THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF RISK MANAGEMENT

VARIANCE REDUCTION TECHNIQUES FOR MONTE CARLO VALUATION OF FINANCIAL DERIVATIVES

STAN KOROSTIN
SPRING 2014

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Actuarial Science
with honors in Risk Management

Reviewed and approved* by the following:

Ron Gebhardtsbauer
Clinical Associate Professor of Actuarial Science
Thesis Supervisor/Honors Advisor

Lisa Posey
Associate Professor of Business Administration
Second Faculty Reader

* Signatures are on file in the Schreyer Honors College.
ABSTRACT

Monte Carlo simulations rely on repeated random sampling to represent real events. They have become a pivotal tool for valuating financial derivatives that do not have a closed form pricing formula. These simulations often take a long time to execute, especially when considering portfolios with thousands of options. There exist methods that mathematically alter the formulas for Monte Carlo simulation to reduce the variance and ultimately minimize the convergence time. This paper applies the naïve, control variate, antithetic, and stratified methods to Asian call options to examine which of the techniques quickens convergence to the true price. This paper also introduces quasi-Monte Carlo methods as efficient sequences for simulation. While the analysis focuses on Asian options specifically, the Monte Carlo methodology has many actuarial applications.
TABLE OF CONTENTS

List of Figures .......................................................................................................................... iii
List of Tables ............................................................................................................................. iv
Acknowledgements ................................................................................................................. v
Chapter 1 Introduction ........................................................................................................... 1
Chapter 2 Asian Options ......................................................................................................... 3
Chapter 3 Naïve Monte Carlo ............................................................................................... 6
Chapter 4 Control Variate Method ....................................................................................... 9
Chapter 5 Antithetic Variate Method ................................................................................... 13
Chapter 6 Stratified Sampling ............................................................................................... 15
Chapter 7 Optimization ......................................................................................................... 21
Chapter 8 Quasi-Monte Carlo Methods ............................................................................... 24
  Van der Corput Sequence ............................................................................................... 24
  Halton Sequence ............................................................................................................. 25
  Faure Sequences .............................................................................................................. 29
Chapter 9 Conclusion ............................................................................................................ 32
Appendix A Referenced Formulas ....................................................................................... 34
Appendix B Referenced Spreadsheets .................................................................................. 35
Appendix C Naïve Monte Carlo Simulation Code for Excel VBA ....................................... 37
REFERENCES .......................................................................................................................... 39
LIST OF FIGURES

FIGURE 1: Asian vs. European Call Payoffs ................................................................. 4
FIGURE 2: Control Variate Monte Carlo valuation ......................................................... 10
FIGURE 3: Impact of Boyle modification ........................................................................... 12
FIGURE 4: Antithetic Monte Carlo valuation ................................................................. 14
FIGURE 5: Chart of the Stratification Method ................................................................. 15
FIGURE 6: Stock path with random uniform random number ........................................... 16
FIGURE 7: Stock path with stratified uniform random uniform ........................................ 16
FIGURE 8: Stratified Monte Carlo valuation ................................................................. 19
FIGURE 9: Naïve Monte Carlo for Dimensions 1 x 2 ..................................................... 27
FIGURE 10: Halton Sequence for Dimensions 1 x 2 ...................................................... 27
FIGURE 11: Halton Sequence for Dimensions 15 x 16 ................................................... 28
FIGURE 12: Halton sequence for arithmetic Asian option valuation .............................. 29
LIST OF TABLES

TABLE 1: Naïve Monte Carlo Valuation ................................................................. 7
TABLE 2: Naïve Arithmetic Asian Call Valuation .................................................... 8
TABLE 3: Control Variate vs. Naïve Monte Carlo values ....................................... 12
TABLE 4: Naïve vs. Antithetic Monte Carlo values ............................................... 14
TABLE 5: Naïve vs. Stratified Monte Carlo valuation ............................................. 19
TABLE 6: Optimization statistics for Monte Carlo valuation .................................. 22
TABLE 7: Two-dimensional Halton Sequence ......................................................... 26
ACKNOWLEDGEMENTS

I would like to thank Ron Gebhardt sbauer for his advisory role in both my academic career and in the formulation of this paper. I would also like to thank Dr. Lisa Posey for her commitment in reviewing this paper.
Chapter 1
Introduction

For years, financial analysts relied on closed-form formulas to valuate financial instruments. However, as financial markets developed with complex derivatives, these formulas became obsolete in capturing the many input uncertainties. In 1964 David Hertz introduced the application of Monte Carlo valuation to the field of finance as a means to analyze complex financial instruments.

Monte Carlo valuations existed prior to this application in a broader sense as algorithms that produce repeated random sampling to simulate future possibilities. While these methods have proven invaluable in simulating models in finance, their innate flaw is that they require many trials to converge to the true value. In an economic climate where firms pay billions of dollars to minimize nano-seconds of computing speed, it becomes paramount that these simulations are modified with variance reduction methods to converge to the solution quicker.

In this paper we discuss ways to manipulate the simulation process to increase efficiency. These techniques leverage characteristics of financial derivatives in order to restructure the way in which the simulations are performed and ultimately reduce the variance. The methods tested in this paper are the control variate, antithetic, and stratification techniques. Each will be tested on an arithmetic Asian Call option to compare the benefits of the methods. While the reduction techniques are not difficult to implement, their results in improving convergence are significant. They also provide a foundation for further research in efficient Monte Carlo simulation.

While many firms employ proprietary methods that help them maintain a competitive advantage, academic papers have been written to advance the science of Monte Carlo methods. Perhaps the most up and coming development is the study of quasi-Monte Carlo methods. These methods rely on deliberate sequences to improve the convergence and give rise to deterministic error bounds. Several famous sequences will be examined to gain a fundamental understanding of this method: Van der Corput, Halton, and Faure. These Quasi-Monte Carlo methods further the efficiency of the simulations.
Monte Carlo methods are irreplaceable in evaluating financial derivatives. They exhibit certain properties that are appealing for the valuation of securities that have many inputs that can vary: exercise prices, stock prices, interest rates, volatility, etc. Still, to converge to the true answer, these simulations often require many averaged trials. The power of variance reduction techniques to improve the efficiency is evidenced in practical application. A comparison of these methods in calculating the price of an arithmetic Asian option will confirm the best approach for simulation.
Chapter 2

Asian Options

Monte Carlo simulations are profoundly shaping the way in which businesses model complex scenarios. However, before modeling it is important to understand the underlying financial derivatives. This paper will focus on a particular option called an Asian option.

For European options, the payoff is the maximum of zero and the difference of the stock, \( S(T) \), at expiration and strike price, \( K \):

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

However, Asian options are an exotic type of option in which the payoff depends on the average of the underlying security prices over a set period of time, defined below as \( A(0,T) \):

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

These options are particularly advantageous for several reasons. One Asian option can replace a series of European options. For example, if you are traveling to Europe over the next five summers, and you are worried the price of the Euro may go up too much, you might use one Asian call option, where the Euro price is averaged over those five summers, instead of purchasing five European call options. Additionally, Asian options limit the risk of the market being manipulated by buyers and sellers by taking averages over time rather than the stock price at exercise time. This same averaging process also reduces natural market volatility or sudden deviations as a result of unexpected news. Due to the lower risk of the option, they are cheaper than their European counterparts.

For consistency, this analysis of Monte Carlo methods will compare techniques executed on the arithmetic average price Asian call option. This option takes the arithmetic average of the stock prices over set periods of time and utilizes a fixed strike price to determine the payoff at the time of exercise. The following example will illustrate the basic payoff calculations of the arithmetic and geometric Asian options.
Example 1

Ron purchased an arithmetic Asian call and Evan purchased a geometric Asian call. Each option averages the stocks quarterly, expires after one year, and has a strike price of 7. Meanwhile a non-actuarial student purchased a one-year European call with strike price 7. The following are the stock prices at the end of each quarter: 10.50, 11.00, 9.75, and 4.00. Determine the payoff for each option.

Solution

The geometric mean is calculated as $\sqrt[4]{10.50 \times 11.00 \times 9.75 \times 4.00} = 8.19$

The arithmetic mean is simply the average of the 4 stock prices = 8.81

Following the payoff formulas:

Ron’s payoff is $\max(0, 8.81 - 7) = 1.81$

Evan’s Payoff is $\max(0, 8.19 - 7) = 1.19$

The non-actuarial student’s payoff is $\max(0, 4 - 7) = 0$.

Fixing the first three quarterly stock prices and varying the terminal stock price produces the following payoff graph:

FIGURE 1: Asian vs. European Call Payoffs
Although this is a basic example it illustrates two important facts about the Asian option. First the arithmetic average is always greater than the geometric average, which impacts the pricing of options (the arithmetic call option is more expensive than the geometric). Second, the example demonstrates how a European option can be affected by an unexpected fluctuation of a stock price at the time of exercise; the Asian options smooth the volatility of the payout.
Chapter 3

Naïve Monte Carlo

Before proceeding to apply complex variance reduction techniques to Monte Carlo simulations, it is important to understand the underlying process. Naïve Monte Carlo valuation is the process of simulating a series of random normally-distributed numbers and using them to make projections of the future. We will use Monte Carlo techniques to model a distribution around the mean riskless rate of return $r$ to project future stock prices, and the riskless rate to discount the payoffs; the assumption of using solely the riskless rate of return has been proven to produce solutions for options valuations, since the expected rate of return is unknown (Cox and Ross 1976). The following steps produce a naïve Monte Carlo valuation of an option (Boyle 1977):

1. Generate random numbers uniform on $[0, 1)$.
2. Transform $U[0, 1)$ into standard normal random number. A standard practice is the inversion method, which performs the inverse operation on the cumulative standard normal distribution.
3. Define:
   
   $r$ as the riskless rate of return
   
   $\delta$ as the dividend rate
   
   $T$ as the terminal time of the option
   
   $\sigma^2$ as the variance

   Calculate the normal random number: $n_i = m + v * z_i$

   where $m = T(r - \delta - .5(\sigma^2))$ and $v = \sigma \sqrt{T}$

4. Perform exponentiation on the normal random number to generate lognormal random numbers $x_i$

5. Multiply lognormal random numbers by the initial stock price to simulate projected stock prices at time $T$

6. Calculate the payoff of the option based on its contract stipulation at expiration

7. Discount the option to time 0 using the riskless rate of return to determine the trial price.
8. Average all of the trials to produce the naïve Monte Carlo price.

Steps 3 through 7 can be synopsized as follows:

\[
Call Price = \max \left( 0, \left( S(0) e^{r(T-t) - \frac{1}{2} \sigma^2 T} + \sigma \sqrt{T} N^{-1}(U_i) - K \right) \right)
\]

The variance for the sample mean of the naïve Monte Carlo Simulation is the variance of the distribution divided by number of trials \( n \).

The following example demonstrates the naïve Monte Carlo process.

**Example 2**

The following information was given about a stock following the lognormal model:

The initial stock price \( S_0 = 100 \)

The risk free rate \( r = 4\% \), volatility \( \sigma = 0.3 \), dividend rate \( \delta = 0.02 \), and time to expiration \( T = 1 \).

Three uniform random numbers on \([0,1)\) are simulated: 0.10, 0.55, 0.80.

Estimate the value of an at-the-money European call option expiring in one year.

**Solution**

Calculate \( m \) and \( v \) using the formula provided:

\[
m = 1 \cdot (0.04 - 0.02 - 0.5 \cdot 0.3^2) = -0.025 \quad \text{and} \quad v = 0.3 / 1 = 0.3
\]

The naïve Monte Carlo valuation is performed in Table 1.

**TABLE 1: Naïve Monte Carlo Valuation**

<table>
<thead>
<tr>
<th>Trial</th>
<th>Uniform Random#</th>
<th>Inversion Method</th>
<th>Normal Random # ( n \sim N(m,v) )</th>
<th>Lognormal Random # ( x \sim LN(m,v) )</th>
<th>Stock Price at time T ( S(T) = S(0)x )</th>
<th>Call Payoff ( \max(0, S(T) - K) )</th>
<th>Discount ( e^{-rt} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.10</td>
<td>-1.28</td>
<td>-0.41</td>
<td>0.66</td>
<td>66.40</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>0.55</td>
<td>0.13</td>
<td>0.01</td>
<td>1.01</td>
<td>101.28</td>
<td>1.28</td>
<td>1.23</td>
</tr>
<tr>
<td>3</td>
<td>0.80</td>
<td>0.84</td>
<td>0.23</td>
<td>1.26</td>
<td>125.54</td>
<td>25.54</td>
<td>24.54</td>
</tr>
</tbody>
</table>

The discounted call payoffs are averaged to estimate the call price of \( 8.59 \).
The follow example demonstrates how to apply a naïve Monte Carlo process to an Asian option.

Example 3

Consider the stock with the given inputs from Example 2.

Four uniform random numbers on [0,1) are provided: .30, .75, .55, .80.

Produce one Monte Carlo trial price for a quarterly averaged, arithmetic Asian call option.

Solution

For the quarterly averaged Asian option, the four uniform random numbers are used to project the path of the stock price: S(1/4), S(1/2), S(3/4), S(1).

Calculate $m = .25(0.04 - 0.02 - 0.5(-.3)\text{ }^2) = -0.00625$ and $v = 0.3(0.5) = 0.15$

<table>
<thead>
<tr>
<th>Time</th>
<th>Uniform Random#</th>
<th>Inversion Method</th>
<th>Normal Random #</th>
<th>Lognormal Random #</th>
<th>Stock Price at time t</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{4}$</td>
<td>0.30</td>
<td>-0.52</td>
<td>-0.08</td>
<td>0.92</td>
<td>91.86</td>
</tr>
<tr>
<td>$\frac{1}{2}$</td>
<td>0.75</td>
<td>0.67</td>
<td>0.09</td>
<td>1.10</td>
<td>101.01</td>
</tr>
<tr>
<td>$\frac{3}{4}$</td>
<td>0.55</td>
<td>0.13</td>
<td>0.01</td>
<td>1.01</td>
<td>102.29</td>
</tr>
<tr>
<td>1</td>
<td>0.80</td>
<td>0.84</td>
<td>0.12</td>
<td>1.13</td>
<td>115.33</td>
</tr>
</tbody>
</table>

The average stock price is 102.62. Therefore the payoff is 2.62, which is then discounted 1-year to price the Arithmetic Asian Call at 2.52.
Chapter 4

Control Variate Method

The Control Variate method for Monte Carlo variance reduction is a powerful variance reduction tool if the underlying option satisfies two characteristics: the underlying option must be highly correlated with another option (the control variate), and that control variate must have a closed form solution. The method takes the difference between the simulated value and the closed form value of the control variate and adjusts the simulated value of the underlying option accordingly to remove standard error from simulation. For the arithmetic Asian option $A^*$, the control variate formula becomes:

$$A^* = \bar{A} + (\bar{G} - \tilde{G})$$

where $\bar{A}$ and $\bar{G}$ are the arithmetic and geometric Monte Carlo prices and $G$ is the Black-Scholes price (Kemna and Vorst 1990).

Evaluating the variance from both sides of the equation yields:

$$Var(A^*) = Var(\bar{A}) + Var(\bar{G}) - 2Cov(\bar{A}, \bar{G})$$

It is clear that for two highly correlated derivatives, a greater covariance will reduce variance.

The following example compares the control variate method to naïve Monte Carlo for an Arithmetic Asian call option.

Example 4

Consider a non-dividend paying stock.

The stock and strike price are equal to 100.

Volatility, $\sigma = .3$ and the risk free rate, $r = .04$

Perform naïve and control variate Monte Carlo simulations with 1000 trials to determine the variance of a 1-year arithmetic Asian call option that averages quarterly under each scenario.
Solution

For the naïve method, the procedure is identical to the procedure outlined in Chapter 2. After running 1000 trials in Excel, the variance was calculated by dividing the sample variance by the number of trials, which yields 0.179.

The control variate method is more complicated because it also requires the closed-form solution for the price of the geometric Asian call option and the simulated geometric price. The formula for the geometric Asian call in Appendix A produces a geometric call price of 8.83. Additionally, the geometric Asian call variance was 0.167 and the covariance between the arithmetic and geometric Asian call options was 0.173. Calculating the sample variance with the formula presented in this chapter and dividing by the number of trials results in a variance of 0.00048 for the control variate method, or 0.3% of the naïve method. Undoubtedly, the variance reduction of approximately 99.7% is tremendous. Figure 2 illustrates the dramatic oscillations of the naïve Monte Carlo method, compared to the quicker convergence of the control variate estimate. The spreadsheet for the naïve Monte Carlo calculation is included in Appendix B.

FIGURE 2: Control Variate Monte Carlo valuation

Control Variate vs. Naïve Monte Carlo
Arithmetic Asian Call

![Graph showing control variate and naïve Monte Carlo comparison](image)
The general closed formula for estimating the value of a non-dividend Asian arithmetic call option is included in Appendix A and suggests a value of approximately 9.25.

Boyle Modification

In *Monte Carlo Methods for Security Pricing*, it is derived that adding a coefficient of correction $\beta$ further minimizes the variance for a specific value by adjusting for the covariance between the underlying option and the covariate (Boyle, Broadie, and Glasserman 1997). Consider the control variate method with $\beta$ as the coefficient of correction for the Monte Carlo geometric valuation:

$$A^* = \bar{A} + \beta(G - \bar{G})$$

The new variance becomes:

$$Var(A^*) = Var(\bar{A} + \beta \bar{G}) = Var(\bar{A}) + \beta^2(\bar{G}) - 2\beta Cov(\bar{A}, \bar{G})$$

This variance is minimized when:

$$\beta = \frac{Cov(A, G)}{Var(G)}$$

After performing the control variate valuation in Example 4, the simulated covariance and variance from the example were leveraged to calculate $\beta$:

$$\beta = \frac{(0.182)}{(0.179)} = 1.03$$

This $\beta$ was implemented as the coefficient of correction in a Monte Carlo valuation using the Control Variate method with Boyle modification. Applied to the same stock and option from the previous example, the Boyle Modification yields a variance of 60% of the pure control variate method and about 0.2% of the naïve estimate for 1000 trials. With the Boyle Modification, the Monte Carlo simulation converges on the true estimated value of the arithmetic Asian call option sooner, as evidenced by the graph of the first 100 trials in Figure 3.
Table 3 summarizes the calculated Monte Carlo price at various trial intervals and further supports the implementation of the control variate method including Boyle Modification.

**TABLE 3: Control Variate vs. Naïve Monte Carlo values**

<table>
<thead>
<tr>
<th>Trial</th>
<th>Naïve</th>
<th>Control Variate</th>
<th>CV with $\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4.1096</td>
<td>8.8314</td>
<td>8.8813</td>
</tr>
<tr>
<td>5</td>
<td>11.7999</td>
<td>8.8475</td>
<td>9.0821</td>
</tr>
<tr>
<td>10</td>
<td>13.0701</td>
<td>8.9230</td>
<td>9.1482</td>
</tr>
<tr>
<td>100</td>
<td>7.9976</td>
<td>9.1524</td>
<td>9.1699</td>
</tr>
<tr>
<td>1,000</td>
<td>9.2343</td>
<td>9.2667</td>
<td>9.2390</td>
</tr>
<tr>
<td>10,000</td>
<td>9.0734</td>
<td>9.2602</td>
<td>9.2471</td>
</tr>
<tr>
<td>500,000</td>
<td>9.2346</td>
<td>9.2505</td>
<td>9.2503</td>
</tr>
<tr>
<td>1,000,000</td>
<td>9.2551</td>
<td>9.2501</td>
<td>9.2500</td>
</tr>
</tbody>
</table>
Chapter 5

Antithetic Variate Method

The antithetic variate method reduces variance by modifying the way in which uniform random numbers are generated (Boyle, Broadie, and Glasserman 1997). For each uniform random number \( U[0,1) \), an antithetic variate

\[ 1 - U[0,1) \]

is generated as well. Hence:

\[ \bar{A} = \frac{(A_1 + A_2)}{2} \]

Since \( \text{Var}(A_1) = \text{Var}(A_2) \), the simulation variance is:

\[ \text{Var}(\bar{A}) = \frac{1}{2} [\text{Var}(A_1) + \text{Cov}(A_1, A_2)] \]

Since the uniform random numbers are clearly negatively correlated with their complements, there will be a variance reduction. However, given that generating each antithetic variate takes the same amount of time as generating the random number itself, this method will indeed reduce the number of trials required for convergence, but may not reduce operational efficiency as significantly as other variance reduction methods.

Example 5

Consider the non-dividend paying stock from Example 4. The stock and strike price are equal to 100.

Perform naïve and antithetic variate Monte Carlo simulations to determine the variance after 1000 trials of a 1-year arithmetic Asian call option that averages quarterly. Note that for each trial, two different stock paths are generated: the naïve path and the antithetic path. These paths are then averaged to calculate antithetic variate method price.

Solution

Following the same procedure and employing the same model inputs as in example three, yields an antithetic method variance of approximately 0.06 for 1000 trials, as compared to the naïve simulation variance of approximately 0.2. For the model, there is a significant 70% reduction in the variance of the estimate. Figure 4 confirms that the antithetic method begins to demonstrate far greater stability in its estimate of the arithmetic Asian call option in the first 100 trials.
Table 4 summarizes the Monte Carlo values for the naïve and antithetic methods at various trial intervals. Considering the previously estimated arithmetic Asian call option price of 9.25, the data suggests that antithetic valuation will converge quicker.

**TABLE 4: Naïve vs. Antithetic Monte Carlo values**

<table>
<thead>
<tr>
<th>Trial</th>
<th>Naïve</th>
<th>Antithetic</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4.1096</td>
<td>5.5379</td>
</tr>
<tr>
<td>5</td>
<td>11.7999</td>
<td>7.2932</td>
</tr>
<tr>
<td>10</td>
<td>13.0701</td>
<td>7.7688</td>
</tr>
<tr>
<td>100</td>
<td>7.9976</td>
<td>8.4939</td>
</tr>
<tr>
<td>1,000</td>
<td>9.2343</td>
<td>9.1327</td>
</tr>
<tr>
<td>10,000</td>
<td>9.0734</td>
<td>9.2226</td>
</tr>
<tr>
<td>500,000</td>
<td>9.2346</td>
<td>9.2414</td>
</tr>
<tr>
<td>1,000,000</td>
<td>9.2551</td>
<td>9.2541</td>
</tr>
</tbody>
</table>
Chapter 6
Stratified Sampling

The stratified sample method performs a transformation on the uniform random numbers generated during simulation to reduce variance by guaranteeing that there is an even distribution of random numbers across all strata.

For instance, to break four uniform random numbers into four strata, uniform numbers would be transformed to ensure that they fall within the intervals: [0, 0.25), [0.25, 0.5), [0.5, 0.75), and [0.75,1). The process for scaling the random numbers into respective strata is to multiply the randomly generated number by the width of the strata and add the lowest left point of each interval to the product (Boyle et al. 1997).

The following example demonstrates stratification.

Example 6
Stratify the following four random numbers into intervals with length 0.25:

0.15, 0.30, 0.95, 0.86

Solution
Following the procedure above: [Random Number] * [Interval Length] + [Left Interval Point]

The uniform random numbers and stratified uniform random numbers are calculated and charted in Figure 5. The colors in the calculations correspond to the arrows that map the locations in the chart.

FIGURE 5: Chart of the Stratification Method
To show the benefit of stratification, uniform random numbers were generated and used to project quarterly stock prices four times. The first uniform random number for each stock projection, $U_1$, was left random and the following three uniform random numbers, $U_2$, $U_3$, and $U_4$, were fixed for each of the four projections to highlight the impact of just the first uniform random number. Figure 6 shows the projected stock paths and demonstrates how simulation randomness in naïve Monte Carlo can yield stock projections that do not capture the full range of possible projections, which then limits the possibility of convergence to the true answer.

**FIGURE 6: Stock path with random uniform random number**

![Stock path with random uniform random number](image)

Then the same uniform random numbers, $U_1$, were each stratified into four strata and the following three uniform random numbers, $U_1$, $U_2$, and $U_4$, were kept constant to produce the stock projections in Figure 7. This illustrates how stratification can help capture a fuller range of projections quicker, and thus simulate results more efficiently.

**FIGURE 7: Stock path with stratified uniform random uniform**

![Stock path with stratified uniform random uniform](image)
The variance from the stratified Monte Carlo method can be calculated as:

\[ \sum_k p_k^2 \frac{\sigma_k^2}{n_k} \]

where \( p_k \) is the probability of each strum occurring, \( n_k \) is the sample size for each stratum, and \( \sigma_k^2 \) is the variance for each stratum.

(Marnay and Strauss 2001)

The following example compares the Monte Carlo simulation with four strata to test the significance of variance reduction.

**Example 7**

Consider the non-dividend paying stock from Example 4.

The stock and strike price are equal to 100.

Perform naïve Monte Carlo simulations and stratified Monte Carlo simulations with four strata and 1000 trials to determine the variance of a 1-year arithmetic Asian call option that averages quarterly under each scenario.

**Solution**

The procedure from Example 5 is followed to generate random numbers for each of the four partitions cyclically for 1000 trials, as follows. Since Asian options are path dependent, it is not appropriate to force an entire path into strata and calculate the stock prices incrementally. Therefore, the solution is to first find the stock price at the terminal point and then to calculate a “Brownian bridge” which generates a stochastic process for the path until the terminal value is reached. Simply employ the stratification method formulized above to calculate terminal stock values. The Brownian Bridge then recursively calculates midpoints of the subintervals.

The first two Brownian motions at the endpoints are:

\[ w(0) = 0 \text{ and } w(1) = \sigma \cdot N^{-1}(u_1) \]

Where \( u_1 \) is a stratified uniform random number and \( w(1) \) is the terminal point.
Then the midpoints are calculated as:

\[ w\left(\frac{1}{2}\right) = \frac{w(1)}{2} + \frac{1}{2} \sigma \cdot N^{-1}(u_2) \]

\[ w\left(\frac{1}{4}\right) = \frac{w\left(\frac{1}{2}\right)}{2} + \frac{\sqrt{2}}{4} \sigma \cdot N^{-1}(u_3) \]

\[ w\left(\frac{3}{4}\right) = \frac{w\left(\frac{1}{2}\right) + w(1)}{2} + \frac{\sqrt{2}}{4} \sigma \cdot N^{-1}(u_4) \]

(Dahl and Benth 2001)

This process, called the Wiener Path, can be generalized by proceeding to fill midpoints of the subintervals and by simply changing the coefficient in front of the second term to:

\[ t_j = \frac{T}{2^j + \log 2^j} \text{ where } j \text{ represents the number of the midpoint step beginning with 1.} \]

Confirm the calculations for the first and second midpoint steps in the example above:

When \( j = 1 \):

\[ t_j = \frac{1}{2^1 + \log 2} = \frac{1}{2} \]

When \( j = 2 \):

\[ t_j = \frac{1}{2^2 + \log 2} = \frac{\sqrt{2}}{4} \]

Then the price of the stock at the various times can be found with the lognormal stock model:

\[ S(t) = S(0) \cdot e^{\left(r - \frac{1}{2} \sigma^2\right)t + w(t)} \]

The variance for the stratified method with four strata equals 0.005, or less than 3% of the naive Monte Carlo Variance. Figure 8 displays the converging capabilities of the stratified method for the first 100 trials.
FIGURE 8: Stratified Monte Carlo valuation

Table 5 summarizes the naïve and stratified Monte Carlo values for the Arithmetic Asian Call at various trial intervals. Considering the true closed form estimate of 9.25, it supports Figure 6 in suggesting that the stratified method will converge to the true answer quicker.

<table>
<thead>
<tr>
<th>Trial</th>
<th>Naïve</th>
<th>Stratified (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4.1096</td>
<td>6.0483</td>
</tr>
<tr>
<td>5</td>
<td>11.7999</td>
<td>7.6604</td>
</tr>
<tr>
<td>10</td>
<td>13.0701</td>
<td>5.5715</td>
</tr>
<tr>
<td>100</td>
<td>7.9976</td>
<td>9.3924</td>
</tr>
<tr>
<td>1,000</td>
<td>9.2343</td>
<td>8.7232</td>
</tr>
<tr>
<td>10,000</td>
<td>9.0734</td>
<td>9.1322</td>
</tr>
<tr>
<td>500,000</td>
<td>9.2346</td>
<td>9.2597</td>
</tr>
<tr>
<td>1,000,000</td>
<td>9.2551</td>
<td>9.2540</td>
</tr>
</tbody>
</table>

While the stratified method is useful in variance reduction, it is even more important in allocating probabilities to various partitions when the calculations require distributions that are not uniform across all values. It is particularly useful in dealing with statistics that require polling, to ensure that all population demographics are proportionally represented. The techniques outlined above for calculating the Brownian
Bridge are also important to understand, as they are a useful foundation for calculating other path dependent options, such as barrier options and American options.
Chapter 7

Optimization

Time is money. What Benjamin Franklin did not know when he spoke these words is how expensive nanoseconds could be. Securities markets are more efficient than ever as high-frequency securities traders have programmed computers to trade in nanoseconds of time. Nanoseconds of time for transactions are so valuable that high-frequency traders invested $2 billion into their infrastructure in 2010.

Similarly, the hedging role requires that an actuary simulate portfolios that are often composed of thousands of securities. After the 2008 financial crisis there is increased attention for the tails of distributions or simulating probabilities of extremely unlikely and expensive events occurring. Large numbers of trials are needed for good resolution in these simulations. That means that program efficiency can mean the difference between an actuarial function requiring hours of time or minutes.

To test the Monte Carlo Methods, each was timed until convergence to the price of an arithmetic Asian call, estimated by the process included in Appendix A. The run was considered to converge when two consecutive trials produced Monte Carlo averages equivalent to the true answer, accurate to the nearest hundredth of a dollar. An exception was considered for the stratified method, which compared the averages after every fourth trial, since the method stratified four partitions of uniform random numbers. Although this is a crude method for testing convergence and in reality it would require many more trials to achieve a permanent convergence, it is sufficient for comparing relative strength of the methods. It is also not practical to increase the accuracy of the simulation, given that the closed form pricing formula for the arithmetic Asian call provides an estimated answer.

The number of trials required until convergence were calculated for each run. A total of 1000 runs were executed for each of the Monte Carlo Methods to reduce variability in the estimates of number of trials and total run time. The VBA code for the naïve simulation is included in Appendix C. Calculations for the other methods are simply derived by implementing the methods following the formulas in the paper. Table 5 summarizes that total time to complete 1000 runs and the average number of trials required for each run to “converge” by meeting the requirements described above. Note that the times may vary with
different computing capabilities, but each of the Monte Carlo methods will yield results that are relative to Table 6.

**TABLE 6: Optimization statistics for Monte Carlo valuation**

<table>
<thead>
<tr>
<th></th>
<th>Naïve</th>
<th>Control Variate</th>
<th>CV with B</th>
<th>Antithetic</th>
<th>Stratified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Time: 1000 runs (min)</td>
<td>4.42</td>
<td>1.45</td>
<td>0.96</td>
<td>3.01</td>
<td>2.30</td>
</tr>
<tr>
<td>Avg # trials per run</td>
<td>23,872</td>
<td>5,831</td>
<td>3,819</td>
<td>5,668</td>
<td>9,690</td>
</tr>
<tr>
<td>Var. for 1000 trials (% of Naïve)</td>
<td>100%</td>
<td>0.30%</td>
<td>0.20%</td>
<td>30%</td>
<td>3%</td>
</tr>
</tbody>
</table>

The naïve Monte Carlo method was first tested and set the benchmark at 4.42 minutes for 1000 runs to converge and an average of 23,872 Monte Carlo trials per run.

It is unsurprising that the control variate method and control variate with Boyle modification reduced time and number of trials until convergence commensurate with their variance reduction capabilities. In fact, the control variate method with Boyle modification was the most successful in reducing the time until convergence and only took a quarter of the time for the naïve Monte Carlo method.

The antithetic method exhibited the lowest number of average trials per run because it generated both a uniform random number and its complement in each trial, so it really needed an average of 2 times 2,834 = 5,668 trials. Execution time in this method was certainly reduced, especially since drawing the complement of each uniform random number is similar to stratification because it captures uniform random numbers from each half of the interval [0,1). Still, the overall execution time was not so impressive because it does consume much more time to generate a uniform random number as it does to take the complement of a uniform random number.

The stratified method required many trials per run to converge, since the calculated price at every fourth trial was considered. Neighboring trials will always oscillate because they are fixed to be in different strata. The method displayed a favorable execution time to calculate 1000 runs because of the simplicity of the formula and the benefits of guaranteeing that each of the strata is equally represented in the data. It is optimal to increase the number of strata for more robust convergence stipulations.

Overall, the optimization statistics showed that the variance reduction methods were successful in reducing the time to execute 1000 converging runs for the Asian option, with the Control Variate with
Boyle Modification converging the quickest. If these tests were scaled to price a portfolio with thousands of options, it becomes apparent that these methods cut down hours of time.
Chapter 8

Quasi-Monte Carlo Methods

The Monte Carlo methods that have been outlined in this paper thus far have relied on generating random numbers to simulate stock prices. However, this methodology is limited in that it produces probabilistic errors and it often takes a long time to converge. Quasi-Monte Carlo methods eliminate these problems by relying on deterministic sequences to simulate stock prices, rather than random numbers. They are referred to as low-discrepancy sequences because they aim to generate successive numbers in the furthest possible position from one another (Boyle, Joy, and Tan 1966). Three popular sequences for pricing options are the Van der Corput, Halton, and Faure sequences. They are derived to replace uniform random numbers in simulation.

Van der Corput Sequence

The Van der Corput sequence is a basic one dimensional sequence that was first published in 1935 by Dutch mathematician J.G. van der Corput. It is constructed by taking the sequence of natural numbers \( (1, 2, 3, \ldots) \) and defining it in base 2 as follows:

\[
n = \sum_{j=0}^{m} a_j(n)2^j
\]

The base representation of the natural numbers is then reflected about the decimal to produce the Van der Corput Sequence:

\[
\Phi(n) = \sum_{j=0}^{m} a_j(n)2^{-j-1}
\]

An example will clarify how to produce a number in the Van der Corput sequence.
Example 8

Produce the value for the number 6 in the Van der Corput sequence.

Solution

First, 6 should be written in base 2:

\[ 6 = 1(2)^2 + 1(2)^1 + 0(2)^0 = 110 \]

This implies:

\[ a_0 = 0, a_1 = 1, \text{ and } a_2 = 1 \]

Now reflect these elements about the decimal:

\[ \Phi_6 = \frac{0}{2} + \frac{1}{2^2} + \frac{1}{2^3} = \frac{3}{8} \]

The first three reflected numbers are \( \frac{1}{2}, \frac{1}{4}, \text{ and } \frac{3}{8} \), which one would think are uniformly spaced across the interval \([0,1)\), but they aren’t uniformly distributed. The mean is \( \frac{1}{2} \) appropriately, but the variance is \( 1/24 \), not the mean of the Uniform Distribution = \( 1/12 \). If we bring in the next 4 numbers (\( 1/8, 5/8, 3/8, 7/8 \)) we see that all 7 numbers look evenly spaced, and the mean is still \( 1/2 \), but the variance is \( 1/16 \).

As more numbers in the sequence are introduced, the variance of the sequence continues to approach the variance of the uniform distribution. In the next section, we will discuss how these numbers are used.

Halton Sequence

The Halton sequence derives quasi-random numbers for multiple dimensions. The dimension is defined by the number of evaluation points in the path of the security. For example, a European call will have one dimension, but the quarterly evaluated arithmetic Asian call option considered in this paper will have four dimensions. The first dimension of the Halton sequence is the Van der Corput sequence calculated in base 2. Each successive dimension is a generalized Van der Corput sequence evaluated in the base of the next greatest prime number (2, 3, 5, 7…).
In general, a number \( n \) in the natural sequence can be defined in base \( r \) as:

\[
n = \sum_{j=0}^{m} a_j(n)r^j
\]

The base representation of the natural numbers is then reflected about the decimal to produce the generalized van der Corput sequence:

\[
\Phi(n) = \sum_{j=0}^{m} a_j(n)r^{-j-1}
\]

Table 7 demonstrates how the Van der Corput sequence in base 2 and the generalized Van der Corput sequence in base 3 are the components of a two-dimensional Halton sequence.

<table>
<thead>
<tr>
<th>( m )</th>
<th>Base 2</th>
<th>( \Phi_n^1 )</th>
<th>Base 3</th>
<th>( \Phi_n^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1/2</td>
<td>1</td>
<td>1/3</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>1/4</td>
<td>2</td>
<td>2/3</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>3/4</td>
<td>10</td>
<td>1/9</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>1/8</td>
<td>11</td>
<td>4/9</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>5/8</td>
<td>12</td>
<td>7/9</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
<td>3/8</td>
<td>20</td>
<td>2/9</td>
</tr>
<tr>
<td>7</td>
<td>111</td>
<td>7/8</td>
<td>21</td>
<td>5/9</td>
</tr>
</tbody>
</table>

Notice how when calculating in base 2, the numbers uniformly fill the gaps on \([0,1)\) when there are either 3, 7, ... numbers in the sequence. In base 3, the gaps are uniformly filled when there are either 2, 8, ... numbers in the sequence. In fact, the number of quasi-Monte Carlo numbers needed to uniformly fill the gaps in a sequence with base \( r \) and any positive \( k \), can be calculated as:

\[
N = r^k - 1
\]

The Halton sequence helps Monte Carlo valuations converge quicker than randomly generated numbers between 0 and 1 by ensuring uniformly spaced quantiles. The comparison of Figures 9 and 10 demonstrate this point for the first two dimensions with 1000 iterations.
However, the drawback of the Halton sequence is that higher dimensions are correlated due to the large prime bases in higher dimensions that have longer cycles to fill the gaps, which causes multi-dimensional clustering. Figure 11 demonstrates multi-dimensional clustering at the fifteenth dimension for 1000 iterations. In practice, the Halton sequence is rarely used when dealing with more than eight dimensions.
The following example demonstrates the convergence strength of the Halton Sequence compared to naïve Monte Carlo valuation.

**Example 9**

Consider the same non-dividend paying stock from Example 3.

Perform naïve Monte Carlo simulation and Halton quasi-Monte Carlo simulation with 500 trials for a 1-year at-the-money arithmetic Asian call option that averages quarterly under each scenario. Graph the results.

**Solution**

The naïve Monte Carlo simulation is performed following the same procedure as outlined in this paper. To perform the Halton quasi-Monte Carlo simulation, a 4-dimensional Halton Sequence was generated in place of uniform random numbers and utilized to calculate the stock path. An extract of this sequence is available in Appendix B. From these stock paths, the arithmetic Asian call payout and price was determined. Figure 12 displays the results for the first 100 trials to demonstrate the quicker convergence for the Halton sequence. Bhat (2001) concluded that the Halton sequence produces a convergence after 100 trials equivalent to the convergence of the naïve Monte Carlo at 1000 trials.
Faure Sequences

The Faure sequence is similar to the Halton sequence, but it uses only one selected base for each of the dimensions and it employs a permutation of vector elements for each dimension. The base is selected as the smallest prime number greater than or equal to the number of dimensions. This selection process enables the Faure sequence to perform better at filling the gaps in the sequence at higher dimensions because the base is much smaller. The permutation of vector elements helps to reduce correlations between sequential higher dimensions.

The formula for generating the first Faure number is the same as the formula for the Halton sequence. The remaining elements of the Faure sequence generated by reordering prior elements as follows:

$$a^k_j = \sum_{i=0}^{m} \binom{i}{j} a^{k-1}_i (n \mod r)$$
Utilizing these elements, successive points in the Faure Sequence can be generated as follows:

\[ \Phi^k_r(n) = \sum_{j=0}^{m} a^k_j(n)r^{-j-1} \]

(Boyle, Joy, and Tan 1966)

The following example describes this process.

**Example 10**

Calculate two dimensions of the Faure sequence for the number \( n = 7 \).

**Solution**

First, select a base that is the smallest prime number less than or equal to the number of dimensions, which is equal to 2. Then write the number 7 in base 2:

\[ 7 = 1(2)^2 + 1(2)^1 + 1(2)^0 = 111 \]

This implies

\[ a_0 = 1, a_1 = 1, \text{and } a_2 = 1 \]

From these elements, the first dimension (the superscript 1 after the \( \Phi \)) is calculated:

\[ \Phi^1_7 = \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} = \frac{7}{8} \]

Now, use the prior elements (111) to calculate the elements for the second dimension:

\[ a^1_0 = ((1)(1) + (1)(1) + (1)(1)) \mod 2 = 1 \]

\[ a^1_1 = ((1)(1) + (2)(1)) \mod 2 = 1 \]

\[ a^1_2 = ((1)(1)) \mod 2 = 0 \]

The Faure sequence for the second dimension is generated as follows:

\[ \Phi^2_7 = \frac{1}{2} + \frac{1}{2^2} + \frac{0}{2^3} = \frac{3}{4} \]
These sequences aim to serve as an introduction to quasi-Monte Carlo methods. There are other sequences, such as the Sobol sequence, that have proved advantageous in higher dimensions. Unlike Monte Carlo simulation, quasi-Monte Carlo simulations will converge at a consistent time, since the sequences are fixed. Their convergence efficiencies under various sensitivity tests can be monitored in future simulation research.
Chapter 9

Conclusion

The simulations performed in this paper have demonstrated the effectiveness in choosing Monte Carlo simulations when closed form pricing methods are not available for complex options. Undoubtedly, while these simulations are effective, they are not always efficient. There are Monte Carlo methods that improve the convergence time to the price by manipulating the structure of the naïve Monte Carlo method.

All methods in this paper specifically focused on a quarterly averaged arithmetic Asian option, because its path dependent nature introduced a multidimensional modeling challenge. The simulated price of all methods are summarized.

The results suggested that the control variate method with Boyle modification yielded the most favorable results, due to the correlated nature of the geometric and arithmetic Asian options. With customized exotic options, sometimes there is no control variate that satisfies the described characteristics necessary for this method. Additionally, the control variate method is not effective for simulation in the tail quintiles of a distribution, because the appropriate control variate in this scenario is often unknown. Working solely on the option itself, the stratified method was also efficient with four strata, and has the potential for further efficiency with optimal strata selection. The antithetic method was slightly more favorable than naïve simulation due its innate ability to generate complements of random numbers and guarantee a more uniform sample of random numbers. Still, its purpose of obtaining additional uniform random numbers by taking the complement was obsolete, because uniform random numbers can be generated quickly with software.

Perhaps the most impressive, recent developments in simulation are the quasi-Monte Carlo sequences. A gentle introduction for creating these sequences was included in this paper and could be used as a stepping stone to further research. A preliminary glance at the basic Halton sequences for pricing a four-dimensional option suggested significant improvements to convergence time.

Moving forward with these findings suggests a future study: testing quasi-Monte Carlo methods at various dimensions to compare efficiencies for pricing a path dependent option. Future research should
focus on options with exact closed form pricing solutions to support more accurate convergence standards. Monte Carlo methods are a powerful tool for increasing pricing efficiency and will continue to play a pivotal role in options pricing.
Appendix A

Referenced Formulas

Geometric Asian Option Formula (McDonald):

The price for a geometric Asian option with \( N \) averaging periods employs the Black-Scholes pricing formula with modified dividend and volatility as follows:

\[
\delta^* = \frac{1}{2} \left[ r \frac{N - 1}{N} + (\delta + 0.5\sigma^2) \frac{N + 1}{N} - \sigma^2 \frac{(N + 1)(2N + 1)}{6N^2} \right]
\]

\[
\sigma^* = \frac{\sigma}{\sqrt{N}} \sqrt{\frac{(N + 1)(2N + 1)}{6}}
\]

Non-dividend Arithmetic Asian Option Estimate (Wiklund)

\[
c \approx e^{-rT} \left[ \left( \frac{1}{n} \sum_{i=1}^{n} e^{\mu_i + \sigma_i^2/2}N \left( \frac{\mu - \ln(K)}{\sigma_x} + \frac{\sigma_i}{\sigma_x} \right) \right) - KN \left( \frac{\mu - \ln(K)}{\sigma_x} \right) \right]
\]

\[
p \approx e^{-rT} \left[ KN \left( -\frac{\mu - \ln(K)}{\sigma_x} \right) - \left( \frac{1}{n} \sum_{i=1}^{n} e^{\mu_i + \sigma_i^2/2}N \left( \frac{\mu - \ln(K)}{\sigma_x} + \frac{\sigma_i}{\sigma_x} \right) \right) \right]
\]

Where:

\[
\mu_i = \ln(S) + (r - \sigma^2/2)(t_1 + (i - 1)\Delta t)
\]

\[
\sigma_i = \sigma \sqrt{(t_1 + (i - 1)\Delta t)}
\]

\[
\sigma_{xi} = \sigma^2 (t_1 + \Delta t((i - 1) - i(i - 1)/2n))
\]

\[
\mu = \ln(S) + (r - \sigma^2/2)(t_1 + (n - 1)\Delta t/2)
\]

\[
\sigma_x = \sigma \sqrt{t_1 + \Delta t(n - 1)(2n - 1)/6n}
\]

\[
\bar{K} = 2K - \frac{1}{n} \sum_{i=1}^{n} e^{\mu_i + \sigma_{xi}(\ln(K) - \mu)} + \frac{\sigma_i^2 - \sigma_{xi}^2/2}{\sigma_x^2}
\]
Appendix B

Referenced Spreadsheets

Naïve Monte Carlo Spreadsheet

The following spreadsheet was used for generating naïve Monte Carlo graphs for this paper. It was modified to calculate the Monte Carlo methods by implementing the principles outlined in this paper.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Average: 9.236</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stock</td>
<td>100</td>
</tr>
<tr>
<td>Strike Price</td>
<td>100</td>
</tr>
<tr>
<td>Time</td>
<td>1</td>
</tr>
<tr>
<td>Interest Rate</td>
<td>0.04</td>
</tr>
<tr>
<td>Dividend</td>
<td>0.1</td>
</tr>
<tr>
<td>Trials</td>
<td>1000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Random #'s</th>
<th>Monte Carlo</th>
<th>Asian Call Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_i = U(0,1)$</td>
<td>$z_i = N^{-1}(u_i)$</td>
<td>$u_i = U(0,1)$</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>1</td>
<td>0.084</td>
<td>-1.376</td>
</tr>
<tr>
<td>2</td>
<td>0.366</td>
<td>-0.343</td>
</tr>
<tr>
<td>3</td>
<td>0.414</td>
<td>-0.207</td>
</tr>
<tr>
<td>4</td>
<td>0.466</td>
<td>-0.085</td>
</tr>
<tr>
<td>5</td>
<td>0.078</td>
<td>-4.148</td>
</tr>
<tr>
<td>6</td>
<td>0.755</td>
<td>0.756</td>
</tr>
<tr>
<td>7</td>
<td>0.949</td>
<td>1.837</td>
</tr>
<tr>
<td>8</td>
<td>0.613</td>
<td>0.303</td>
</tr>
<tr>
<td>9</td>
<td>0.371</td>
<td>-0.329</td>
</tr>
<tr>
<td>10</td>
<td>0.829</td>
<td>0.950</td>
</tr>
<tr>
<td>990</td>
<td>0.345</td>
<td>-0.399</td>
</tr>
<tr>
<td>991</td>
<td>0.613</td>
<td>0.283</td>
</tr>
<tr>
<td>992</td>
<td>0.429</td>
<td>-0.153</td>
</tr>
<tr>
<td>993</td>
<td>0.051</td>
<td>-1.634</td>
</tr>
<tr>
<td>994</td>
<td>0.788</td>
<td>0.790</td>
</tr>
<tr>
<td>995</td>
<td>0.292</td>
<td>-0.546</td>
</tr>
<tr>
<td>996</td>
<td>0.910</td>
<td>1.343</td>
</tr>
<tr>
<td>997</td>
<td>0.496</td>
<td>-0.009</td>
</tr>
<tr>
<td>998</td>
<td>0.481</td>
<td>-0.049</td>
</tr>
<tr>
<td>999</td>
<td>0.372</td>
<td>-0.462</td>
</tr>
<tr>
<td>1000</td>
<td>0.568</td>
<td>0.171</td>
</tr>
</tbody>
</table>
Halton Sequence Extract

The following is an extract of the Halton sequence used to perform quasi-Monte Carlo projections in Example 9 (Dias). Note: The first 15 numbers in the sequence are eliminated to improve sequence uniformity in the higher dimensions, as proven by Galani and Jung, 2007.

<table>
<thead>
<tr>
<th>Trial</th>
<th>Base Dimension 1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>0.0313</td>
<td>0.5926</td>
<td>0.3200</td>
<td>0.3265</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0.5313</td>
<td>0.9259</td>
<td>0.5200</td>
<td>0.4694</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>0.2813</td>
<td>0.0741</td>
<td>0.7200</td>
<td>0.6132</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>0.7813</td>
<td>0.4074</td>
<td>0.9200</td>
<td>0.7561</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0.1563</td>
<td>0.7407</td>
<td>0.1600</td>
<td>0.8980</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>0.6563</td>
<td>0.1852</td>
<td>0.3600</td>
<td>0.0612</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>0.4063</td>
<td>0.5185</td>
<td>0.5600</td>
<td>0.2041</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>0.9063</td>
<td>0.8519</td>
<td>0.7600</td>
<td>0.3469</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>0.9038</td>
<td>0.2963</td>
<td>0.9600</td>
<td>0.4898</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>0.5308</td>
<td>0.6296</td>
<td>0.0080</td>
<td>0.6327</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>0.3438</td>
<td>0.9630</td>
<td>0.2080</td>
<td>0.7755</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>0.8438</td>
<td>0.0123</td>
<td>0.4080</td>
<td>0.9184</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>0.2198</td>
<td>0.3457</td>
<td>0.6080</td>
<td>0.0816</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>0.7188</td>
<td>0.6790</td>
<td>0.8080</td>
<td>0.2245</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>0.4688</td>
<td>0.1235</td>
<td>0.0480</td>
<td>0.3673</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>0.9688</td>
<td>0.4568</td>
<td>0.2480</td>
<td>0.5102</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>0.0156</td>
<td>0.7901</td>
<td>0.4480</td>
<td>0.6531</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>0.5156</td>
<td>0.2346</td>
<td>0.6480</td>
<td>0.7959</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>0.2656</td>
<td>0.5679</td>
<td>0.8480</td>
<td>0.9388</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>0.7656</td>
<td>0.9072</td>
<td>0.0880</td>
<td>0.1030</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>0.1406</td>
<td>0.0494</td>
<td>0.2880</td>
<td>0.2449</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>0.6406</td>
<td>0.3827</td>
<td>0.4880</td>
<td>0.3878</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>0.3906</td>
<td>0.7160</td>
<td>0.6880</td>
<td>0.5306</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>0.8906</td>
<td>0.1605</td>
<td>0.8880</td>
<td>0.6735</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>0.0781</td>
<td>0.4938</td>
<td>0.1280</td>
<td>0.8163</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>0.5781</td>
<td>0.8272</td>
<td>0.3280</td>
<td>0.9902</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>0.3281</td>
<td>0.2716</td>
<td>0.5280</td>
<td>0.1224</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>0.8281</td>
<td>0.6049</td>
<td>0.7280</td>
<td>0.2653</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>0.2301</td>
<td>0.9383</td>
<td>0.9280</td>
<td>0.4082</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>0.7031</td>
<td>0.0864</td>
<td>0.1680</td>
<td>0.5510</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>0.4531</td>
<td>0.4198</td>
<td>0.3680</td>
<td>0.6939</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>0.9531</td>
<td>0.7351</td>
<td>0.5680</td>
<td>0.8367</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>0.0469</td>
<td>0.1975</td>
<td>0.7680</td>
<td>0.9796</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>0.5469</td>
<td>0.5309</td>
<td>0.9680</td>
<td>0.0029</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>0.2969</td>
<td>0.6642</td>
<td>0.0160</td>
<td>0.1408</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>0.7969</td>
<td>0.3086</td>
<td>0.2160</td>
<td>0.2886</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>0.1719</td>
<td>0.6420</td>
<td>0.4160</td>
<td>0.4315</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>0.6719</td>
<td>0.9753</td>
<td>0.6160</td>
<td>0.5743</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>0.4219</td>
<td>0.0247</td>
<td>0.8160</td>
<td>0.7172</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>0.9219</td>
<td>0.3580</td>
<td>0.0560</td>
<td>0.8601</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>0.1094</td>
<td>0.6914</td>
<td>0.2560</td>
<td>0.0233</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>0.6094</td>
<td>0.1358</td>
<td>0.4560</td>
<td>0.1682</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>0.3594</td>
<td>0.4691</td>
<td>0.6560</td>
<td>0.3090</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>0.8594</td>
<td>0.8025</td>
<td>0.8560</td>
<td>0.4519</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>0.2344</td>
<td>0.2469</td>
<td>0.0960</td>
<td>0.5948</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>0.7344</td>
<td>0.5802</td>
<td>0.2960</td>
<td>0.7376</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>0.4844</td>
<td>0.9136</td>
<td>0.4960</td>
<td>0.8805</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>0.9044</td>
<td>0.0617</td>
<td>0.6960</td>
<td>0.0437</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>0.0078</td>
<td>0.3951</td>
<td>0.8960</td>
<td>0.1186</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>0.5078</td>
<td>0.7284</td>
<td>0.1360</td>
<td>0.3294</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix C

Naïve Monte Carlo Simulation Code for Excel VBA

```
Option Explicit
'
Description: Record time for naïve monte carlo trials accurate to nearest hundredth
Sub Naive()

' declare variables
Dim timestart, timeend, elapsed As Double
Dim totalprice As Double, counter As Long, time As Single
Dim a As Double, b As Double, priorprice As Double
Dim n As Long, mcprice As Double, arithcall As Double, trialprice As Double
Dim s1 As Double, s2 As Double, s3 As Double, s4 As Double
Dim t As Double, r As Double, rr As Double, rrr As Double, rrrr As Double
Dim z As Double, zz As Double, zzz As Double, zzzz As Double
Dim s As Double, k As Double, i As Double, v As Double, d As Double
|
' Perform 1000 runs
For counter = 1 To 1000

' Initialize variables
  timestart = 0
  timeend = 0
  elapsed = 0
  trialprice = 0
  totalprice = 0
  priorprice = 0
  mcprice = 0
  a = 0
  b = 0
  timestart = Timer
  mcprice = 0
  priorprice = 0
  n = 0

' Read inputs
  s = Range("b2").Value
  k = Range("b3").Value
  t = Range("b4").Value
  i = Range("b5").Value
  d = Range("b6").Value
  v = Range("b7").Value
  arithcall = Range("e2").Value
```
Do
\[ n = n + 1 \]

Do
\[ r = \text{Rnd()} \]
\[ rr = \text{Rnd()} \]
\[ rrr = \text{Rnd()} \]
\[ rrrr = \text{Rnd()} \]

*\text{Rnd()} can return 0*
Loop Until \( r > 0 \) And \( rr > 0 \) And \( rrr > 0 \) And \( rrrr > 0 \)

\[ z = \text{Application.WorksheetFunction.NormInv}(x, 0, 1) \]
\[ zz = \text{Application.WorksheetFunction.NormInv}(rr, 0, 1) \]
\[ zzz = \text{Application.WorksheetFunction.NormInv}(rrr, 0, 1) \]
\[ zzzz = \text{Application.WorksheetFunction.NormInv}(rrrr, 0, 1) \]

*Calculate stock prices*
\[ s_1 = s \cdot \exp((1 - d - 0.5 \cdot v \cdot v) \cdot t / 4 + v \cdot \sqrt{t / 4} \cdot z) \]
\[ s_2 = s_1 \cdot \exp((1 - d - 0.5 \cdot v \cdot v) \cdot t / 4 + v \cdot \sqrt{t / 4} \cdot z) \]
\[ s_3 = s_2 \cdot \exp((1 - d - 0.5 \cdot v \cdot v) \cdot t / 4 + v \cdot \sqrt{t / 4} \cdot z) \]
\[ s_4 = s_3 \cdot \exp((1 - d - 0.5 \cdot v \cdot v) \cdot t / 4 + v \cdot \sqrt{t / 4} \cdot z) \]

*Calculate trial price and monte carlo price*
\[ \text{trialprice} = \text{Application.WorksheetFunction.Max}(0, (s_1 + s_2 + s_3 + s_4) / 4 - k) \cdot \exp(-r \cdot t) \]
\[ \text{totalprice} = \text{trialprice} + \text{totalprice} \]
\[ \text{priorprice} = \text{mprice} \]
\[ \text{mprice} = (\text{totalprice}) / n \]

*Test convergence stipulations*
\[ a = \text{Abs}(\text{arithmetic} - \text{mprice}) \]
\[ b = \text{Abs}(\text{arithmetic} - \text{priorprice}) \]
Loop Until \( a < 0.01 \) And \( b < 0.01 \)

\[ \text{timeend} = \text{Timer} \]
\[ \text{elapsed} = \text{timeend} - \text{timestart} \]

*Report time and number trials*
If ActiveSheet.Range("W1") = "" Then
\[ \text{Range}("L2") = \text{elapsed} \]
Else
\[ \text{Range}("L" & \text{Rows.Count}).End(xUp).Offset(1) = \text{elapsed} \]
End If

If ActiveSheet.Range("W2") = "" Then
\[ \text{Range}("W2") = n \]
Else
\[ \text{Range}("H" & \text{Rows.Count}).End(xUp).Offset(1) = n \]
End If

Next

End Sub
REFERENCES

Baumann, Nick. "Too Fast to Fail: Is High-Speed Trading the Next Wall Street Disaster?" *Mother Jones*.


Sugiyama, Sam. "Monte Carlo Simulation/Risk Analysis on a Spreadsheet."


ACADEMIC VITA

Stan Korostin
sak5389@gmail.com

Education

B.S. Actuarial Science, 2014, Pennsylvania State University, State College, PA

Honors in Actuarial Science

Thesis: Variance Reduction Techniques for Monte Carlo Valuation of Asian Options

Statistic minor, 2014, Pennsylvania State University, State College, PA

Actuarial Exams and Validation by Educational Experience

Probability, March 2012

Financial Mathematics, December 2012

Models for Financial Economics, July 2013

Models for Life Contingencies, November 2013

Fulfilled VEE requirements for Statistics, Finance, and Economics

Honors and Awards

Smeal College of Business, Risk Management Student Marshal

Beta Gamma Sigma, member, Fall 2012

John Culver Wooddy Scholarship, Fall 2013

CAMAR Scholarship, Fall 2013

Association Memberships/Activities

Actuarial Science Club, President Spring 2013-Fall 2013

Sigma Nu Fraternity, member Fall 2012-Spring 2014

Schreyer Honors College, member Fall 2010-Spring 2014

Professional Experience


Ernst & Young, Insurance and Actuarial Advisory Services Intern, Summer 2013