

**Monte Carlo and Binomial Simulations for
European Option Pricing**

Robert Hon

Computer Science with Mathematics

Session 2012/2013

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student)_____

Summary

This project looks into how options are priced, exploring different methods and investigating how efficient the approximation techniques are.

This project is structured in the following way: Chapter 1 details the aim, objectives, requirements and methodology of the project. Chapter 2 presents the background research done in the area of financial modelling, the concept of financial options and how they are price is introduced. Continued research into option pricing led to the famous Black-Scholes model, which is essential for European options. In Chapter 3, the Monte Carlo approximation method is implemented and compared to the Black-Scholes model. Discrete Euler and Milstein schemes are also researched and implemented. The orders of convergence of the implemented methods are analysed. Chapter 4 explores a more computationally efficient binomial approximation. This method is compared to the Black-Scholes model and in a similar fashion to Chapter 3, the convergence order is analysed. Chapter 5 summarises the findings and compares the models to determine which implementation outperformed other implementations. In Chapter 6, the final conclusions are given and future work pointers are documented.

Contents

Chapter 1 - Introduction	1
1.1 Overview	1
1.2 Aim and Objectives	1
1.3 Minimum Requirements	1
1.4 Methodology	2
1.5 Schedule	3
Chapter 2 - Background Research	4
2.1 Options	4
2.1.1 Call options	4
2.1.2 Different Types of Options	4
2.2 Asset Price Model	5
2.3 Itô's Lemma	5
2.4 The Black-Scholes Model	6
2.4.1 Assumptions of the Black Scholes Model	6
2.4.2 The Black-Scholes Partial Derivative	7
2.4.3 The Black-Scholes Formula for European Options	8
2.5 Approximation Techniques	9
2.6 Time Discretising Techniques	9
Chapter 3 - Monte Carlo Simulation	10
3.1 Monte Carlo Method	10
3.1.1 Convergence	11
3.2 Time Discretising Methods	12
3.2.1 Euler Scheme	13
3.2.2 Milstein Scheme	14
3.2.3 Convergence	16
3.2.4 Runge-Kutta Method	20

Chapter 4 – Binomial Method.....	21
4.1 Valuing the Option	21
4.2 Convergence to the Black-Scholes Model.....	23
Chapter 5 - Evaluation.....	25
5.1 Evaluation of Black-Scholes Model	25
5.2 Evaluation of the Monte Carlo Method	26
5.3 Evaluation of Euler and Milstein Schemes.....	26
5.4 Evaluation of the Binomial Method	26
5.5 Aim and Objectives	28
5.6 Minimum Requirements	29
Chapter 6 - Conclusions	31
6.1 Future Work	31
Bibliography	32
Appendix A – Personal Reflection.....	34
Appendix B – External materials used	36
Appendix C – Ethical Issues.....	37

Chapter 1 - Introduction

1.1 Overview

In a growing option trading financial market, accurate option prices are necessary. There are many different approximation techniques that have been developed through the years. We look into the derivation of these models and analyse which methods are most accurate and computationally efficient.

1.2 Aim and Objectives

The aim of this project is to explore option pricing approximation techniques and how the approximations can be adapted to improve the accuracy.

The objectives of the project are to:

- 1) Research into how options are priced.
- 2) Understand the Black-Scholes equation and adapt it to model price European options.
- 3) Implement a Monte Carlo simulation of the European option.
- 4) Explore different time stepping methods, such as the Euler and Milstein schemes, to improve the accuracy of the approximation.
- 5) Implement the binomial model to approximate the European option.
- 6) Evaluate the accuracy of the implemented models and apply them to American and Asian options.

1.3 Minimum Requirements

- 1) Produce a Black-Scholes model simulation of the European option.
- 2) Produce a Monte Carlo approximation of the European option.
- 3) Produce a Monte Carlo approximation of the European option using the Euler time stepping method.
- 4) Produce a Monte Carlo approximation of the European option using the Milstein time stepping method.
- 5) Produce a binomial approximation of the European option.

Possible Extensions:

- 1) Applying refined models to American and Asian options.
- 2) Produce a Monte Carlo approximation of the European option using the Runge-Kutta time stepping method.

1.4 Methodology

The first stage of the project is to acquire the underlying theory of option pricing. Therefore research is conducted around financial options before models can be derived, discretised and simulated. The research explores the different kinds of options and identifies how they are priced differently. Focussing on European options, the derivation and the importance of the Black-Scholes partial differential equation from Wilmott et al. (1995) became the fundamental basis. Consequently the Black-Scholes model is implemented at this stage. The coding was done in Matlab to reduce the learning overhead of the syntax and GUI of Matlab, as it has been used in COMP2647 Numerical Computation and Visualisation and COMP3920 Parallel Scientific Computing.

Research in approximation techniques, from Glasserman (2003) gave a Monte Carlo approach developed by Boyle (1977). Hence the Monte Carlo simulation is implemented and evaluated at this stage.

Glasserman (2003) also gives the basis of discretisation methods, therefore the Euler and Milstein time stepping method is documented. After understanding the underlying theory the two methods are then implemented at this stage. The two schemes are evaluated by proving the convergence measures documented in Kloeden and Platen (1999).

Building on theory from the taught module, MATH3733 Stochastic Financial Modelling, research into the binomial approximation developed by Cox et al. (1979) is conducted. This method is implemented and evaluated against the exact value of the European call option from the Black-Scholes equation as well as looking at the convergence measure.

Finally the binomial model is adapted to implemented American and Asian options based on methods from Shreve (2005).

1.5 Schedule

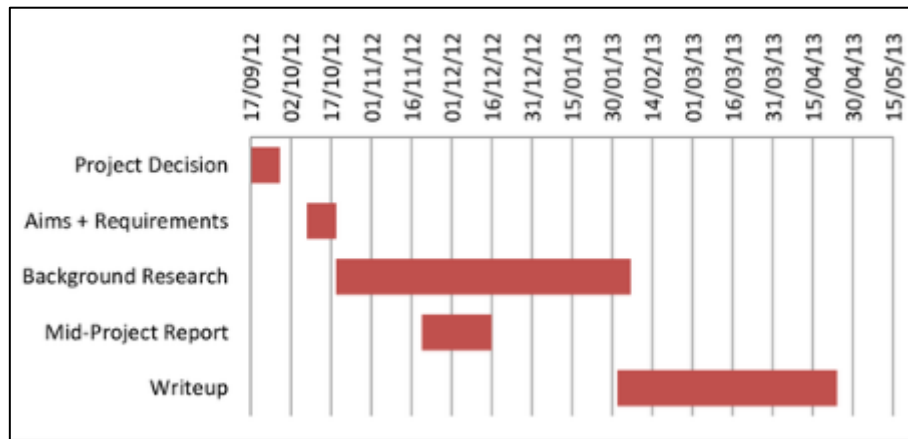


Figure (1.1) The proposed plan at the mid-project report stage.

The plan at the mid-project report stage shown in Figure (1.1) was not detailed as there was not a clear direction of the project due to other work commitments. Consequently after the January examination period, a clear direction of the project was formulated. Therefore the implementation section was added and the different stages detailed accordingly, the on-going testing and evaluation of different implementations was also added as it is a necessity. The revised Gantt chart is shown in Figure (1.2).

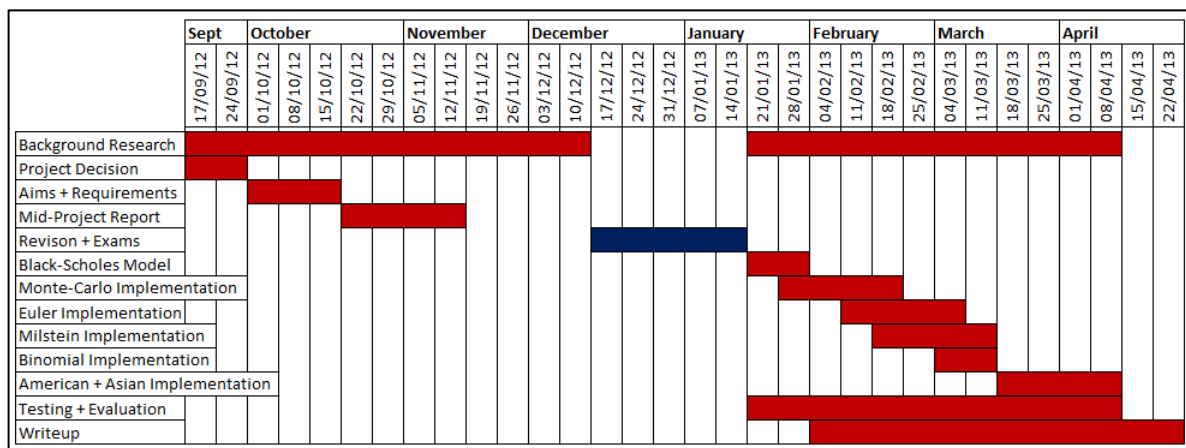


Figure (1.2) A revised Gantt chart outlining the schedule of the project.

Chapter 2 - Background Research

2.1 Options

To classify what a financial option is, we look at the simplest European call option. It is a contract with the following conditions and description from Wilmott et al. (1995):

At a predetermined time T in the future, also known as the time of expiry, the owner of the option may purchase the underlying asset for a predetermined price, also known as the exercise price. The other party of the option is the writer; they have an obligation to sell the asset to the holder if they want to buy it.

In this project we investigate how much the holder should pay for this option, consequently valuing the option.

2.1.1 Call options

Key notation:

S , asset price.

K , exercise price.

T , time of expiry in years.

When $S > K$ at expiry, it would make financial sense to exercise the call option, hence handing over the amount K , obtaining an asset worth S . The profit would be $S - K$. If $S < K$ at expiry, a loss of $K - S$ would be made, therefore the option is worthless. The call option can be written as:

$$C(S, T) = \max(S - K, 0).$$

2.1.2 Different Types of Options

Reviewing papers from Broadie and Detemple (2004), Hobson (2004) and Wilmott et al. (1995), below is a brief summary of the most common option styles:

Plain Vanilla option types:

- European options can only be exercised at the time of maturity.
- American options can be exercised at any time before the time of maturity.

Path dependent options:

- Asian options: The pay-off is calculated from the average price of the asset over a period of time.
- Lookback options: The pay-off is dependent on the maximum price reached by the asset over the lifetime of the option.

Exotic options:

- Bermudan options can only be exercised a set number of times at certain dates.
- Barrier options: The option contract can be triggered when the asset price hits a predetermined value at any time before maturity.
- Digital/Binary options: The pay-off is fixed if the asset crosses the barrier.

2.2 Asset Price Model

The asset price model from Wilmott et al. (1995) states that asset prices must move randomly, therefore:

- The past history is reflected fully in the current price.
- The markets immediately respond to new information on an asset.

Consequently any unanticipated changes on the asset price are classified as a Markov Process. A small time step of dt indicates the change from S to $S + dS$. To model the corresponding return on the asset dS/S , we first use r , a measure of the average growth rate of the asset, known as the drift. This leads to the term

$$r dt.$$

To model the random element of the asset, the term dW is used, which is a random variable from the normal distribution, also known as a Wiener process. σ is a measure of the standard deviation of returns, also known as the volatility. This leads to the contribution

$$\sigma dW.$$

Putting these two results together gives rise to the stochastic differential equation

$$\frac{dS}{S} = r dt + \sigma dW,$$

this equation can be rearranged to give

$$dS = rSdt + \sigma SdW. \quad (2.1)$$

2.3 Itô's Lemma

The following derivation is detailed in 10.2 Hull (1993), B.2 Glasserman (2003), Wilmott et al. (1995) and Kloeden and Platen (1999). Itô's lemma is a key result that is used to manipulate the random variables that are dealt with in this paper.

$$dW^2 \rightarrow dt \text{ as } dt \rightarrow 0. \quad (2.2)$$

The call option $C(S)$ is a function of S . If S is varied by a small amount dS then C will also vary by a small amount dC . From the Taylor series expansion, dC can be written as

$$dC = \frac{dC}{dS} dS + \frac{1}{2} \frac{d^2C}{dS^2} dS^2 + \dots, \quad (2.3)$$

the remaining terms, denoted by the dots, is the remainder which is smaller than the other terms we have retained. Recall Equation (2.1) for dS , dS^2 can be calculated by squaring dS , hence

$$\begin{aligned} dS^2 &= (rSdW + \mu Sdt)^2 \\ &= r^2 S^2 dW^2 + 2r\mu S^2 dt dW + \mu^2 S^2 dt^2. \end{aligned} \quad (2.4)$$

Examining the order of magnitude for the terms in Equation (2.4), (See P29-30 of Wilmott et al. (1995) for order notation). From Equation (2.2) we can see that

$$dW = O(\sqrt{dt}),$$

hence the first term of Equation (2.4) dominates the other terms as it is largest for small dt . This leads to the order

$$dS^2 = r^2 S^2 dW^2 + \dots$$

Since $dW^2 \rightarrow dt$ from Equation (2.2), this becomes

$$dS^2 \rightarrow r^2 S^2 dt. \quad (2.5)$$

By substituting Equation (2.5) into Equation (2.3) and also using the Equation (2.1), we retain terms which are only as large as $O(dt)$, therefore we define Itô's lemma as

$$\begin{aligned} dC &= \frac{dC}{dS} (rSdW + \mu Sdt) + \frac{1}{2} r^2 S^2 \frac{d^2 C}{dS^2} dt \\ &= rS \frac{dC}{dS} dW + \left(\mu S \frac{dC}{dS} + \frac{1}{2} r^2 S^2 \frac{d^2 C}{dS^2} \right) dt. \end{aligned} \quad (2.6)$$

2.4 The Black-Scholes Model

The Black-Scholes model developed by Black and Scholes (1973) is widely accepted to be the standard way to price European options, the following sections define the key properties of the model and how it is adapted to price European options.

2.4.1 Assumptions of the Black Scholes Model

Black and Scholes (1973) stated that the following "ideal conditions" are assumed within the market, the stock and the option:

- 1) The interest rate remains constant through time.
- 2) The stock price follows a Geometric Brownian motion where the drift and volatility remain constant.
- 3) No dividends are paid from the stock.
- 4) The formula is only valid for European options; therefore the option can only be exercised at the time of maturity.
- 5) No transaction costs are charged when buying or selling.
- 6) Any fraction of the price of a security can be borrowed to buy it or to hold it, at the short

term interest rate.

- 7) Short selling does not incur any penalties. The seller will accept the price of a security from a buyer if they do not own a security, and agree to settle with the buyer at some future date by paying him an amount equal to the price of the security on that date.

Therefore the Black-Scholes model is a simplified version of the real world model due to these assumptions.

2.4.2 The Black-Scholes Partial Derivative

From Wilmott et al. (1995), we can use Itô's lemma to specify the random walk the value of the option follows. In this project we are only interested in the call option; hence applying $C(S, t)$ to Itô's lemma defined in Equation (2.6) gives the random walk followed by C

$$dC = \sigma S \frac{\partial C}{\partial S} dW + \left(rS \frac{\partial C}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + \frac{\partial C}{\partial t} \right) dt. \quad (2.7)$$

Constructing a portfolio consisting of a call option and number of the underlying asset

$$\Pi = C - \Delta S. \quad (2.8)$$

The difference of the value of the portfolio in one time-interval is

$$d\Pi = dC - \Delta dS.$$

Applying Equations (2.1), (2.7) and (2.8) together, we can show that Π follows the random walk

$$d\Pi = \sigma S \left(\frac{\partial C}{\partial S} - \Delta \right) dW + \left(rS \frac{\partial C}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + \frac{\partial C}{\partial t} - r\Delta S \right) dt.$$

The Black-Scholes model gives an exact solution; hence the random element of dW in this random walk needs to be eliminated. This is done by substituting

$$\Delta = \frac{\partial C}{\partial S}. \quad (2.9)$$

The increment portfolio becomes completely deterministic with this substitution

$$d\Pi = \left(\frac{1}{2} \sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + \frac{\partial C}{\partial t} \right) dt.$$

The assumption of no transaction costs indicates a growth of $r\Pi dt$ in a time interval of dt , hence

$$r\Pi dt = \left(\frac{1}{2} \sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + \frac{\partial C}{\partial t} \right) dt. \quad (2.10)$$

Substituting (2.8) and (2.9) into (2.10) and dividing by dt , the resulting equation is the Black-Scholes partial differential equation

$$\frac{\partial C}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + rS \frac{\partial C}{\partial S} - rC = 0. \quad (2.11)$$

An alternative derivation is given in Section 6 of Merton (1973).

2.4.3 The Black-Scholes Formula for European Options

The following boundary conditions are defined in Wilmott et al. (1995) for European options to find the unique solution for a European call option.

As we are only interested in the call option in this paper, the payoff of the call option is known at time of expiry when $t = T$ from Section 2.1.1, hence $C(S, T) = \max(S - K, 0)$.

When S is zero, from Equation (2.1), dS will also be zero, therefore the asset price S , can never evolve. Hence $S = 0$ at expiry and there is no payoff, consequently the call option is worthless, therefore $C(0, t) = 0$.

If the asset price increases without a limit, the option will be ever more likely to be exercised, the magnitude of the exercise price becomes negligible as $S \rightarrow \infty$. Hence the call option can be written as $C(S, t) \sim S$ as $S \rightarrow \infty$.

Consequently we put the defined boundary conditions

$$C(S, T) = \max(S - K, 0), \quad C(0, t) = 0, \quad C(S, t) \sim S \text{ as } S \rightarrow \infty,$$

through Equation (2.11), the partial differential equation. The result gives the following explicit solution for the European call:

$$C(S, t) = SN(d_1) - Ke^{-rt}N(d_2), \quad (2.12)$$

where d_1 and d_2 are denoted as

$$d_1 = \frac{\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)t}{\sigma\sqrt{t}},$$

$$d_2 = d_1 - \sigma\sqrt{t} = \frac{\ln\left(\frac{S}{K}\right) + \left(r - \frac{\sigma^2}{2}\right)t}{\sigma\sqrt{t}},$$

and N is the cumulative distribution function

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{1}{2}y^2} dy.$$

The detailed derivation of this equation can be found in Chapter 5 of Wilmott et al. (1995), P225-228 in Lovelock et al. (2007) and Chapter 7 of Björk (2004).

Using Equation (2.12) of the European call, we can compute an option price using this Black-Scholes model; therefore a test case is defined. We set the following initial conditions and model parameters: $S = 250$, $K = 200$, $T = 1$, $r = 0.05$, $\sigma = 0.2$. For the purpose of consistency and validity, these values

will remain the same in consequent chapters to provide a comparison of results between the implementation of different models.

To calculate the price of the call option, these variables applied to Equation (2.12), to price the call option. The result is $C = 61.472088609819394$.

2.5 Approximation Techniques

- Monte Carlo Method – the derivation and implementation of the Monte Carlo method is documented in Section 3.1.
- Binomial Method – the details of the binomial method is documented and implemented in Chapter 4.

2.6 Time Discretising Techniques

- Euler method – the derivation and implementation of the Euler scheme is documented in Section 3.2.1.
- Milstein Scheme – the derivation and implementation of the Milstein scheme is documented in Section 3.2.2.
- Runge-Kutta Methods – the theory of this method is documented in Section 3.2.4.

Chapter 3 - Monte Carlo Simulation

3.1 Monte Carlo Method

Recall that the asset price evolution follows the stochastic differential Equation (2.1). This equation can be solved to give the Monte Carlo simulation developed by Boyle (1977) to simulate asset paths.

$$S(T) = S(0) \exp \left[\left(r - \frac{\sigma^2}{2} \right) T + \sigma W(T) \right]. \quad (3.1)$$

$W(T)$ is a random variable, normally distributed with mean 0 and variance T . From Glasserman (2003), $W(T)$ can be manipulated to become the distribution $\sqrt{T}Z$, where Z is a standard normal random variable with mean 0, and variance 1. Substituting this back into equation (3.1) gives

$$S(T) = S(0) \exp \left[\left(r - \frac{\sigma^2}{2} \right) T + \sigma \sqrt{T} Z \right].$$

To generate the sample paths of the asset, the interval $[0, T]$ is divided into intervals of length dt .

$$S_{t+1} = S_t \exp \left[\left(r - \frac{\sigma^2}{2} \right) dt + \sigma \sqrt{dt} Z \right].$$

This equation, also stated in Rachev and Ruschendorf (1995) is used to produce Algorithm (3.1) for the Monte Carlo simulation of a European call option with the initial conditions and model parameters specified in Section 2.4.3.

```
1  function MonteCarlo {  
2      n → number of paths  
3      m → number of steps  
4      S = 250  K = 200  r = 0.05  sigma = 0.2  T = 1  dt = T/m;  
5      sum = 0;  
6      for i = 1 to n {  
7          S = 250;  
8          for j = 1 to m {  
9              S = S*exp[(r-0.5*sigma^2)*dt + sigma*sqrt(dt)*Z];  
10             }  
11             sum = sum + exp(-r*T)*max(S-K,0);  
12         }  
13         value = sum/n;  
14         return value  
15     }
```

Algorithm (3.1) A Monte Carlo simulation of a European call option.

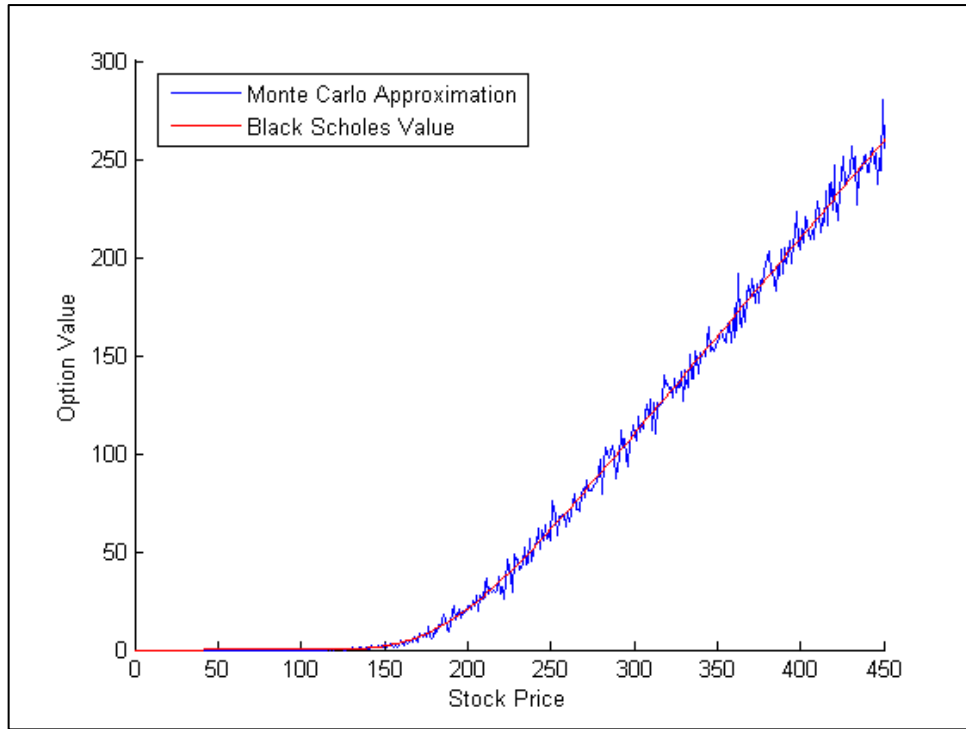


Figure (3.1) A plot showing the Monte Carlo approximation against the Black-Scholes value.

To prove the validity of Algorithm (3.1), Figure (3.1) shows a plot that has been made to compare the Monte Carlo approximation to the exact Black-Scholes value using Equation (2.11). The Monte Carlo approximation is shown to stay in the region of the Black-Scholes equation, with $n = m = 100$. As part of the evaluation we analyse how the approximation varies with different number of paths and time steps in Section 3.1.1. Several simulations with the same parameters must be run in order to get an accurate error measure due to the random variable Z .

3.1.1 Convergence

The Monte Carlo implementation is tested by increasing the number of paths. Table (3.1) shows the results of 3 runs. There is not a definitive convergence from the results but the values are varying less as n increases and converging to the Black-Scholes value of 61.472088609819394.

n	Monte Carlo Approximations		
	Run 1	Run 2	Run 3
100	63.3503394907603	58.4009814589328	56.2415022780091
1000	58.8155617086671	63.0848877179165	61.3960398447864
10000	61.0807704771667	61.0454150957182	61.5249390360755
100000	61.5221276592072	61.4207368011622	61.5062692709526
1000000	61.5002713161830	61.4782137951941	61.4142958435407
10000000	61.4513457572676	61.4817903010193	61.4905538215245

Table (3.1) Monte Carlo approximations with $m = 1000$ and increasing n .

Boyle et al. (1997) and Broadie and Kaya (2006) states that the convergence of the error for a Monte Carlo simulation is $O\left(\frac{1}{\sqrt{n}}\right)$. This means that the error will approximately halve as number of paths quadruples.

Table (3.2) shows the mean of 3 runs. n is quadrupled from its previous simulation for each run. The mean absolute error is calculated by taking the absolute value of the difference between the mean and the exact Black-Scholes value. From the error ratio column, after the paths increase past 1600, it is relatively close to 2. This shows that the error is roughly halving as the number of paths quadruples. Hence the earlier claim that the $error \sim O\left(\frac{1}{\sqrt{n}}\right)$ is proved.

n	Mean	Mean Absolute Error	Error Ratio	Computation Time (s)
100	59.330941076	2.141147534	-	0.0093
400	61.971686953	0.499598344	4.285738	0.0373
1600	61.844874826	0.372786216	1.340174	0.1479
6400	61.311686021	0.160402589	2.324066	0.5616
25600	61.351385277	0.120703333	1.328899	2.3062
102400	61.416574267	0.055514343	2.174273	8.9035
409600	61.504032460	0.031943850	1.737873	35.7370
1638400	61.456494455	0.015594155	2.048450	145.8424
6553600	61.464659185	0.007429425	2.098972	573.2664

Table (3.2) Shows the mean call options and errors as n quadruples.

The computation times in Table (3.2) are also roughly quadrupling as the number of paths is quadrupled as expected. These computation times were recorded in order to make a comparison with the binomial method in Chapter 4.

3.2 Time Discretising Methods

Recalling Equation (2.1) the stochastic differential equation, we simulate the asset during the time interval $[0, T]$, in which the increments, dt , are equally spaced and the size of the dt depends on the number of steps taken. Integrating dS_t from t to $t + dt$ produces

$$S_{t+dt} = S_t + \int_t^{t+dt} \mu(S_u, u) du + \int_t^{t+dt} \sigma(S_u, u) dW_u. \quad (3.2)$$

3.2.1 Euler Scheme

This derivation is from Glasserman (2004) and Kloeden and Platen (1999). The Euler discretisation approximates the two integrals from Equation (3.2). The first integral approximation is

$$\begin{aligned}\int_t^{t+dt} \mu(S_u, u) du &\approx \mu(S_t, t) \int_t^{t+dt} du \\ &= \mu(S_t, t) dt.\end{aligned}$$

The second integral is approximated with the same method as the first integral to produce

$$\begin{aligned}\int_t^{t+dt} \sigma(S_u, u) dW_u &\approx \sigma(S_t, t) \int_t^{t+dt} dW_u \\ &= \sigma(S_t, t) (W_{t+dt} - W_t) \\ &= \sigma(S_t, t) \sqrt{dt} Z.\end{aligned}$$

The Euler discretisation becomes:

$$S_{t+dt} = S_t + \mu(S_t, t) dt + \sigma(S_t, t) \sqrt{dt} Z.$$

The Euler discretisation is applied to the Black-Scholes model where $\mu(S_t, t) = rS_t$ and $\sigma(S_t, t) = \sigma S_t$, substituting these values in yields

$$S_{t+dt} = S_t + rS_t dt + \sigma S_t \sqrt{dt} Z.$$

This equation is used to produce Algorithm (3.2) for the Euler simulation of a European call option with the initial conditions specified.

```

1  function Euler {
2      n → number of paths
3      m → number of steps
4      S = 250  K = 200  r = 0.05  sigma = 0.2  T = 1  dt = T/m;
5      sum = 0;
6      for i = 1 to n {
7          S = 250;
8          for j = 1 to m {
9              S = S + r*S*dt + sigma*S*sqrt(dt)*Z;
10             }
11             sum = sum + exp(-r*T)*max(S-K,0);
12         }
13     value = sum/n;
14     return value
15 }
```

Algorithm (3.2) An Euler scheme simulation of a European call option.

3.2.2 Milstein Scheme

This derivation is from Glasserman (2004) and Kloeden and Platen (1999). Recall from Equation (2.1) that the asset price follows the Wiener process:

$$dS = rSdt + \sigma SdW.$$

For the general case, this stochastic differential equation becomes

$$\begin{aligned} dS_t &= \mu(S_t, t) dt + \sigma(S_t, t) dW_t \\ &= \mu_t dt + \sigma_t dW_t. \end{aligned}$$

This can be rewritten in integral form as

$$S_{t+dt} = S_t + \int_t^{t+dt} \mu_s ds + \int_t^{t+dt} \sigma_s dW_s. \quad (3.3)$$

Applying Itô's lemma defined in Equation (2.6) to functions μ_t and σ_t ,

$$\begin{aligned} d\mu_t &= \left(\mu'_t \mu_t + \frac{1}{2} \mu''_t \sigma_t^2 \right) dt + (\mu'_t \sigma_t) dW_t \\ d\sigma_t &= \left(\sigma'_t \mu_t + \frac{1}{2} \sigma''_t \sigma_t^2 \right) dt + (\sigma'_t \sigma_t) dW_t \end{aligned}$$

The integral form at time s (with $t < s < t + dt$)

$$\begin{aligned} \mu_s &= \mu_t + \int_t^s \left(\mu'_u \mu_u + \frac{1}{2} \mu''_u \sigma_u^2 \right) du + \int_t^s (\mu'_u \sigma_u) dW_u \\ \sigma_s &= \sigma_t + \int_t^s \left(\sigma'_u \mu_u + \frac{1}{2} \sigma''_u \sigma_u^2 \right) du + \int_t^s (\sigma'_u \sigma_u) dW_u \end{aligned}$$

Substituting back into equation (3.3)

$$\begin{aligned} S_{t+dt} &= S_t + \int_t^{t+dt} \left(\mu_t + \int_t^s \left(\mu'_u \mu_u + \frac{1}{2} \mu''_u \sigma_u^2 \right) du + \int_t^s (\mu'_u \sigma_u) dW_u \right) ds \\ &\quad + \int_t^{t+dt} \left(\sigma_t + \int_t^s \left(\sigma'_u \mu_u + \frac{1}{2} \sigma''_u \sigma_u^2 \right) du + \int_t^s (\sigma'_u \sigma_u) dW_u \right) dW_s \end{aligned}$$

Terms of order higher than one are ignored, hence

$$S_{t+dt} = S_t + \mu_t \int_t^{t+dt} ds + \sigma_t \int_t^{t+dt} dW_s + \int_t^{t+dt} \int_t^s (\sigma'_u \sigma_u) dW_u dW_s$$

Applying Euler discretisation and Itô's lemma to the last term gives:

$$\begin{aligned} \int_t^{t+dt} \int_t^s (\sigma'_u \sigma_u) dW_u dW_s &\approx \frac{1}{2} \sigma'_u \sigma_u [(W_{t+dt} - W_t)^2 - dt] \\ &= \frac{1}{2} \sigma'_u \sigma_u [(\Delta W_t)^2 - dt] \\ S_{t+dt} &= S_t + \mu_t \int_t^{t+dt} ds + \sigma_t \int_t^{t+dt} dW_s + \frac{1}{2} \sigma'_u \sigma_u [(\Delta W_t)^2 - dt] \end{aligned}$$

using the same principle from Glasserman (2004) where ΔW_t is equal in distribution to $\sqrt{dt}Z$ and where Z is a standard normal distribution, we arrive at the Milstein scheme stated in Chapter 3 of Kahl and Jackel (2006) which was developed and derived by Milstein (1975)

$$S_{t+dt} = S_t + \mu_t dt + \sigma_t \sqrt{dt} Z + \frac{1}{2} \sigma'_t \sigma_t dt (Z^2 - 1).$$

The following substitutions from Equation (2.1), $\mu(S_t) = rS_t$ and $\sigma(S_t) = \sigma S_t$, of the Black-Scholes model are applied to produce the Milstein scheme for a European call option

$$S_{t+dt} = S_t + rS_t dt + \sigma S_t \sqrt{dt} Z + \frac{1}{2} \sigma^2 dt (Z^2 - 1). \quad (3.4)$$

Equation (3.4) is used to produce Algorithm (3.3) for the Milstein simulation of a European call option with the initial conditions specified.

```

1  function Milstein {
2      n → number of paths
3      m → number of steps
4      S = 250  K = 200  r = 0.05  sigma = 0.2  T = 1  dt = T/m;
5      sum = 0;
6      for i = 1 to n {
7          S = 250;
8          for j = 1 to m {
9              S = S + r*S*dt + sigma*S*sqrt(dt)*Z + 0.5*sigma^2*dt*(Z^2 - 1);
10             }
11             sum = sum + exp(-r*T)*max(S-K,0);
12         }
13         value = sum/n;
14         return value
15     }

```

Algorithm (3.3) A Milstein scheme simulation of a European call option.

The validity of Algorithm (3.2) and (3.3) can be checked by implementing them with a comparison to the exact Black-Scholes values. Figure (3.2) shows this overlay plot of the two schemes and that the approximations stay in close proximity to the Black-Scholes values.

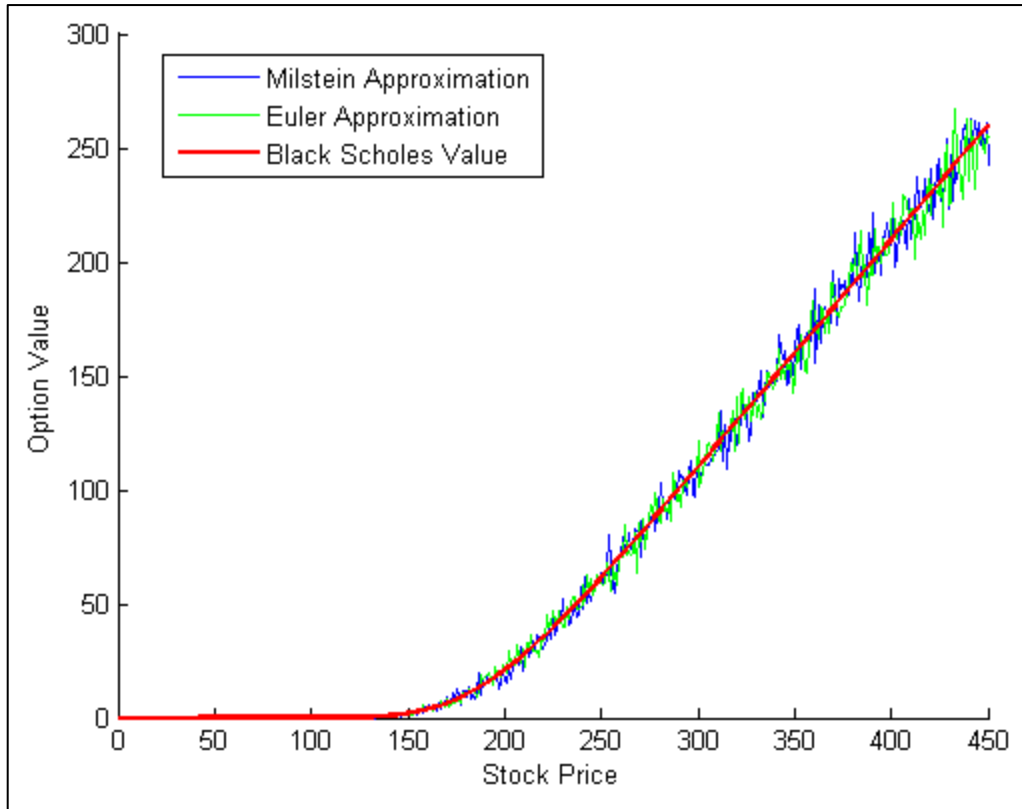


Figure (3.2) Shows a plot of Euler and Milstein approximations to the Black-Scholes value with $n = m = 100$.

3.2.3 Convergence

To measure the accuracy of the Euler and Milstein schemes, we compare them to the solution to the stochastic equation, Equation (2.1), given by the Monte Carlo simulation. We will implement the Euler and Milstein alongside so the asset price undergoes the same variation of the random variable Z . With this method, we can eliminate the sampling error from the results and focus on the time stepping errors and how the two schemes differ.

Consequently Algorithm (3.4) is derived from Equation (3.5) to calculate the mean absolute error of the Euler and Milstein schemes where both schemes use the same random variable Z in each time step. The errors are calculated from the exact solution of the Monte Carlo simulation. To calculate the mean absolute error, Equation (3.5) is used

$$\text{Mean Absolute Error} = \frac{1}{n} \sum_{i=1}^n |SM_i - SA_i|, \quad (3.5)$$

where n denotes the number of paths, SM denotes the exact value from the Monte Carlo approximation and SA denotes the approximation of either the Euler or Milstein scheme.

```

1  function MeanAbsoluteError {
2      n → number of paths
3      m → number of steps
4      S = 250  K = 200  r = 0.05  sigma = 0.2  T = 1  dt = T/m;
5      sum = 0;
6      for i = 1 to n {
7          exact = SE = SM = 250;
8          for j = 1 to m {
9              exact = S*exp[(r-0.5*sigma^2)*dt + sigma*sqrt(dt)*Z]
10             SE = SE + r*SE*dt + sigma*S*sqrt(dt)*Z;
11             SM = SM + r*SM*dt + sigma*SM*sqrt(dt)*Z + 0.5*sigma^2*dt*(Z^2 - 1);
12         }
13         EulerSum = EulerSum + abs(exact - SE)
14         MilsteinSum = MilsteinSum + abs(exact - SM)
15     }
16     EulerError = EulerSum/n
17     MilsteinError = MilsteinSum/n
18     return EulerError, MilsteinError
19 }

```

Algorithm (3.4) Algorithm to compute the mean absolute errors of the Euler and Milstein schemes.

With the mean absolute error calculated, we look into the convergence of the errors. Seydel (2009) defines strong convergence as

$$\epsilon(dt) = E(|SM_T - SA_T^{dt}|) = O(dt^\gamma), \quad (3.6)$$

where ϵ defines the error, SM_T denotes the exact value at time T , SA_T^{dt} denotes the approximation at T with intervals of size dt , and γ denotes the order of convergence where $\gamma > 0$. Chapter 10.2 and 10.3 of Kloeden and Platen (1999) shows that the Euler scheme has a strong convergence 0.5 and the Milstein scheme has a strong convergence 1.

As a result we can compute the convergence order of the Euler scheme using Equation (3.6)

$$O\left(dt^{\frac{1}{2}}\right) = O\left(\frac{\sqrt{T}}{\sqrt{m}}\right) = O\left(\frac{1}{\sqrt{m}}\right).$$

Therefore the error of the Euler scheme can be expressed as

$$|Euler\ Error| \propto \frac{1}{\sqrt{m}} = k \frac{1}{\sqrt{m}}, \quad (3.7)$$

where k is the proportionality constant.

In a similar fashion, the convergence order of the Milstein scheme is also computed using Equation (3.6)

$$O(dt^1) = O\left(\frac{T}{m}\right) = O\left(\frac{1}{m}\right).$$

Therefore the error of the Milstein scheme can be expressed as

$$|Milstein Error| \propto \frac{1}{m} = k \frac{1}{m}, \quad (3.8)$$

where k is the proportionality constant.

Algorithm (3.4) is implemented and the steps are quadrupled with each simulation in the same fashion as the Monte Carlo simulation as shown in Table (3.2). Each variation of steps is repeated 3 times and the mean error is calculated for both schemes.

Euler Approximation		
m	Mean Error	Ratio
100	0.5884380688	2.01552
400	0.2919535652	1.95665
1600	0.1492108539	2.01565
6400	0.0740261141	1.98783
25600	0.0372396102	1.96128
102400	0.0189874479	2.09566
409600	0.0090603765	1.94111
1638400	0.0046676360	2.01072
6553600	0.0023213774	-

Table (3.3) A table of the mean absolute errors of the Euler scheme

From Table (3.3), the ratio column indicates that as the steps quadruple, the error halves.

Milstein Approximation		
m	Mean Error	Ratio
100	0.0220310669	4.06401
400	0.0054210152	3.89562
1600	0.0013915666	4.04384
6400	0.0003441203	3.87866
25600	0.0000887215	4.20694
102400	0.0000210893	3.82297
409600	0.0000055165	4.09904
1638400	0.0000013458	4.01179
6553600	0.0000003355	-

Table (3.4) A table of mean absolute errors of the Milstein scheme

From Table (3.4), the ratio column indicates that as the steps quadruple, the error is a quarter of its previous simulation. The results from Tables (3.3) and (3.4) are promising and are showing convergence.

Euler Approximation			
m	Mean Error	$\frac{6}{\sqrt{m}}$	$\frac{ Error }{6/\sqrt{m}}$
100	0.5884380688	0.60000000	0.9807301
400	0.2919535652	0.30000000	0.9731786
1600	0.1492108539	0.15000000	0.9947390
6400	0.0740261141	0.07500000	0.9870149
25600	0.0372396102	0.03750000	0.9930563
102400	0.0189874479	0.01875000	1.0126639
409600	0.0090603765	0.00937500	0.9664402
1638400	0.0046676360	0.00468750	0.9957623
6553600	0.0023213774	0.00234375	0.9904544

Table (3.5) The error is compared to $\frac{6}{\sqrt{m}}$ with a ratio comparison.

Milstein Approximation			
m	Mean Error	$\frac{2}{m}$	$\frac{ Error }{2/m}$
100	0.0220310669	0.0200000000	1.1015533
400	0.0054210152	0.0050000000	1.0842030
1600	0.0013915666	0.0012500000	1.1132533
6400	0.0003441203	0.0003125000	1.1011850
25600	0.0000887215	0.0000781250	1.1356352
102400	0.0000210893	0.0000195313	1.0797735
409600	0.0000055165	0.0000048828	1.1297731
1638400	0.0000013458	0.0000012207	1.1024757
6553600	0.0000003355	0.0000003052	1.0992345

Table (3.6) The error is compared to $\frac{2}{m}$ with a ratio comparison.

For the Euler scheme, Table (3.5) shows the ratio of the error and $\frac{6}{\sqrt{m}}$ to be close to one. From this we can deduce that the proportionality constant k of Equation (3.7) is 6, hence

$$|Euler Error| \approx 6 \frac{1}{\sqrt{m}}.$$

Similarly for the Milstein Scheme, Table (3.6) shows the ratio of the error and $\frac{2}{m}$ to be close to one.

From this we can deduce that the proportionality constant k of Equation (3.8) is 2, hence

$$|Milstein Error| \approx 2 \frac{1}{m}.$$

From these results, the Milstein scheme is proved to be an improvement of the Euler scheme by outperforming the Euler scheme in each simulation. It is also proved to have a higher order of convergence as shown from the results in Tables (3.5) and (3.6).

3.2.4 Runge-Kutta Method

Another approach is the Runge-Kutta method. Seydel (2009) gives the following approximation for the Runge-Kutta method

$$\bar{S} = S_i + rdt + \sigma\sqrt{dt}$$
$$S_{i+1} = S_i + rdt + \sigma\sqrt{dt}Z + \frac{1}{2\sqrt{dt}}(dtZ^2 - dt)[\sigma(\bar{S}) - \sigma(S)].$$

This method was not implemented due to the time constraints of the project.

Chapter 4 – Binomial Method

4.1 Valuing the Option

Cox et al. (1979), Hull (1993) and Pliska (1997) gives the following basis and derivation of the binomial model. A stock price is valued as S . The binomial model indicates that the stock price moves with probability p to S_u and to S_d with probability $1 - p$. This occurs in a small time interval dt shown in Figure (4.1). S_u denotes an increase in the stock price S with proportion u and S_d denotes a decrease in the stock price S with proportion d .

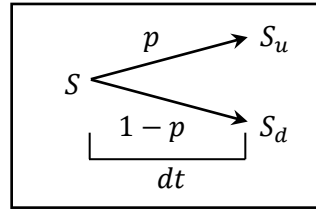


Figure (4.1) Shows the possible paths that the stock might evolve to in one time step.

Hence the call options can be defined as

$$C_u = \max(S_u - K, 0),$$

$$C_d = \max(S_d - K, 0).$$

A portfolio can be constructed, the portfolio will cost $\Delta S + B$. Where Δ is the number of shares of stock and the amount of riskless bonds, B . Consequently, the portfolio will cost

$$\Delta uS + e^{r dt} B = C_u, \quad (4.1)$$

$$\Delta dS + e^{r dt} B = C_d. \quad (4.2)$$

Equations (4.1) and (4.2) are solved to find Δ and B

$$\Delta = \frac{C_u - C_d}{(u - d)S}, \quad B = \frac{uC_d - dC_u}{(u - d)e^{r dt}}.$$

Substituting Δ and B back into the call option

$$\begin{aligned} C &= \Delta S + B \\ &= \frac{C_u - C_d}{(u - d)S} + \frac{uC_d - dC_u}{(u - d)e^{r dt}} \\ &= \frac{\left[\left(\frac{e^{r dt} - d}{u - d} \right) C_u + \left(\frac{u - e^{r dt}}{u - d} \right) C_d \right]}{e^{r dt}}. \end{aligned}$$

The coefficients of C_u and C_d correspond to the probability measures in Figure (4.1), hence the following substitutions can be made

$$p = \left(\frac{e^{r dt} - d}{u - d} \right) \text{ and } 1 - p = \left(\frac{u - e^{r dt}}{u - d} \right),$$

the European call option can now be written as

$$C = \frac{[pC_u + (1 - p)C_d]}{e^{r\Delta t}}. \quad (4.3)$$

Equation (4.3) forms the fundamental method to construct the binomial tree. Consider Figure (4.2), a binomial tree with two periods of call options and shows the three possibilities which the call option could evolve to in two time steps of Δt .

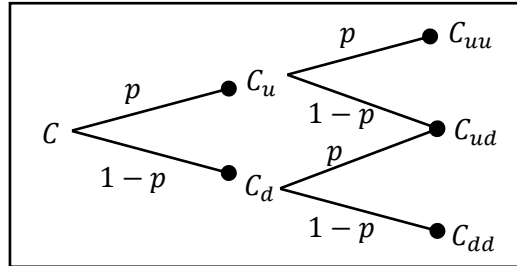


Figure (4.2) A binomial tree with two periods of call options.

From Figure 4.2, it can be shown that C_u and C_d can be evaluated using the same method as Equation (4.3). Therefore C in Figure (4.2) can be evaluated using C_u and C_d defined in Equations (4.4) and (4.5) respectively

$$C_u = \frac{[pC_{uu} + (1 - p)C_{ud}]}{e^{r\Delta t}}, \quad (4.4)$$

$$C_d = \frac{[pC_{ud} + (1 - p)C_{dd}]}{e^{r\Delta t}}. \quad (4.5)$$

Using the same principle that was used to evaluate Equations (4.4) and (4.5), Algorithm (4.1) can be produced to evaluate the binomial approximation for a European call for any predetermined number of periods.

Algorithm (4.1) evaluates all the values at the time of expiry T where the call options only depend on the number of steps up and down. The tree is then traversed backwards to work out the rest of the call values before returning the final call value approximation C .

```

1  function Binomial {
2      m → number of steps
3      S = 250  K = 200  r = 0.05  sigma = 0.2  T = 1  dt = T/m;
4      u = exp(sigma*sqrt(dt));
5      d = 1/u;
6      p = (exp(r*dt)-d)/(u-d);
7      for i = 1 to m+1 {
8          S(i) = S*(d^(m+1-i))*(u^(i-1));
9          C(i) = max(S(i)-K, 0);
10     }
11     for j = m to 1 step -1 {
12         for i = 1 to j {
13             C(i) = (p*C(i+1)+(1-p)*C(i))/exp(r*dt);
14         }
15     }
16     value = C(1);
17     return value
18 }

```

Algorithm (4.1) A binomial simulation for European call options.

4.2 Convergence to the Black-Scholes Model

Qu (2010) gives a direct proof that the binomial model converges to the Black-Scholes model and the rate of convergence of the error is found to be $\frac{1}{m}$ Chang and Palmer (2007). Figure (4.3) shows how the approximation converges to the exact Black-Scholes over 100 steps.

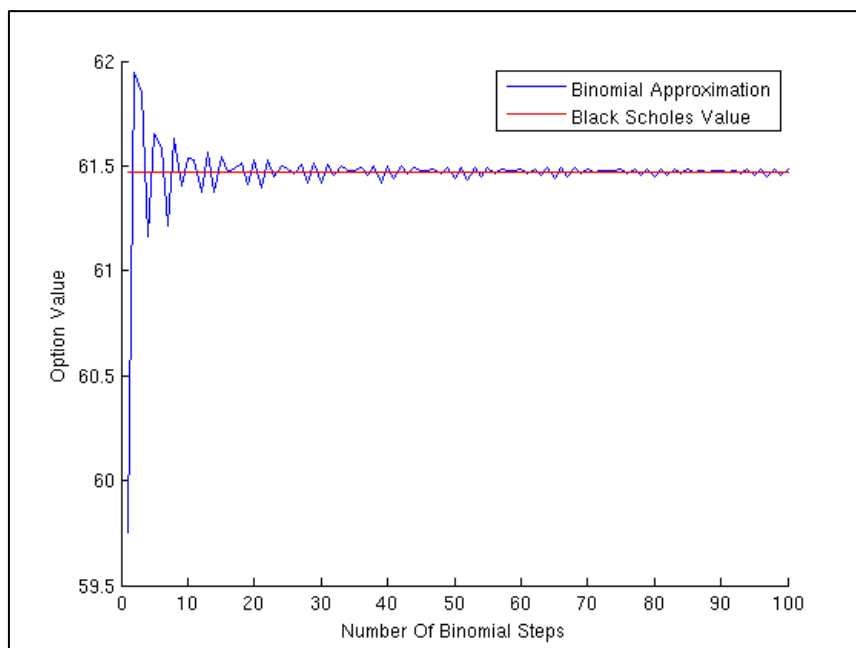


Figure (4.3) A graph showing convergence to the exact Black Scholes value as steps increases.

The absolute error is computed at each simulation of m with

$$\text{Absolute Error} = |(\text{Binomial Approximation}) - (\text{Black Scholes Solution})|,$$

where the Black-Scholes solution is 61.472088609819394, recalling from Section 2.4.3.

m	Binomial Value	Error	$\frac{1}{m}$	$\frac{ Error }{1/m}$	Computation Time (s)
10	61.536162042336336	0.0640734325169987	0.1	1.5607092685954	0.00229
100	61.483739749246872	0.0116511394274994	0.01	0.8582851541882	0.002098
500	61.474458535456925	0.0023699256375949	0.002	0.8439083354656	0.019553
1000	61.473044256421915	0.0009556466025984	0.001	1.0464119239068	0.064034
2500	61.471734930580652	0.0003536792386996	0.0004	1.1309682792543	0.370574
5000	61.472320146750931	0.0002315369316008	0.0002	0.8637930831043	1.510842

Table (4.1) The table shows that the error is approximate to $1/m$.

Table (4.1) shows the test values from implementing the binomial method. The computational time required for the binomial method is significantly less than the schemes implemented in Chapter 3.

Chapter 5 - Evaluation

To thoroughly evaluate all the models we have implemented, the evaluation criteria detailed below is followed:

- Black-Scholes model
 - We evaluate our implementation with at least one other source to ensure that the values are consistent across different sources.
- Monte Carlo method
 - Evaluate approximations against the verified Black-Scholes model to ensure the approximation is correct.
 - Prove the order of convergence derived from Boyle et al. (1997) and Broadie and Kaya (2006) is $O\left(\frac{1}{\sqrt{n}}\right)$, to ensure that the implementation verifies the claims from the literature.
- Euler scheme and Milstein scheme
 - Evaluate approximation values from both schemes against the verified Black-Scholes model to ensure the approximation is correct.
 - Prove the order of convergence derived from Kloeden and Platen (1999) is $O(\sqrt{dt})$ for the Euler scheme and $O(dt)$ for the Milstein Scheme to ensure that the implementation verifies the claims from the literature.
- Binomial method
 - Evaluate with another verified source of implementation to ensure that the values are consistent across different sources.

5.1 Evaluation of Black-Scholes Model

The Black-Scholes model implemented in 2.4.3 is evaluated against another solution to check its validity. Recall that the exact value computed was 61.472088609819394.

http://www.soarcorp.com/black_scholes_calculator.jsp returns 61.472.

<http://www.tradingtoday.com/black-scholes> returns 61.4721.

The computed value was accurate to 15 decimal places, whereas the previous two online solutions are only accurate to 3 and 4 decimal places respectively. Therefore I searched for an Excel implementation where I could manipulate the output. As shown below, the result matches the computed solution up to 13 decimal places.

www.alumni.hbs.edu/new_alumni/toolkit/tools/blackscholes.xls returns 61.4720886098194.

5.2 Evaluation of the Monte Carlo Method

The validity of the implementation was tested by plotting approximations against the Black-Scholes value, it is shown in Figure (3.1) that the approximations stay close to the Black-Scholes values hence proving the validity of the implemented method.

Reviewing literature from Boyle et al. (1997) and Broadie and Kaya (2006), the convergence was found to be $O\left(\frac{1}{\sqrt{n}}\right)$. This was proved from the ratio comparisons in Table (3.2)

5.3 Evaluation of Euler and Milstein Schemes

The validity of the Euler and Milstein schemes is tested by plotting approximations against the Black-Scholes value, it is shown in Figure (3.2) that the approximations stay close to the Black-Scholes values hence proving the validity of the two schemes.

To eliminate the random sampling error from the Monte Carlo simulation, the Euler and Milstein schemes were run alongside the Monte Carlo method using the same random element in each step, this meant the two schemes can be compared directly. The Milstein scheme was shown to outperform the Euler scheme as it had an error convergence of $O(dt)$ compared to the Euler scheme which had an error convergence $O(\sqrt{dt})$ as shown in Section 3.2.3. This was due to the fact that the Milstein scheme expanded terms to $O(dt)$ as shown in Glasserman (2004) which improves the convergence order.

5.4 Evaluation of the Binomial Method

To evaluate the binomial method that was implemented in Chapter 4, we check the computed values are verified with another solution. Using an online source:

<http://www.mngt.waikato.ac.nz/kurt/frontpage/ModelsAcademic/ExcelModels/Binomial%20Option%20Model.xls>

We compare the tree values computed from the implemented program to the tree values that are calculated using this Excel spread sheet implementation.

Period	Binomial Option Values
0	61.53616204
1	75.83519609
2	91.75695279
3	109.14324412
4	127.87598559
5	147.92369309
6	169.33892409
7	192.21125597
8	216.63611590
9	242.71516284
10	270.55669437

Table (5.1) Option values from the top node of the binomial tree for each period.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Binomial Option Pricing Model: Visualizing the Tree											
2	S (S_0)	\$ 250.00	current share price			Test with CRRBinomial function				B&S Formula		
3	r (rf)	5%	riskfree rate p.a.			CallPutFEUR_Amer_Flag						
4	σ (sigma)	20%				c	e	61.5362	61.4721			
5	u	1.065	u = EXP(σ*ΔT^0.5)							1.71798		
6	d	0.939	d = EXP(-σ*ΔT^0.5)			p	e	1.7820				
7	X	\$ 200.00										
8	T	1				c	a	61.5362				
9	n	10										
10	ΔT	0.1	ΔT = T / n			p	a	1.8825				
11	p	0.5238	p = [exp(rΔT) - d] / (u - d)									
12	European Call											
13												
14												
15	t	0.0000	0.1000	0.2000	0.3000	0.4000	0.5000	0.6000	0.7000	0.8000	0.9000	1.0000
16	n	0.00	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00
18	S (\$)	250.00	266.32	283.71	302.23	321.97	342.99	365.38	389.23	414.65	441.72	470.56
19	Option price	61.5362	75.84	91.76	109.14	127.88	147.92	169.34	192.21	216.64	242.72	270.56
20	1		234.68	250.00	266.32	283.71	302.23	321.97	342.99	365.38	389.23	414.65
21			46.46	59.12	73.60	89.69	107.17	125.93	145.96	167.37	190.23	214.65
22	2			220.30	234.68	250.00	266.32	283.71	302.23	321.97	342.99	365.38
23				33.01	43.82	56.68	71.40	87.67	105.21	123.96	143.98	165.38
24	3				206.79	220.30	234.68	250.00	266.32	283.71	302.23	321.97
25					21.48	30.13	41.08	54.26	69.30	85.70	103.23	121.97
26	4					194.12	206.79	220.30	234.68	250.00	266.32	283.71
27						12.19	18.41	27.02	38.28	51.99	67.32	83.71
28	5						182.22	194.12	206.79	220.30	234.68	250.00
29							5.47	9.13	14.91	23.61	35.68	50.00
30	6							171.06	182.22	194.12	206.79	220.30
31								1.50	2.87	5.51	10.58	20.30
32	7								160.57	171.06	182.22	194.12
33									0.00	0.00	0.00	0.00
34	8									150.73	160.57	171.06
35										0.00	0.00	0.00
36	9										141.49	150.73
37											0.00	0.00

Figure (5.1) An Excel implementation of the binomial method with the same model parameters defined in this paper.

The original outputs are only accurate to 4 decimal places for the final approximation and 2 decimal points for the remaining nodes. Hence in a similar fashion to Section 5.1, the output is modified to match the level of accuracy of Table (5.1).

Period	Option Values From Online Source
0	61.53616204
1	75.83519609
2	91.75695279
3	109.14324412
4	127.87598559
5	147.92369309
6	169.33892409
7	192.21125597
8	216.63611590
9	242.71516284
10	270.55669437

Table (5.2) Modified outputs from online source.

The option values computed from the binomial implementation shown in Table (5.1), correspond with the values in row 19 of Figure (5.1), where the level of accuracy of the outputs were increased, as shown in Table (5.2).

To achieve the same level of accuracy as the Monte Carlo method, the binomial method computation time required is significantly less than the Monte Carlo method. Recall from Table (3.2), the most accurate simulation of the Monte Carlo method took approximately 573 seconds, with an error of 0.007429425. Compared with a simulation of 2500 steps with the binomial method in Table (4.1), the computation time was 0.370574 seconds with an error of 0.0003536792386996. The binomial method is much quicker and achieves a greater accuracy compared to the Monte Carlo method.

5.5 Aim and Objectives

The project aim from Section 1.2 was achieved as the Monte Carlo approximation technique was implemented in Section 3.1. The approximations were improved with the Euler and Milstein time stepping methods implemented in Section 3.2. The binomial method, a different approach to improve the approximations was implemented in Chapter 4.

The objectives from Section 1.2 were all achieved apart from the last one and are documented in the following sections:

- 1) Chapter 2.
- 2) Section 2.4.
- 3) Section 3.1.
- 4) Section 3.2.1 and Section 3.2.2.
- 5) Chapter 4.
- 6) The accuracy of the implemented models are documented in Sections 3.1.1, 3.2.3 and 4.2. They were not applied to American and Asian options due to time constraints but a starting point is documented in Section 6.1.

5.6 Minimum Requirements

- 1) *Produce a Black-Scholes model simulation of the European option.*

The Black-Scholes partial derivative from Black and Scholes (1973) was documented and derived in Section 2.4.2. This was necessary in order to apply boundary conditions of a European call option so a Black-Scholes model can be derived. This model was implemented to calculate the exact Black-Scholes value for our European call option with the initial model parameters as shown in Section 2.4.3.

- 2) *Produce a Monte Carlo approximation of the European option.*

The Monte Carlo approach was derived and implemented in Section 3.1.

- 3) *Produce a Monte Carlo approximation of the European option using the Euler time stepping method.*

The Euler scheme was documented and implemented in Section 3.2.1 with a direct comparison to the Milstein scheme, the convergence measure of the error is derived and shown to be $O(\sqrt{dt})$ in Section 3.2.3.

- 4) *Produce a Monte Carlo approximation of the European option using the Milstein time stepping method.*

The Milstein scheme was documented and implemented in Section 3.2.2 with a direct comparison to the Euler scheme, the convergence measure of the error is derived and shown to be $O(dt)$ in Section 3.2.3.

- 5) *Produce a binomial approximation of the European option.*

The binomial method was derived and implemented in Chapter 4. The convergence was shown to be $O\left(\frac{1}{m}\right)$ and it was also proved to be computationally more efficient, as it

achieved a better level of accuracy with significantly less computation time than the Monte Carlo method shown in section 5.4.

Possible Extensions:

1) *Applying refined models to American and Asian options.*

Unfortunately due to time constraints, the implemented models were not refined for American and Asian options, but the research and a starting point is documented in Section 6.1.

2) *Produce a Monte Carlo approximation of the European option using the Runge-Kutta time stepping method.*

Unfortunately due to time constraints, this method was not implemented, but the research is documented in Section 3.2.4.

Chapter 6 - Conclusions

The project successfully analysed option pricing methods by implementing the initial Black-Scholes model. The Monte Carlo approximation was also implemented and verified with the Black-Scholes model. Euler and Milstein discretisation methods were implemented and critically analysed by comparing their convergence orders.

The binomial approach was different and was proved to require much less computation time to the same level of accuracy as the previous methods.

6.1 Future Work

American and Asian options

Due to time constraints, these options were not implemented. However, these methods were encountered during the background research for this project. Shreve (2005) and Nelson and Ramaswamy (1990) documents the basis and derivation of pricing American options using the binomial method. Section 6.3 of Seydel (2009) derives the basis of the Asian option.

Antithetic Variates

Seydel (2009) also documents a method which requires little extra computation. With each Monte Carlo simulation, the random variable $-Z$ is also used to calculate the mirror image of the simulation path, known as the “antithetic variate”. It is shown to have smoother convergence, hence research into this area could prove interesting.

Parallelising Implementations

As shown from Table (3.2), computational times can reach up to 500 seconds. Building on MPI material taught in COMP3920 Parallel Scientific Computing, the methods could be parallelised by splitting the simulation of paths between processors, and then collating all the simulation results and calculating the mean.

Bibliography

- BJÖRK, T. 2004. *Arbitrage theory in continuous time*, Oxford, Oxford University Press.
- BLACK, F. & SCHOLES, M. 1973. Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81, 637-654.
- BOYLE, P., BROADIE, M. & GLASSERMAN, P. 1997. Monte Carlo methods for security pricing. *Journal of Economic Dynamics & Control*, 21, 1267-1321.
- BOYLE, P. P. 1977. Options - Monte-Carlo Approach. *Journal of Financial Economics*, 4, 323-338.
- BROADIE, M. & DETEMPLE, J. B. 2004. Option pricing: Valuation models and applications. *Management Science*, 50, 1145-1177.
- BROADIE, M. & KAYA, O. 2006. Exact simulation of stochastic volatility and other affine jump diffusion processes. *Operations Research*, 54, 217-231.
- CHANG, L. B. & PALMER, K. 2007. Smooth convergence in the binomial model. *Finance and Stochastics*, 11, 91-105.
- COX, J. C., ROSS, S. A. & RUBINSTEIN, M. 1979. Option Pricing - Simplified Approach. *Journal of Financial Economics*, 7, 229-263.
- GLASSERMAN, P. 2003. *Monte Carlo methods in financial engineering*, New York ; London, Springer.
- HOBSON, D. 2004. Review Paper. A Survey of Mathematical Finance. *Proceedings: Mathematical, Physical and Engineering Sciences*, 460, 3369-3401.
- HULL, J. 1993. *Options, Futures, and Other Derivative Securities*, Prentice Hall.
- KAHL, C. & JACKEL, P. 2006. Fast strong approximation Monte Carlo schemes for stochastic volatility models. *Quantitative Finance*, 6, 513-536.
- KLOEDEN, P. E. & PLATEN, E. 1999. *Numerical solution of stochastic differential equations*, Berlin ; New York, Springer.
- LOVELOCK, D., MENDEL, M. & WRIGHT, A. L. 2007. *An introduction to the mathematics of money : saving and investing*, New York, Springer.
- MERTON, R. C. 1973. Theory of Rational Option Pricing. *Bell Journal of Economics*, 4, 141-183.
- MILSTEIN, G. N. 1975. Approximate Integration of Stochastic Differential Equations. *Theory of Probability & Its Applications*, 19, 557-562.
- NELSON, D. B. & RAMASWAMY, K. 1990. Simple Binomial Processes as Diffusion Approximations in Financial Models. *Review of Financial Studies*, 3, 393-430.
- PLISKA, S. R. 1997. *Introduction to mathematical finance : discrete time models*, Malden, Mass. ; Oxford, Blackwell.

- QU, X. G. 2010. A Direct Justification of the Binomial Pricing Model as an Approximation of the Black-Scholes Formula. *Pakistan Journal of Statistics*, 26, 187-193.
- RACHEV, S. T. & RUSCHENDORF, L. 1995. Models for Option Prices. *Theory of Probability and Its Applications*, 39, 120-152.
- SEYDEL, R. 2009. *Tools for computational finance*, Berlin, Springer.
- SHREVE, S. E. 2005. *Stochastic calculus for finance 1. Binomial asset pricing model*, New York, Springer.
- WILMOTT, P., HOWISON, S. & DEWYNNE, J. 1995. *The mathematics of financial derivatives : a student introduction*, Cambridge, Cambridge University Press.

Appendix A – Personal Reflection

This final year project proved to be an extremely challenging yet rewarding experience. It is key to choose a subject of genuine interest for the student.

This project built on theory I had learnt in my MATH3733 Stochastic Financial Modelling module. Therefore I took on the challenge to simulate models as I had a good base of the theory and the coding knowledge to implement these models.

The project unintentionally took a back seat during the first semester due to studying 4 modules at the same time; the workload was much higher than expected and was not managed well. As a result, background research was not done thoroughly and details were lacking, this led to a poor mid-project report which did not have a clear direction. I would advise students have a good mid-project report put together at this stage as it will be one of the rare times you will get invaluable feedback from your assessor. This is something I regret that I did not make better use of.

Due to the poor foundations laid in the first semester, this meant the second semester was extremely tough. At the start of the second semester the project took a more focussed direction into option pricing from further research and the Monte Carlo approximation model was implemented. This gave the basis to expand my implementation to Euler and Milstein schemes and a different binomial approach.

Due to unforeseen errors and bugs from the previous implementations, the original plan to implement American and Asian options was not achieved due to time constraints. This could have been avoided by conducting more thorough research in the first semester and having at least one model implemented by the mid-project report stage.

I would advise any student who tackles any project to have a contingency plan in their schedule, such as setting your own deadline to finish a week before the actual deadline. This will give breathing space in the last week to touch up any minor changes instead of rushing the remaining sections. Furthermore, expect things to go wrong, not everything will go to plan or work as you think it would, so always backup your files. My write up was hampered when my references all changed to completely irrelevant ones which corrupted the whole file; I lost half of my report due to this

unforeseen problem. Thankfully, I had made periodic backups which meant I was able to recover what was lost very quickly.

I also advise students who tackle Monte Carlo simulations to look into parallelising the method as computation times were reaching over 500 seconds. This could easily be sped up by splitting the simulation of paths between processors, and then collating all the simulation results and calculating the mean.

Appendix B – External materials used

Lee Pei Yuen's past project on Multilevel Monte Carlo Simulation for Options Pricing was useful as it produces some of the same solutions which I have implemented. It allowed me to verify that my implementation was correct, and provided some helpful research direction pointers.

Appendix C – Ethical Issues

There were no ethical issues considered for this project.