risk-averse investor, purchasing the stock outright may not be the ideal solution because that results in a lack of protection against the risk of the stock price falling. Instead, you purchase a call option on KO, which gives you the option to not purchase the stock in the future if the stock price falls beyond a certain threshold, giving you a form of downside insurance. Before we delve further into the specifics of this example, let us review some of the basic terminology associated with option contracts [11].
Expiration The expiration dictates the length of an option, or the date on which the option can be exercised – denoted by $T$ in this paper.

Exercise Price The exercise price, or commonly called the strike price, is the price for which the buyer or seller – depending on the type of option – pays or receives for the underlying asset upon exercising the option – denoted by $K$ in this paper.

Underlying Asset The underlying asset is the asset for which the option depends on. In our forward contract example, the underlying asset was oil.

Exercise Style The exercise style specifies when the option can be exercised. The three main types are American, Bermudan, and European. An American style option can be exercised at any point during the life of an option. A European style option can only be exercised at the expiration of the option. If the option can only be exercised at specific time periods during the duration of the option, the option is called a Bermudan style option.

Short/Long Position One is said to have the long position in the option if the person buys the option. On the other hand, someone has the short position in the option if they write the option. These positions should not be confused with holding short/long positions in the underlying asset. A short position in the underlying asset refers to the position that has profitable outcomes associated with a decline in the value of the asset. On the contrary, a long position in the underlying assets becomes favorable as a result of an increase in asset value.

Now that we have discussed the basic terminology of options, let us discuss the details of the call option contract in our aforementioned example. Recall that you foresee the price of $KO$ to increase and as a result, you want to purchase a call option that will exploit this speculation. Suppose the current price of $KO$ is $100 and you purchase a European call option on $KO$ with an exercise price of $100 and an expiration of one year. This allows you the option to buy $KO$ at the expiration price of $100 in one year. The payoff at expiry is:

\[
\text{Call Option Payoff} = \max(S(T) - K, 0)
\]

where $K$ denotes the strike price. Let us walk through the three distinct cases.

**Note:** Figure 2 displays the potential payoffs of our call option, while Figure 3 displays its corresponding put option with the equivalent exercise price.

**Case 1:** $S(1) > 100$

Since we speculated that the price of the stock will increase from its original price of $100 and we based our investment strategy around this assumption, an increase in the stock price puts us in the most lucrative position. As a result of a stock price greater than $100, we can exercise our position and in turn,
immediately sell the stock back on the market for a payoff of \((S(1) - 100)\), which is strictly positive in this case. In Figure 2, the upward sloping portion of the payoff diagram displays the possible outcomes resulting from a stock price greater than $100.

**Case 2:** \(S(1) = 100\)

The case of equality between the future stock price and the strike price does not hold much interest. In a frictionless market, a market existing without transaction costs, the long position holder of the option will be indifferent between exercising or not exercising due to the fact that both decisions will result in zero profit.

**Case 3:** \(S(1) < 100\)

In the case of a decrease in price, the “insurance” portion of our option takes over. Since we do not have the obligation to exercise our option at this point and exercising would result in a negative profit, we would decide not to exercise the option, resulting in a payoff of zero. Contrary to the payoff diagram of the forward contract, our payoff while the current stock price is below $100 remains zero because of our option to exercise instead of an obligation as in the corresponding forward contract.

**Note:** In our discussions of potential payoffs, we are neglecting interest rate and initial cost assumptions.

**Figure 2: European Call Option Payoff (Long Position), with \(K= 100\).**
3 Applications of Derivatives

Throughout various industries, derivative options, especially stock options, have copious uses and applications. We will now discuss some of the most popular uses for derivative securities [11].

3.1 Risk-reducing Capabilities

As with our first example, derivatives can be used to reduce the risks associated with different types of business. The forward contract example is an exceptional example of the risk-reducing nature of derivatives. It involves two parties that entered into the contract in search of hedging the risk associated with the oil business. While the buyer of oil wants to mitigate the risk of the oil price rising, the producer of oil desires to reduce the risk associated with the price falling. Options can also be used for insurance purposes as well. In essence, buying homeowner’s insurance is equivalent to purchasing a put option. When the value of the policyholder’s house declines, the policyholder is given insurance payments, which are analogous to that of the payoff given to the long position of a put option as a result of the underlying asset price – the house – declining. Due to this striking similitude, every form of insurance can be thought of as some type of derivative.
3.2 Investment Tools

Although derivatives are often used for their powerful hedging capabilities, they can also be utilized in a speculative manner. Derivative securities can be thought of as “bets” on the future value of the underlying asset. As with our call option example, the investor forecasted that the stock price will increase in the future and bet on this speculation in hopes of a lucrative gain. The call option had a positive payoff if the stock price rose and had risk protection if the stock price fell.

In order to make this a fair contract, compensation for the writer of this option comes in the form of a derivative premium, which is the price that the long position holder must pay for the contract. We will be discussing the problem of derivative pricing in subsequent sections.

3.3 Regulatory Arbitrage

Due to innumerable regulatory restrictions and taxes imposed on financial markets, investors and companies are always searching for potential ways to evade them. As a result, the versatility of derivatives offers an innovative solution to bypassing these regulations. Companies often invoke the assistance of financial engineers to design derivatives in such a way that exploits the loopholes that exist within the regulatory laws.

4 Epilogue

Over the past few decades, derivative securities have had a significant impact on the way in which trading has taken place and how trading strategies have been implemented and structured. However, some important questions remain that plague financial engineers worldwide. How do you properly price different derivative securities so that the transaction is “fair” for each party? Do analytically tractable solutions exist for the price of a derivative? If not, how do you estimate such a price and what are some of the other problems associated with the pricing of derivatives? In the next few chapters, we will discuss the answers to these questions and some of the research regarding the efficient pricing of barrier options – a complex, yet frequently traded derivative.

Part II

The Black-Scholes Framework

Historically, the valuation of derivative securities – a complex, yet popular variety of financial instruments – has remained the subject of a considerable amount of research and controversy. When tasked with the responsibility of valuing a derivative, academics and financial engineers initially posited that the security
price depended on an array of subjective variables, such as risk aversion and utility, which consequently relegated the possibility of deriving an analytical or unifying pricing formula to an impossible task.

However, in their breakthrough paper regarding the pricing of corporate securities, Black and Scholes [1] presented a modeling framework which circumvented unobservable variables. Their derivative pricing formula depended solely on variables that can be easily obtained from market data – asset volatility, short-term interest rates, current asset price. Their discovery impacted not only the academic landscape, but it also catalyzed the rapid growth of option trading within global option markets in the second half of the twentieth century. It provided financial institutions with the ability to properly price derivatives, specifically European call and put options, in an accurate and efficient manner.

5 Black-Scholes Model

In their derivation of the option pricing formula, Black and Scholes [1] assumed ‘ideal conditions’ in the market for the stock and option. The conditions are as follows:

1. The market is frictionless, or in other words, transaction costs, taxes, and any other expenses are non-existent.

2. The short-term, risk-free interest rate is known and constant.

3. The market allows for uninhibited, unregulated short-selling and borrowing of securities.

4. Traders can buy and sell fractional amounts of securities.

5. Equities do not pay dividends.

6. The stock price follows a geometric Brownian motion in which its drift and volatility are known and constant – analogously, stock prices are lognormally distributed.

7. The option can only be exercised at the expiration date. Hence, they are European options.

The lognormality assumption of Condition 6 assumes that stock returns are continuously compounded and normally distributed. If we let

- $\mu$: Annualized expected stock return
- $\sigma$: Annualized volatility of stock returns
- $\Delta t = u - v$: Time horizon in years (assuming $u > v$)
- $Z(t)$: Standard Brownian motion

Then, the geometric Brownian motion of the stock price process is defined through the stochastic differential equation as follows:

$$\frac{dS(t)}{S(t)} = \mu dt + \sigma dZ(t)$$
By solving this SDE, Glasserman [5] showed that

\[ S(u) = S(v)e^{(\mu - \frac{\sigma^2}{2})\Delta t + \sigma Z(\Delta t)} \]

and

\[ \ln \left( \frac{S(u)}{S(v)} \right) \sim \text{Normal}\left((\mu - \frac{\sigma^2}{2})\Delta t, \sigma^2 \Delta t\right) \]

Under these conditions, Black and Scholes [1] derived a partial differential equation – known as the Black-Scholes Equation – which governs the option price process over time.

\[
rS(t) \frac{\partial V(S(t), t)}{\partial S(t)} + \frac{1}{2} \sigma^2 S(t)^2 \frac{\partial^2 V(S(t), t)}{\partial S(t)^2} + \frac{\partial V(S(t), t)}{\partial t} = rV(S(t), t)
\]

where

- \( S(t) \): stock price at time \( t \)
- \( K \): strike price
- \( V(S(t), t) \): value of the derivative as a function of time and the stock price
- \( t \): time (in years)
- \( \sigma \): volatility, or standard deviation, of the stock’s returns
- \( r \): risk-free, short-term interest rate (i.e. 3-month U.S. treasury bill rate)
- \( \mu \): annualized drift rate of \( S(t) \)

Upon solving the Black-Scholes equation with a European option’s corresponding terminal and boundary conditions, Black and Scholes [1] derived the Black-Scholes formula, which calculates the price of call and put options. Within this market framework, the value of an option depends solely on variables that can be obtained from readily available market data, such as asset price volatility or the risk-free interest rate. Although only the Black-Scholes formula for a non-dividend paying stock is listed below, further publications have extended this framework to include for dynamic interest rates, dividend pay outs, transaction costs, and various other extensions [11]. While the Black-Scholes formula below is given exclusively for put options, the corresponding call option can be easily calculated with the put-call parity [11].

Let

\[
d_1 = \frac{\ln\left( \frac{S(t)}{K} \right) + (r + \frac{\sigma^2}{2})(T - t)}{\sigma \sqrt{T - t}}
\]

\[
d_2 = d_1 - \sigma \sqrt{T - t}
\]
Then,

\[ V(S(t), t) = N(-d_2)Ke^{-r(T-t)} - N(-d_1)S(t) \]

where \( N(x) \) is the cumulative distribution function for a standard normal distribution.

6 Conclusion

Although the Black-Scholes formula has been vigorously used throughout the last half century and presents a reasonably accurate asymptotic approximation to the actual price of an option, the original Black-Scholes framework presents two main limitations. As we will see in Part III, empirical data suggests that this framework is inadequate to describe financial markets. We will discuss these shortcomings and a solution presented by Robert Merton [12], one of the main contributors to the original Black-Scholes model.

Part III
The Jump-Diffusion Model

Despite its widespread ramifications and incontrovertible utility, the Black-Scholes model and its underlying assumptions contain two main shortcomings that prevent the model from being an adequate representation of financial markets. In this section, we will discuss these limitations and then examine a generalization of the original Black-Scholes model that is aimed at mitigating this inadequacy.

7 Critiques of the Log-normal Model

After an examination of historical stock returns and normally distributed stock returns of the Black-Scholes model, it can be seen that actual stock returns follow a leptokurtic distribution, which is characterized by excess kurtosis and skewness compared to its Gaussian distribution counterpart. As illustrated in Figure 4, the leptokurtic property implies that large variations in stock returns are more likely to occur than in the original constant-variance, lognormal model.
There are two main generalizations of the Black-Scholes model. One of them removes the constant volatility assumption and models the volatility via a deterministic function or stochastic process [4]. This is supported by the empirical observation that the volatility tends to vary with respect to the strike price and expiry – commonly known as the 'volatility smile' [11].

The Black-Scholes framework implies that asset prices follow a continuous path. However, empirical studies of stock price data have supported the existence of jumps in stock prices, going back to the study of Fama [3]. For example, on October 28, 2008, Volkswagen experienced one of the largest short squeezes of all time. In response to Porsche’s sudden and unexpected takeover of Volkswagen, their stock price shot up severely within a day’s time. At the stock’s highest point during the day, it was up 93% from its beginning of the day price [8]. Under the framework of a pure diffusion process, this dramatic increase is virtually impossible, assuming a reasonably bounded volatility. Hence, a modification of the original model must be provided to account for these large, relatively instantaneous jumps, which are consequences of unexpected arrivals of important information about a specific firm or industry, such as a better-than-expected company earnings report or surprising dividend announcement. A few years after the original Black-Scholes publication, Merton compensated for this major limitation by introducing a generalization, the Jump-Diffusion Model [12].

8 Merton’s Jump-Diffusion Model

Merton [12] introduced a generalization to the original lognormal model to allow for jumps at finitely many points. His so-called “jump-diffusion model”
decomposed the stochastic process into two components: the ‘normal’ continuous fluctuations and the ‘abnormal’ jumps that occur at discrete points. The continuous component results from the day-to-day changes in market supply and demand and is described by a lognormal distribution similar to that of the Black-Scholes model. In contrast, the discontinuous element is a consequence of sudden, unexpected information that comes to the market as in our previous example. These unexpected events are modeled by a Poisson process. Consequently, the jump-diffusion model accounts for the additional volatility precipitated by the uncertainty of the jumps.

In his original paper, Merton described the jump process as a Poisson-driven process, which dictates the number of jumps in a given time horizon, with its corresponding arrival times having an exponential distribution. For reference purposes, the corresponding probability mass function of a Poisson-process and the probability density function of an exponential distribution are listed below [11].

Let $X$ be a discrete random variable with a Poisson distribution with expected value $\lambda$ and $Y$ be a continuous random variable with an exponential distribution with expected value $\frac{1}{\lambda}$. Thus, the probability mass function of $X$ is given by

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

and the probability density function of $Y$ is given by:

$$f(y; \lambda) = \lambda e^{-\lambda y}$$

for non-zero $Y$ and zero elsewhere.

The dynamics of the stock price under the mixture of the jump and diffusion processes can be described by the stochastic differential equation [12]:

$$\frac{dS(t)}{S(t)} = (\mu - \lambda k)dt + \sigma dZ(t) + dq(t)$$

or rewritten as

$$\frac{dS(t)}{S(t)} = (\mu - \lambda k)dt + \sigma dZ(t) + (J - 1)q(t)$$

where $q(t) = 1$ if a jump occurs at time $t$ and zero otherwise.

The instantaneous expected return on the stock and standard Weiner processes are defined as $\mu$ and $dZ(t)$, respectively.

Similar to our pure diffusion equation, the $dt$ portion describes the non-random component of the process resulting from the asset’s drift, while the $dZ$ component explains the random price vibrations as a consequence of the asset’s riskiness. As a result of the addition of the jump process, the $dq$ term describes the change resulting from the discrete jumps that occur (no more than one jump can occur at any instant of time). The $(J - 1)q(t)$ term explains the magnitude of
a jump. Due to the fact that the $dq$ term is only non-zero in the case of a jump, one can intuitively see that this process generalizes the pure diffusion process described in Black-Scholes analysis. If no jumps occur, the $dq$ remains zero and the jump-diffusion process reduces to the original Black-Scholes stochastic differential equation. However, suppose a jump occurs at some arbitrary time $t$ with a corresponding jump size of $Y$. As a consequence, the stock price has an instantaneous finite jump from $S(t)$ to $S(t)Y$. Therefore, the stock price will follow a continuous path except at a finite number of points in a given time interval. If we let $k$ be some constant and $J(n)$ be a product of $n$ i.i.d. jumps $\{J_i\}_{i=1}^n$, then the stock price can be written as follows,

$$S(t) = S(0)e^{(r - \frac{\sigma^2}{2} - \lambda k)t + \sigma Z(t)}J(n)$$

To account for the additional uncertainty in the risk-neutral measure, Glasserman [5] shows that you must correct the drift by the $-\lambda k$ term, where $k = E[J] - 1$ or, in other words, it is the percentage change of the stock price instantaneously after the jump.

Unfortunately, as one might expect, some problems arise with the addition of jumps into our model. As in the no-arbitrage method for pricing options, one must create a portfolio process that exactly replicates the payoff of the option. However, the delta hedging strategy that worked for mitigating the risks associated with small fluctuations in the stock price might not protect against large stock price movements corresponding to jumps. Consequently, risk-neutral pricing might be difficult or even impossible to implement under the jump-diffusion model. For analytical tractability reasons, Merton [12] bypassed this difficulty and assumed that the jump sizes $\{J_i\}$ are lognormally distributed. As a result, the stock price process conditional on a specific number of jumps also follows a lognormal distribution. Hence, Merton [12] was able to derive a closed-form solution for a European call option in the form of an infinite series. Utilizing the law of total expectation and the lognormality of the conditional stock price process, Merton employed the Black-Scholes formula to express the option pricing formula as follows:

$$V(S(t), t) = \sum_{k=1}^{\infty} P(X = k)E_{B-S}[V(S(0), 0)|N(T) = k]$$

where $P(X = x)$ is the probability mass function for the underlying Poisson-process and $N(T)$ is the number of jumps up until time $T$.

Due to computational limitations, financial engineers often utilize this formula by approximating an option price with the series’ $n$th partial sum where $n$ is determined according to the engineers’ accuracy preference.

Unfortunately, this tractable formula only applies when calculating the price of a plain vanilla option under the assumption of lognormal jump sizes. When computing the value of more exotic options, such as barrier options, or assuming different jump size distributions, analytical solutions are fairly difficult, if not impossible, to derive. Hence, to price these types of derivatives, one must usually resort to numerical methods – in particular, Monte Carlo methods.
9 Monte Carlo and Randomized Quasi-Monte Carlo Methods

Monte Carlo methods are a broad subclass of numerical methods that use random sampling to approximate solutions to quantitative problems. Many real-world or mathematical problems can be difficult or impossible to solve analytically. In such problems, Monte Carlo methods may be the only viable option. A typical Monte Carlo algorithm involves generating inputs from a specific probability distribution, performing a deterministic computation on the inputs, and then averaging over the results to obtain an approximate solution [5].

For example, if we wanted to estimate the integral $I$ of some function $f$,

$$I = \int_{0}^{1} f(x) dx$$

we could represent this integral as an expectation $E[f(U)]$, where $U$ is a uniform random variable distributed over the unit interval. By generating $\{U_i\}_{i=1}^{N}$, our Monte Carlo estimate for $I$ is as follows [5]:

$$\hat{I} = \frac{1}{N} \sum_{i=1}^{N} f(U_i)$$

In computational finance, Monte Carlo algorithms are often employed to simulate a financial instrument’s price. For example, the price of a European call stock option depends solely on the stock price at the expiration date – assuming the strike price and risk-free rate are constant. Therefore, a common Monte Carlo valuation would involve simulating numerous ending stock prices, calculating the respective option values, and finally, taking an arithmetic average of all of the simulated option values. According to the law of large numbers and if no bias is present, as the number of individual simulations increases, the estimated option price will converge to its true value. Since this approach may involve a sizable number of simulations to satisfy one’s desired accuracy, constructing an efficient algorithm is imperative. But what is meant by efficiency? Computational efficiency can be divided into two components: computational time and accuracy. Mathematically, we will define this as:

$$\text{Efficiency} = \text{Accuracy} \times \text{Computational Time}$$

Accuracy is measured by the sample standard deviation of the estimates. Computational time and accuracy are generally inversely proportional. When one increases the number of simulations, the accuracy of the estimation improves; however, increasing the computations will increase the computational time. In the context of high-frequency trading, the difference between computational speeds of merely milliseconds can be the difference between millions gained or lost. Thus, constructing an algorithm that balances this trade-off is...
one of the most important considerations in valuing a derivative. Since we want
to minimize both the variance and computational time of our estimates, we
want to choose an algorithm that minimizes our defined measure of efficiency
or, equivalently, maximizes its reciprocal.

In subsequent sections, I will use Monte Carlo methods in order to price bar-
errier options under the jump-diffusion framework. I will apply various variance
reduction techniques, such as the control variate method, to improve both the
variance and computational time of the simulation. Additionally, I will compare
the Monte Carlo random number sampling approach to a quasi-Monte Carlo im-
plementation (QMC). In Monte Carlo simulations, random number samples are
generated using pseudorandom numbers which are drawn to mimic randomness
[5]. In QMC, samples are generated using low-discrepancy sequences in which
points are deterministically chose in order to increase accuracy by more evenly
distributing the sequence. As a result, QMC methods have the potential to
accelerate the convergence from $O(\frac{1}{\sqrt{n}})$ in MC to nearly $O(\frac{1}{n})$ under certain
dimensional conditions ($n$ is the number of points generated). Although QMC
methods often reduce simulation variances, they are also frequently more com-
putationally demanding than MC methods, thus we must quantify this trade-off
with our measure of efficiency.

In our QMC simulations, we will be drawing samples from Halton sequences,
one of the most used low-discrepancy sequences. However, there exist two main
drawbacks of using Halton sequences in option pricing. First, there is no practi-
cal way to access the accuracy of the estimates. The QMC method only provides
us with a single estimate; thus, it is impossible to generate multiple option price
estimates to quantify the standard deviation of multiple estimates and measure
the estimate error. Many researchers have addressed this problem by introduc-
ing randomized quasi-Monte Carlo methods, thereby allowing us to generate
multiple price estimates so that we can calculate the standard deviation of our
simulations [15].

The second problem with using Halton sequences is a dimensional issue.
It is well known that the variance reduction advantages of using Halton se-
quences diminishes as the dimension of the problem increases, where certain
components of the sequences exhibit poor uniformity. To overcome this prob-
lem, permuted Halton sequences were introduced. An efficient algorithm for
generating random-start, permuted Halton sequences was given by Ökten and
Xu [21], thereby overcoming both of these QMC problems.

Part IV

Review of Related Publications

There have been three main publications that offer unbiased algorithms for
the valuation of barrier options under the jump-diffusion framework. I will
focus on comparing the efficiency of my proposed algorithms with respect to
the algorithms of these three previous papers.

**Barrier Options**

Barrier options, one of the most popular exotic derivatives, have a payoff at expiry similar to that of a European option; however, the payoff is also dependent on whether the underlying asset price crosses a pre-defined barrier level. If it crosses the barrier price, the option could either be knocked-in or knocked-out. If it is a knock-in option, then it can only be exercised if the barrier has been crossed. In contrast, a knock-out option is only valid if the barrier is never crossed during the life of the option. Due to the price parity relationship between knock-in and knock-in options, given by the equation below, it is only necessary to study one type [11].

\[
\text{European Call} = \text{Knock-in Call} + \text{Knock-out Call}
\]

**Note:** The same relationship in Equation 1 holds for put options as well.

In this paper, we will consider down-and-out call options. If at any time during the life of the option the underlying asset’s price drops below the barrier level, then the option is knocked-out and its value immediately goes to zero. If the barrier is not crossed, the payoff at expiration \(T\) is the same as a vanilla call option. The barrier can be either monitored continuously or discretely. In the real-world, barrier options are monitored discretely – usually at the end of the roughly 250 trading days per annum. However, for tractability reasons, most algorithms and pricing formulas offered in recent publications consider the continuous case.

Hence, if we let \(B\) denote the barrier price and \(T\) be the expiration date for our down-and-out call option \(DOC\), then

\[
DOC = e^{-rT}E[\max(S(T) - K, 0)1(\tau_B > T)]
\]

where \(1(.)\) is the indicator function of the event in the braces and \(\tau_B = \inf\{t > 0 : S(t) \leq B\}\) [16].

The valuation of these options requires a solution to the first-passage time problem, or boundary-crossing problem, which has been studied for the last fifty years [19]. Within a standard lognormal framework, closed-formed solutions have been derived for the valuation of barrier options [14]. However, closed-formed expressions for a jump-diffusion framework are rather limited. Kou and Wang [9] derive an expression for the first-passage problem when jump-sizes come from a double-exponential distribution. Arising from this difficulty in obtaining tractable solutions, Monte Carlo and other numerical methods are often employed. In our case, we will focus solely on applying Monte Carlo simulations.
Algorithm A: Standard MC Approach

In a standard MC approach, we would generate $N$ price paths according to a jump-diffusion process and then compute the expected discounted payoff of the option by averaging over the $N$ estimates. This calculation is encapsulated by the equation below.

$$\hat{DOC} = e^{-rT} \sum_{i=1}^{N} \max(S_i(T) - K, 0)1(\tau_{i,B} > T)$$

where $\hat{DOC}$ is the estimator for our down-and-out call option and $\tau_{i,B}$ denotes our first-passage crossing time of the $i$th estimate.

However, since we are only considering the case of a barrier option with a continuously monitored barrier, our pricing estimate will have an inherent discretization bias. Our standard MC estimates will be greater than any unbiased estimates since the discretely-monitored barrier will not return as many zero estimates as a continuously-monitored barrier would. In addition to the discretization bias, computational times increase rather rapidly as our time discretization $\Delta t$ goes to zero.

Algorithm B: First Unbiased Estimator Using Brownian Bridge

The first unbiased Monte Carlo approach to simulate continuously monitored barrier option prices was given by Metwally and Atiya [13]. The authors introduced a MC algorithm which utilized the Brownian bridge concept, removing the time discretization bias inherent in most traditional MC simulations of option prices. In a standard MC approach, the barrier is discretely monitored; however, the introduction of the Brownian bridge concept made it possible to obtain the boundary-crossing probability density, given the two end-points. This, in turn, allows one to monitor the barrier continuously and entirely remove discretization bias from the price estimation. Lemma 1 was provided by Metwally and Atiya [13].

Lemma 1

Suppose $\{X(t)\}$ is a Brownian motion process with drift $\mu$ and variance $\sigma^2$. Let $\omega = t_i - t_{i-1}$. If $X(t_{i-1}) > \ln(B)$ and $X(t_i) > \ln(B)$, then

$$P(\inf_{t_{i-1} \leq s \leq t_i} X(s) > \ln(B) | X(t_{i-1}) = a, X(t_i) = b) = 1 - \exp \left( -\frac{2(\ln(B) - a)(\ln(B) - b)}{\omega \sigma^2} \right)$$

Lemma 1 provides us with the probability that a Brownian motion process will remain above the barrier level $\ln(B)$ within the time interval $[t_{i-1}, t_i]$, conditional on the two end-points $X(t_{i-1})$ and $X(t_i)$. 

To Generate Estimate of $E[V(S(0), 0)]$ (Atiya & Metwally):

1. Generate jump times $\{t_i\}_{i=1}^m$ by drawing from the inter-jump time distributions (i.e. exponential distribution).

2. Set the number of jumps during the life of the option equal to $m$.

3. For $i = 1$ to $m$:
   
   (a) Generate $N$, a normal random variable with mean $\mu(t_i - t_{i-1})$ and variance $\sigma^2 \sqrt{t_i - t_{i-1}}$. Set $S_i^- = S_{i-1}^+ e^N$.
   
   (b) Simulate the jump-size $J_i$ according to the given jump-size distribution (i.e. lognormal). Set $S_i^+ = S_i^- J_i$.

4. Generate a final normal random variable $N$ with mean $\mu(T - t_m)$ and variance $\sigma^2 \sqrt{T - t_m}$. Set $S(T) = S_m^+ e^N$.

5. For $i = 1$ to $m + 1$:
   
   (a) Compute the intra-period probability of not crossing the barrier $P_i$ using Lemma 1.
   
   (b) Generate a uniform random number $U_i \in [0, 1]$. If $U_i > P_i$, return a zero estimate.
   
   (c) If $S_i^- \leq B$ or $S_i^+ \leq B$, return a zero estimate.

6. No barrier crossing has occurred during the duration of the option. Return $\max(S(T) - K, 0) e^{-rT}$.

Note: $S_i^-$ and $S_i^+$ denote the stock price at time $t$ before and after the $i$th jump, respectively.

Although this paper introduced a method for removing bias from the estimates, it did not provide any variance reduction techniques to improve the variance of the simulations.

Algorithm C: First Estimator Constructed With Variance Reduction Techniques

Joshi and Leung [7] were the first to apply variance reduction techniques to our continuous problem. One of the main inhibitors to the substantial variance of Atiya and Metwally’s implementation was that a relatively large proportion of the estimates would return a zero estimate for the price, which contributed very little to the accuracy of the estimate. Instead, Joshi and Leung used importance sampling techniques in order to remove the possibility of a zero price estimate and draw only non-zero estimates, resulting in a substantial improvement in variance while still maintaining zero bias. This is done by adjusting the probability measure such that zero estimates are not sampled, while also adjusting the expectation in order to maintain unbiasedness.
For example, suppose our original probability measure is $P$ with density $\psi$, whereas our adjusted probability measure is $Q$ with density $\Gamma$. Hence, we can rewrite our expectation below.

$$E_P[f(x)] = \int f(x)\psi(x)dx = \int f(x)\frac{\psi(x)}{\Gamma(x)}\Gamma(x)dx = E_Q\left[f(x)\frac{\psi(x)}{\Gamma(x)}\right]$$

where the weight $\frac{\psi(x)}{\Gamma(x)}$ is the likelihood ratio or Radon-Nikodym derivative evaluated at $x$ [5].

**Note:** By choosing $\Gamma(x)$ so that $\frac{\psi(x)}{\Gamma(x)}f(x)$ is constant or close to constant, the variance of the estimate will be small [7].

In Joshi and Leung’s approach, this technique is utilized when generating jump-sizes and lognormal processes for each estimate. Although this did remove the possibility of non-zero estimates and drastically reduced the variance of each simulation, this came at the expense of additional computational time since each simulation, on average, requires more random number sampling and other more demanding computations.

In addition to importance sampling, Joshi and Leung [7] decomposed their price estimate by conditioning on whether there were zero jumps or greater than zero jumps. This price is given by:

$$E[V(S(0), 0)] = E[V(S(0), 0)|N(T) = 0]Pr[t_1 > T]$$

$$+ E[V(S(0), 0)|N(T) > 0]Pr[t_1 \leq T]$$

where $t_1$ denotes the first-passage time of the barrier.

As a result, in the case of no jumps, they utilized the analytical, closed-form pricing formula to evaluate the price conditional on no jumps during the life of the option. This formula is given by Lemma 2 [7].

**Lemma 2**

Let $\nu = r - \frac{\sigma^2}{2} - \lambda k$ and $\tilde{\nu} = \nu + \sigma^2$. The price of a down-and-out call option is:

$$E[V(S(0), 0)|N(T) = 0] =$$

$$S(0)e^{-\lambda k T} \left( \Phi \left( \tilde{\nu} T + \ln \left( \frac{S(0)}{K} \right) \frac{\nu}{\sigma \sqrt{T}} \right) - \left( \frac{B}{S(0)} \right)^{2\tilde{\nu} \sigma^{-2}} \Phi \left( \ln \left( \frac{B^2}{K S(0)} \right) + \tilde{\nu} T + \frac{\nu}{\sigma \sqrt{T}} \right) \right)$$

$$- Ke^{-r T} \left( \Phi \left( \nu T + \ln \left( \frac{S(0)}{K} \right) \frac{\nu}{\sigma \sqrt{T}} \right) - \left( \frac{B}{S(0)} \right)^{2\nu \sigma^{-2}} \Phi \left( \ln \left( \frac{B^2}{K S(0)} \right) + \nu T + \frac{\nu}{\sigma \sqrt{T}} \right) \right)$$

where $N(t)$ denotes the number of jumps up to time $t$ and $\Phi(x)$ denotes the cumulative distribution function of a standard normal evaluated at $x$. 

21
To Generate Estimate of $E[V(S(0), 0) \mid N(T) > 0]$ (Joshi & Metwally):

1. Set likelihood ratio $L = 1$.

2. Generate the first jump time $t_1$ given that $t_1 < T$ using importance sampling of an exponential distribution. Set $L = L \times \Pr [t_1 < T]$.

3. Generate the remaining jump times $t_i$ from an exponential distribution. Set the number of jumps during the life of the option equal to $m$.

4. For $i = 1$ to $m$:
   a. Generate the pre-jump stock price $S_{i-}^-$ by using importance sampling to ensure that it is above the barrier.
   b. Set $L = L \times \Pr [S_{i-}^- > B | S_{i-1}^+]$.
   c. Calculate the intra-period probability of no barrier crossing $\tilde{P}$ using Lemma 1. Set $L = L \times \tilde{P}$.
   d. Generate the jump-size $J_i$ by using importance sampling to ensure that $S_{i-}^+$ is above the barrier and set $S_{i-}^+ = S_{i-}^- J_i$.
   e. Set $L = L \times \Pr [S_{i-}^+ > B | S_{i-}^-]$. 

5. Calculate the option value $V(S(t_m), t_m)$ at time $t_m$ using the analytical option pricing formula. Return $V(S(t_m), t_m)Le^{-r(t_m)}$.

Algorithm D: Application of Stratified Sampling and Conditional Expectation

Ross and Ghamami [18] were the next to apply variance reduction techniques. Their variance reduction consists of two parts: stratified sampling and conditional expectation. These variance reduction techniques not only significantly improved the estimation error from Atiya and Metwally’s implementation [13], but they were also not as computationally demanding as Joshi and Leung’s estimator [7].

First, Ross and Ghamami [18] employed a stratified sampling approach, allowing them to break up the estimations by stratifying on the number of jumps incurred during the life of the option. This permitted them to utilize the properties of conditional expectations and the well-known barrier option formula for the case of no jumps.

$$DOC = E[V(S(0), 0)] = \sum_{m=0}^{q} E[V(0) \mid N(T) = m]P_m + E[V(0) \mid N(T) > q]\tilde{P}_q$$

where

$$P_m = P(N(T) = m) = \frac{(\lambda T)^m e^{-\lambda T}}{m!}$$

$$\tilde{P}_q = P(N(T) > q)$$
Note: In order to minimize variance, \( q \) should be chosen so that \( \bar{P}_q \) is small. Ross and Ghamami [18] use \( q \geq \lambda T + 3\sqrt{\lambda T} \). Also, we choose the number of simulations to perform for each conditional expectation based on a proportional stratified sampling procedure, which is done by choosing the number of simulations proportional to the probability of being in that stratum. This has been shown to guarantee variance reduction [17]. Hence, the number of simulations for the conditional expectation given \( m \) jumps is \( n_m = \frac{NP}{1-P_0} \) and \( n_{q+1} = N - \sum_{i=1}^{q} n_i \) where \( N \) is the total number of Monte Carlo simulations.

Hence, to estimate \( E[V(S(0), 0)] \), we must generate estimates for three different cases.

**Case 1 (To Generate Estimate of \( E[V(S(0), 0)|N(T) = 0] \)):**

Ross and Ghamami [18] employed the analytical formula similar to how Joshi and Leung did for the case of no jumps. Consequently, as \( P_0 \) increases, the variance decreases as well since more of the simulation is dependent on a closed-form expression.

1. Compute \( E[V(S(0), 0)|N(T) = 0] \) by using Lemma 2.

**Case 2 (To Generate Estimate of \( E[V(S(0), 0)|N(T) = m] \)):**

Let \( E_m \) denote the conditional expectation given \( m \) jumps. To estimate \( E_m[V] \), we must first generate the jump-sizes \( \{J_i\}_{i=1}^{m} \) for \( m \) jumps and let \( J = \prod_{i=1}^{m} J_i \).

Since there will only be a non-zero estimate when \( B < K < S(T) \) and \( S(T) = S(0)e^N \) (\( N \) is a normal random variable with mean \( \mu T \) and variance \( \sigma^2 T \)), our conditional expectation simplifies to:

\[
E_m[V|J] = E_m[V|K < S(T), J] Pr[K < S(T)]
\]

\[
= E_m[V|K < S(T), J] Pr[K < S(0)e^N]
\]

\[
= E_m[V|K < S(T), J] Pr[\ln(K/S(0)J) < N]
\]

\[
= E_m[V|K < S(T), J] \left[ 1 - \Phi\left( \frac{\ln(K/S(0)J) - \mu T}{\sigma \sqrt{T}} \right) \right]
\]

Hence, to estimate \( E_m[V|J] \), we generate \( m \) jump-times \( \{t_i\}_{i=1}^{m} \) by sampling \( m \) uniform numbers from \( (0, T) \) and then order them such that \( t_j > t_i \) for all \( j > i \). Then, we simulate \( N \) and set \( S(T) = S(0)e^N \). Let \( \{X_i\}_{i=1}^{m+1} \) be i.i.d. normal random variables with mean \( \mu(t_i - t_{i-1}) \) and variance \( \sigma^2(t_i - t_{i-1}) \). Then,
Thus, conditional on this terminal stock price $S(T)$ and $N = X_1 + ... + X_{m+1}$, we now must sequentially generate $\{X_i\}$. We first generate $X_1$ given $N$, then $X_2$ given $X_1$ and $N$, and so on. This calculation can be done using Lemma 3.

Note: In Ross and Ghamami’s publication [18], erroneous formulas were given in their Lemma for calculating the conditional distribution of jointly normal random variables; thus, we derive the correct expressions below.

**Lemma 3**

Suppose $\{X_i\}_{i=1}^n$ are independently distributed normal random variables with means $\mu_i$ and variances $\sigma_i^2$, respectively. Let $Q = \sum_{j=1}^n X_j$. If $\tilde{X}_i$ has the distribution of $X_i$ given $Q = q$, $X_1 = x_1, ..., X_{i-1} = x_{i-1} \ \forall i \in [1, 2, 3, ..., n]$, then $\tilde{X}_i$ is a normal random variable with mean:

$$
\mu_i + \frac{\sigma_i^2 (q - \sum_{j=1}^{i-1} x_j - \sum_{j=i}^n \mu_j)}{\sum_{j=i}^n \sigma_j^2},
$$

and variance

$$
\sigma_i^2 (1 - \frac{\sigma_i^2}{\sum_{j=i}^n \sigma_j^2}).
$$

**Proof** The conditional distribution of $X_i$ given $X_1 = x_1, ..., X_{i-1} = x_{i-1}, Q = q$ is equivalent to the conditional distribution of $X_i$ given $\sum_{j=1}^n X_j = q - \sum_{j=1}^{i-1} X_j$. Consequently, the joint distribution of $X_i$ and $\sum_{j=1}^n X_j$ is a bivariate normal with a correlation of $\frac{\sigma_i}{\sqrt{\sum_{j=i}^n \sigma_j^2}}$. By the properties of jointly normal random variables [20], we know that, given some jointly normal random variables $U$ and $V$, $E[U|V] = E[U] + \rho_{U,V} \frac{\sigma_U}{\sigma_V} (V - E[V])$ and $Var[U|V] = \sigma_U^2 (1 - \rho_{U,V}^2)$. Hence, Lemma 3’s result follows.

Using this iterative process, we can generate an estimate of $E_m[V]$ doing the following:

1. Generate jump-sizes $\{J_i\}_{i=1}^m$ from given jump-size distribution (i.e. log-normal). Set $J = \prod_i J_i$ for $i = 1, 2, 3, ..., m$

2. Generate $m$ uniform random numbers from $(0, T)$. Set them equal to $\{t_i\}_{i=1}^m$ where $t_p > t_q$ for all integers $m > p > q > 0$.

3. Simulate the terminal stock price $S(T)$ conditional on it being greater than $K$. To achieve this, generate $N$, a normal random variable with mean $\mu T$ and variance $\sigma^2 T$ conditional on $N$ being greater than $\ln(\frac{K}{S(0)J})$. Set $S(T) = S(0)J e^{N}$.
4. For $i = 1$ to $m$:
   (a) Generate $X_i$ given $N$ using Lemma 3.
   (b) Set $S_i^- = S_{i-1}^+ e^{X_i}$ and $S_i^+ = S_i^- J_i$.
   (c) If $S_i^- \leq B$ or $S_i^+ \leq B$, then return a zero estimate.

5. For $i = 1$ to $m$
   (a) Compute the probability of no barrier crossing $C_i$ using Lemma 1.

6. Set likelihood ratio $L = \prod_{i=1}^{m} C_i$.

7. No barrier crossing has occurred during the duration of the option. Return $\max(S(T) - K, 0)Le^{-rT} \left[ 1 - \Phi\left(\frac{\ln\left(\frac{S(T)}{K}\right) - \mu T}{\sigma \sqrt{T}}\right)\right]$ where $\Phi(x)$ denotes the cumulative distribution function of a standard normal evaluated at $x$.

Case 3 (To Estimate $E[V(S(0), 0)|N(T) > q]$):

1. Generate $Z$, a Poisson random variable with mean $\lambda T$ conditional on the event $Z > q$.

2. Use the algorithm for Case 2 to estimate $E[V|N(T) = Z]$.

Part V
New Algorithms

Building upon these past implementations, we now propose two new algorithms: one combining the methodologies of Algorithms C and D and another that applies a control variate technique to Algorithm D (Ross and Ghamami’s implementation) [18]. As our numerical results suggest, the latter algorithm produces the most efficient estimator with respect to all preceding algorithms.

Algorithm E: Hybrid Algorithm (Combining Algorithm C and D)

Our first proposed algorithm combines the methodologies of Ross & Ghamami (Algorithm D) with those of Joshi & Leung (Algorithm C). Since the estimator of Algorithm D is the most efficient – in terms of our previously defined measure of efficiency – to date, we posited that by stratifying on the number of jumps similar to that algorithm and sampling only non-zero estimates using importance sampling, our hybrid algorithm will provide a more efficient estimator than both Algorithms C and D.

Similar to Algorithm D, we decompose our price estimator into:
$$\text{DOC} = E[\mathbb{V}(S(0), 0)] = \sum_{m=0}^{q} E[\mathbb{V}|N(T) = m]P_m + E[\mathbb{V}|N(T) > q]\tilde{P}_q$$

But in this case, we calculate $E[\mathbb{V}|N(T) = m]$ similar to Algorithm C’s methodology:

1. Set likelihood ratio $L = 1$.
2. Generate jump-sizes $\{J_i\}_{i=1}^{m}$ from given jump-size distribution (i.e. log-normal). Set $J = \prod_{i=1}^{m} J_i$.
3. Generate $m$ uniform random numbers from $(0, T)$. Set them equal to $\{t_i\}_{i=1}^{m}$ where $t_p > t_q$ for all integers $m > p > q > 0$.
4. For $i = 1$ to $m$:
   (a) Generate the pre-jump stock price $S_i^-$ by using importance sampling to ensure that it is above the barrier.
   (b) Set $L = L \times \text{Pr}[S_i^- > B|S_i^{-1}]$.
   (c) Calculate the intra-period probability of no barrier crossing $\tilde{P}$ using Lemma 1. Set $L = L \times \tilde{P}$.
   (d) Generate the jump-size $J_i$ by using importance sampling to ensure that $S_i^+$ is above the barrier and set $S_i^+ = S_i^- J_i$.
   (e) Set $L = L \times \text{Pr}[S_i^+ > B|S_i^-]$.
5. Calculate the option value $\mathbb{V}(S(t_m), t_m)$ at time $t_m$ using the analytical option pricing formula. Return $\mathbb{V}(S(t_m), t_m)Le^{-r(t_m)}$.

Decomposing our estimator in this manner allows us to not only remove the possibility of non-zero estimates by using importance sampling, but it also allows us to more evenly sample through the use of proportional stratified sampling.

**Algorithm F: Applying The CV Method (Algorithm D + CV)**

From our preliminary numerical results, we determined that Ross and Ghamami’s implementation (Algorithm D) is the most efficient of the three previous publications. Hence, we analyzed whether utilizing the control variate method and applying it to Algorithm D would further reduce the sample variance of simulations.

The control variate method remains one of the most frequently used and most effective variance reduction techniques used in Monte Carlo simulations [5]. It utilizes the error estimates in known quantities to reduce variance in simulating unknown quantities. Since not many additional computations are
required for this method, the standard and control variate methods essentially take the same amount of time to simulate.

Suppose \( \{X_i\}_{i=1}^n \) are \( n \) estimates generated from a simulation and are i.i.d. random variables. In our case, \( X_i \) might be the estimated discounted payoff of a barrier option. Also, suppose that we have \( n \) i.i.d. random variables \( \{Y_i\}_{i=1}^n \) generated from the same simulation; however, contrary to the uncertainty surrounding the distribution of the \( X_i \)'s, the expected value and variance of each \( Y_i \) is known. Hence, we can use this knowledge about the distribution of the \( Y_i \)'s in order to provide a more accurate estimator for \( X \).

In a standard MC approach, our usual, unbiased estimator is
\[
\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i.
\]

However, the unbiased, control variate estimator is:
\[
\bar{X}(b) = \bar{X} - b(\bar{Y} - E[Y])
\]
where \( b \) is any fixed constant.

It is shown in [5] that the optimal coefficient \( b^* \) minimizes the variance of the control variate estimator and is given by:
\[
b^* = \frac{\sigma_X}{\sigma_Y} \rho_{XY} = \frac{\text{Cov}[X, Y]}{\text{Var}[Y]}
\]

The effectiveness of the control variate method depends on the correlation between \( X \) and \( Y \). The higher the correlation, the more effective \( Y \) is able to predict the value of \( X \). Therefore, in our proposed algorithm, we must choose a random variable that is highly correlated to the value of the option. Since the value of the option is an increasing function of our total jump-size \( J = \prod_{i=1}^m J_i \) and its distribution is known, we utilize \( J \) as our control variable. Thus, our estimator is:
\[
\hat{DOC} = \frac{1}{N} \sum_{i=1}^N (V_i - \hat{b}(J^{(i)} - E[J]))
\]
where \( V_i \) is the \( i \)th estimate of \( E[V | N(T) = m] \) using Algorithm D, \( J^{(i)} = \prod_{j=1}^m J^{(i)}_j \) is the total jump-size for the \( i \)th estimate, and \( \hat{b} \) is calculated as follows:
\[
\hat{b} = \frac{\sum_{i=1}^N (J^{(i)} - \bar{J})^2 (V_i - \bar{V})^2}{\sum_{i=1}^N (J^{(i)} - \bar{J})^2}
\]

\section{Numerical Results}

In this section, we analyze the efficiency of our proposed algorithms for pricing down-and-out call options using our previously defined measure of efficiency:
\[
E = \sqrt{\text{Var}[\hat{DOC}]} \times C
\]
where $E$ denotes the estimated efficiency, $C$ represents the total computational time for generating the estimates, and $\text{Var}[\hat{DOC}]$ represents the sample variance of $M$ option estimates, with $N$ price paths per option estimate.

For our numerical results, $M = 50$ option estimates and $N = 100,000$ price paths. Hence, each simulation performs $M \times N = 50 \times 100,000 = 5$ million iterations. We also consider risk-neutral pricing where the drift of the geometric Brownian motion of the stock price process is $\mu = r - \frac{\sigma^2}{2} - \lambda(E[J] - 1)$.

We run our simulations using a 2.4GHz Intel Core i5 MAC OS X and implement these algorithms in C++.

10.1 Monte Carlo Comparison

Our first example tests our algorithms as we vary the jump intensity $\lambda$ and measure the efficiency improvement from each algorithm. We are using Algorithm B as our base case of efficiency. The efficiency improvement $I_i$ is defined by:

$$I_i = \left( \frac{E_B}{E_i} - 1 \right) \times 100\%$$

where $E_i$ is the estimated efficiency for Algorithm $i$.

The value of the parameters are: $S(0) = 50$, $K = 55$, $B = 45$, $T = 1$ year, $r = 0.05$, $\sigma = 0.30$, $\mu_{\text{jump}} = 0$, and $\sigma_{\text{jump}} = 0.05$. The results are displayed in Table 1 and 2.

**Table 1: Relative Efficiency Improvements, with $M = 50$ and $N = 10^5$**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$\lambda = 2$</th>
<th>$\lambda = 4$</th>
<th>$\lambda = 8$</th>
<th>$\lambda = 16$</th>
</tr>
</thead>
<tbody>
<tr>
<td>B (Atiya &amp; Metwally)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>C (Joshi &amp; Leung)</td>
<td>14.9%</td>
<td>-14.5%</td>
<td>-15.0%</td>
<td>-17.6%</td>
</tr>
<tr>
<td>D (Ross &amp; Ghamami)</td>
<td>41.5%</td>
<td>25.8%</td>
<td>48.7%</td>
<td>32.7%</td>
</tr>
<tr>
<td>E (Alg. C + Alg. D)</td>
<td>16.2%</td>
<td>24.1%</td>
<td>41.1%</td>
<td>12.8%</td>
</tr>
<tr>
<td>F (Alg. D + C.V.)</td>
<td>59.9%</td>
<td>55.4%</td>
<td>65.4%</td>
<td>41.4%</td>
</tr>
</tbody>
</table>

**Table 2: Comparison of Variances, with $M = 50$ and $N = 10^5$**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$\lambda = 2$</th>
<th>$\lambda = 4$</th>
<th>$\lambda = 8$</th>
<th>$\lambda = 16$</th>
</tr>
</thead>
<tbody>
<tr>
<td>B (Atiya &amp; Metwally)</td>
<td>8.41 * 10^{-4}</td>
<td>1.01 * 10^{-3}</td>
<td>1.23 * 10^{-4}</td>
<td>1.60 * 10^{-3}</td>
</tr>
<tr>
<td>C (Joshi &amp; Leung)</td>
<td>2.21 * 10^{-4}</td>
<td>5.54 * 10^{-4}</td>
<td>5.94 * 10^{-4}</td>
<td>1.06 * 10^{-3}</td>
</tr>
<tr>
<td>D (Ross &amp; Ghamami)</td>
<td>1.76 * 10^{-4}</td>
<td>3.65 * 10^{-4}</td>
<td>3.95 * 10^{-4}</td>
<td>7.41 * 10^{-4}</td>
</tr>
<tr>
<td>E (Alg. C + Alg. D)</td>
<td>2.76 * 10^{-4}</td>
<td>3.27 * 10^{-4}</td>
<td>3.13 * 10^{-4}</td>
<td>6.54 * 10^{-4}</td>
</tr>
<tr>
<td>F (Alg. D + C.V.)</td>
<td>1.54 * 10^{-4}</td>
<td>2.01 * 10^{-4}</td>
<td>2.76 * 10^{-4}</td>
<td>5.82 * 10^{-4}</td>
</tr>
</tbody>
</table>

In every case of $\lambda$, Algorithm F, the control variate implementation of Algorithm D, has the most significant efficiency improvement. For the low jump intensity value of $\lambda = 2$, the correlation $\rho_{V,J}$ between the total jump-size $J$
and the value of the option $\hat{V}$ was roughly $0.25 < \rho_{\hat{V}, j} < 0.30$, whereas for the high values of $\lambda$, $0.45 < \rho_{\hat{V}, j} < 0.50$. This indicates that, as $\lambda$ increases, the correlation between the two variables increases and the jump-sizes contribute more to the price estimate. This is substantiated by the efficiency improvement from Algorithm D to Algorithm F.

We also find that although Joshi & Leung’s implementation (Algorithm C) initially improves upon the efficiency compared to Algorithm B, the greater values of $\lambda$ significantly increase the computational time, putting substantial downward pressure on its efficiency. Thus, the advantages of drawing no non-zero estimates quickly diminish as $\lambda$ increases.

In almost every case of Table 2, the variance of the hybrid method (Algorithm E) was lower than that of Algorithms B-D. However, this hybrid algorithm still suffers from the slower computational times of importance sampling, thereby continuing to make this algorithm an inefficient one compared to Algorithms D and F.

Our second example tests the variance convergence of the algorithms as $N$, the price paths per estimate, increases, which is represented by Figure 5. We consider five cases: $N = 10^3, 50^3, 10^4, 50^4, 10^5$. The parameters for this example are: $S(0) = 100, K = 110, B = 85, T = 1$ year, $r = 0.05, \sigma = 0.25, \lambda = 2, \mu_{\text{jump}} = 0.0, \text{ and } \sigma_{\text{jump}} = 0.10$.

Figure 5: Convergence of Variances as $N$ Increases, with $M = 50$

As we have seen in our previous examples, the control variate implementation (Algorithm F) appears to provide a lower bound for the variances, converging in variance the quickest among all other algorithms. Also, suggested by the stark contrast in variances between algorithms with and without variance reduction methodologies, these methodologies play an important role in efficiency improvements.
10.2 Monte Carlo vs. Randomized Quasi-Monte Carlo Comparison

In this section, we numerically compare Ross and Ghamami’s procedure (Algorithm D) with its RQMC implementation. Whereas our MC implementation draws random numbers from a Merseenne Twister pseudorandom number generator, the RQMC procedure generates estimates using the random-start, permuted Halton sequence algorithm given by Ökten and Xu [21].

The parameters for this example are: $S(0) = 50$, $K = 55$, $B = 45$, $T = 1$ year, $r = 0.05$, $\sigma = 0.30$, $\mu_{\text{jump}} = 0.0$, and $\sigma_{\text{jump}} = 0.05$. As in our previous example, the base case for the efficiency improvement is Atiya and Metwally’s implementation (Algorithm B). Our results are exhibited in Table 3 and Figure 6.

<table>
<thead>
<tr>
<th></th>
<th>MC</th>
<th>RQMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variance</td>
<td>$1.40 \times 10^{-3}$</td>
<td>$6.13 \times 10^{-4}$</td>
</tr>
<tr>
<td>Comp. Time (sec.)</td>
<td>20.79140</td>
<td>86.47420</td>
</tr>
<tr>
<td>Efficiency</td>
<td>0.7773</td>
<td>2.1409</td>
</tr>
<tr>
<td>Efficiency Improvement</td>
<td>4.73%</td>
<td>-61.98%</td>
</tr>
</tbody>
</table>

Figure 6: Convergence of MC and RQMC Variances as $N$ Increases, with $M = 50$

We see that although the variance of our RQMC simulation converges at a faster rate, the additional computational time required makes the algorithm highly inefficient. The additional computational time is over four times as much than by using pseudorandom number sampling. This could be remedied by drawing from a randomly-shifted Sobol’ sequence instead of a random-start, permuted Halton sequence, which may provide a necessary further speed-up.
Part VI

Conclusion

Usually, due to their complexity, the valuation of exotic options lacks analytical solutions, especially when the underlying stock price follows a jump-diffusion process. Given the importance of generating efficient, unbiased estimates to these problems, many financial engineers and academics have applied popular variance reduction techniques in an attempt to find the most efficient computational schemes.

In regards to the problem of pricing barrier options within the jump-diffusion framework, we have provided two efficient, unbiased procedures to the continuous problem. Due to the high correlation between the jump-size $J$ and the option price, Algorithm F, the control variate implementation, presents the most efficient estimator among all previously proposed algorithms.

Additionally, we presented a RQMC implementation that converged in variance quicker than its MC counterparts; however, the additional computational time required significantly reduced the efficiency of this simulation, thereby offsetting any variance improvements. As future work, we will consider using a Sobol’ sequence to draw samples since generators of Sobol’ sequences generate at speeds similar to that of many pseudorandom number generators.
References


