

Multilevel Monte Carlo Simulation in Options Pricing

Funmilayo Eniola Kazeem



UNIVERSITY of the

WESTERN CAPE

A thesis submitted in partial fulfilment of the requirements for
the degree Magister Scientiae in the Department of Mathematics
and Applied Mathematics, University of the Western Cape.

Supervisor: Prof. Kailash C. Patidar

Co-supervisor: Dr. Raouf Ghomrasni

November 2014

Multilevel Monte Carlo

Simulation in Option Pricing

Funmilayo Eniola Kazeem



KEYWORDS

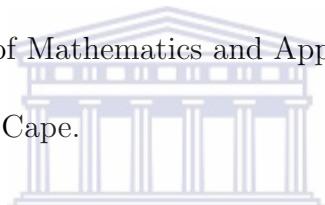
- Geometric Brownian Motion
- Monte Carlo simulation
- Multilevel Monte Carlo method
- Option pricing
- Computational cost
- Risk-neutral measure
- Quasi-Monte Carlo method
- Least squares Monte Carlo method
- Euler discretisation
- Milstein discretisation

Abstract

Multilevel Monte Carlo Simulation in Option Pricing

Funmilayo Eniola Kazeem

MSc Thesis, Department of Mathematics and Applied Mathematics,
University of the Western Cape.



In Monte Carlo path simulations, which are used extensively in computational finance, one is interested in the expected value of a quantity which is a functional of the solution to a stochastic differential equation [M.B. Giles, Multilevel Monte Carlo Path Simulation: Operations Research, **56(3)** (2008) 607-617] where we have a scalar function with a uniform Lipschitz bound. Normally, we discretise the stochastic differential equation numerically. The simplest estimate for this expected value is the mean of the payoff (the value of an option at the terminal period) values from N independent path simulations. The multilevel Monte Carlo path simulation method recently introduced by Giles exploits strong convergence properties to improve the computational complexity by combining simulations with different levels of resolution. This new method improves on the computational complexity of the standard Monte Carlo approach by considering Monte Carlo simulations with a geometric sequence of different timesteps following the approach of Kebaier [A.

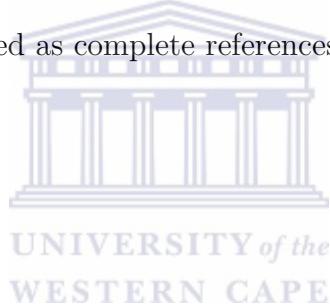
Kebaier, Statistical Romberg extrapolation: A new variance reduction method and applications to options pricing. *Annals of Applied Probability* **14(4)** (2005) 2681-2705]. The multilevel method makes computation easy as it estimates each of the terms of the estimate independently (as opposed to the Monte Carlo method) such that the computational complexity of Monte Carlo path simulations is minimised. In this thesis, we investigate this method in pricing path-dependent options and the computation of option price sensitivities also known as Greeks.

November 2014.



Declaration

I declare that **Multilevel Monte Carlo Simulation in Option Pricing** is my own work, that it has not been submitted before for any degree or examination in any other university, and that all the sources I have used or quoted have been indicated and acknowledged as complete references.



Funmilayo Eniola Kazeem

November 2014

Signed:.....

Acknowledgement

Firstly, I thank God Almighty for this Master's opportunity and for strength and wisdom to write this project.

Sincere thanks also go to my supervisor, Prof. Kailash C. Patidar and my co-supervisor, Dr. Raouf Ghomrasni, who were very kind toward me and meticulous in their supervision of this project.

I also very much appreciate the contribution of Professor Michael B. Giles of the University of Oxford towards the coding part of this thesis.

A very big thank you to my family for their support, prayers and encouragement.

At this point, I will not fail to mention AIMS because that is where this whole journey began. Thank you so much for believing in me.

To all the friends I made in South Africa, thank you for making my stay a pleasant one. I will remember you always.

The only constant thing in life is change and though we be out of sight, we will endeavour not to be out of mind.

DEDICATION

This thesis is dedicated to the Almighty God for His unfailing love and never-ending mercies. All glory be to His name forever.



Contents

Keywords	i
Abstract	iii
Declaration	iv
Acknowledgement	v
Dedication	vi
List of Figures	x
List of Tables	1
1 General Introduction	2
1.1 Motivation for the research	2
1.2 Some preliminaries	3
1.2.1 The underlying process and options	3
1.3 Monte Carlo simulation in general	5
1.3.1 Standard Monte Carlo method	6
1.3.2 Confidence intervals and variance reduction	8
1.4 Improved Monte Carlo methods	12
Least Squares Monte Carlo method	12
Quasi Monte Carlo method	13

1.5	Literature review	14
1.6	Outline of the thesis	17
2	Monte Carlo Simulation for Options Pricing	19
2.1	Modelling the stock price and pricing options	19
2.2	Simulating a geometric Brownian motion	20
2.3	Pricing options using geometric Brownian Motion	22
2.3.1	European options	24
2.3.2	Asian options	25
2.3.3	Lookback options	26
2.3.4	Convergence test	27
2.4	Quantitative analysis and complexity theorem for the standard Monte Carlo method	28
3	Multilevel Monte Carlo Simulation in Option Pricing: Theoretical Framework	33
3.1	Computational complexity	38
3.2	Multilevel path simulation	39
3.2.1	Complexity improvements	39
3.3	Comments on the complexity theorem	41
4	Multilevel Monte Carlo Simulation: Numerical Experiments	42
4.1	Multilevel Monte Carlo algorithm	42
4.1.1	Application to the European option	44
4.1.2	Application to the Asian option	46
4.1.3	Application to the Lookback option	48
4.2	An improvement on the Multilevel approach	50
4.2.1	Application to the European option with the Milstein scheme	51
5	Computing Greeks Using Multilevel Monte Carlo Method	54
5.1	Greek estimates for European call options	56

5.2	Monte Carlo Simulation of Greeks	63
5.2.1	Pathwise sensitivities	63
5.2.2	Likelihood Ratio method	63
5.3	Valuation of Greeks using Multilevel Monte Carlo Simulation	64
6	Concluding remarks and scope for future research	65
A	Multilevel Monte Carlo Algorithm	67
	Conclusion	68



List of Figures

4.1	Geometric Brownian motion with European Option (option value ≈ 0.104476).	44
4.2	Effect of ε for Geometric Brownian motion with European Option . . .	45
4.3	Geometric Brownian motion with Asian option (option value ≈ 0.057641).	47
4.4	Effect of ε for Geometric Brownian motion with Asian Option	47
4.5	Geometric Brownian motion with lookback option (option value ≈ 0.172175).	49
4.6	Effect of ε for Geometric Brownian motion with lookback option. . . .	49
4.7	Geometric Brownian motion with European option (option value ≈ 0.103495) using the Milstein scheme.	52
4.8	Effect of ε for Geometric Brownian motion with European option using the Mistein scheme.	52

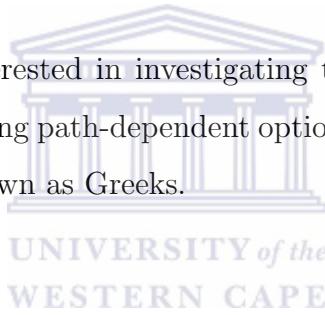
List of Tables

1.1	Option types and their payoffs	5
2.1	Payoffs of European options	25
2.2	Payoffs of an Asian option when $n = 10$	25
2.3	Payoffs of an Asian option when $n = 50$	25
2.4	Payoffs of an Asian option when $n = 200$	25
2.5	Payoffs of a lookback option when $n = 10$	26
2.6	Payoffs of a lookback option when $n = 50$	26
2.7	Payoffs of a lookback option when $n = 200$	26
2.8	Error Vs M	28
4.1	$\varepsilon^2 C / (\log \varepsilon)^2$ for the European option.	45
4.2	MC costs, Multilevel MC (mlmc) costs and their respective savings.	46
4.3	$\varepsilon^2 C / (\log \varepsilon)^2$ for the Asian option.	48
4.4	MC costs, Multilevel MC (MLMC) costs and their respective savings.	48
4.5	$\varepsilon^2 C / (\log \varepsilon)^2$ for the lookback option.	50
4.6	MC costs, Multilevel MC (MLMC) costs and their respective savings.	50
4.7	$\varepsilon^2 C / (\log \varepsilon)^2$ for the European option using the Milstein scheme.	53
4.8	MC costs, Multilevel MC (MLMC) costs and their respective savings using the Milstein scheme.	53

Chapter 1

General Introduction

In this thesis, we are interested in investigating the multilevel Monte Carlo path simulation method in pricing path-dependent options and the computation of option price sensitivities also known as Greeks.



1.1 Motivation for the research

Many situations arise, where modelling and computer simulation are vital tools and where mathematical models employed there have been shown to portray reality accurately. However, the parameters appearing in the models often have to be estimated from measurements and therefore, are subject to uncertainty. This uncertainty spreads through the simulations and weighing its impact on the results obtained is often-times of great import. In applications, what is of interest usually is the expected value of a quantity which is a functional of the solution of a stochastic differential equation (SDE). This could for example, be the price of an option, like in the instance of computational finance.

There are two basic methods of approximating functions of solutions of SDEs: the *finite difference method* and the *Monte Carlo algorithm*. Although finite differences

produce very accurate results in simple cases, they are very computationally expensive for higher-dimensional problems. On the other hand, although less accurate, Monte Carlo algorithms are relatively simple to implement and extend to more complex cases. The well-known slow rate of convergence of the standard Monte Carlo method implies that we require many such realisations to obtain accurate results, and the standard Monte Carlo approach quickly becomes infeasible.

The computational cost of solving SDEs is a major challenge in computational finance. In this thesis, we address the problem of the large cost of solving SDEs. The strategy used is based on a revolutionary variance reduction technique for the standard Monte Carlo method, called the multilevel Monte Carlo (MLMC) method. The basic idea was introduced by Heinrich [41] to speed up Monte Carlo computations of high-dimensional, parameter dependent integrals and to solve integral equations. Ideas similar to this were used by Brandt et al. in statistical mechanics [1, 2]. The method was extended by Giles [30, 31] to infinite-dimensional integration related to stochastic differential equations in finance. Since then, it has found application in many areas of mathematics related to differential equations, in particular stochastic differential equations [30, 40] and several types of partial differential equations (PDEs) with random forcing [32] or random coefficients [7].

1.2 Some preliminaries

Here, we briefly give some background on some of the terms that will be useful in the reading of this work.

1.2.1 The underlying process and options

Financial instruments can be classified into cash instruments and derivative instruments:

CHAPTER 1. GENERAL INTRODUCTION

- *Cash Instrument* - e.g., stocks, shares, bonds and commodities;
- *Derivative Instruments* - e.g., derivatives, which promise some payment or delivery in the future, they are called derivatives because their value is derived from the underlying stock(s).

Stock prices are often modelled as stochastic processes (S_t) where t is a (discrete or continuous) index indicating time. The current stock price is denoted S_0 .

Options are a specific type of derivative, that give the holder the right (but not the obligation) to buy or sell a certain amount of the underlying stock on (or sometimes before) a specified date (the *expiration date* or *maturity*) for a specified price (often called the *strike price*). We will denote the *maturity* by T and the *strike price* by K . Options that give the holder the right to buy are called *call options*, while options that give the holder the right to sell are called *put options*. When an option is used to buy or sell stock, we say that it is *exercised*. Options come in many flavours, of which we will mention some of the well-known ones.

- (i) *European options*. These options give the holder the right to buy (call option) or sell (put option) a number of units of the underlying stock on time T for strike price K . The payoff of a European call option is $(S_T - K)^+$ and that of a put option is $(K - S_T)^+$, where $(x)^+ = \max(x, 0)$.
- (ii) *American options*. These are like European options, but they can be exercised at or before time T .
- (iii) *Bermudan options*. Options that can be exercised at any fixed period of time.
- (iv) *Asian options*. Options whose payoff depends on the average underlying asset over a certain period of time.
- (v) *Barrier options*. Options either come into existence after a barrier is breached (up-and-in or down-and-in) or drop out of existence as a result of breaching the barrier (up-and-out or down-and-out).

- (vi) *Lookback options.* Options that depend on the minimum (for call) or maximum (for put) value of the stock price over a certain period of time.
- (vii) *Digital options.* Options whose payoff is fixed after the underlying asset exceeds the exercise price.

See below a table summarising different option types and their corresponding payoffs.

Table 1.1: Option types and their payoffs

Option	Payoff
European call	$(S - K)^+$
European put	$(K - S)^+$
Asian call	$(\text{mean}(S) - K)^+$
Asian put	$(K - \text{mean}(S))^+$
Lookback call	$\left(K - \min_{0 \leq t \leq T} S(t) \right)^+$
Lookback put	$\left(\max_{0 \leq t \leq T} S(t) - K \right)^+$

In the next section, we present the general framework of the Monte Carlo method which will be used for option pricing problems considered in this thesis.

1.3 Monte Carlo simulation in general

Mathematical modelling traditionally focused on realistic yet tractable models. It was often the case that a simple model was favoured over a complex but more realistic one, since the latter, more complex model was harder to analyse mathematically. Nowadays, fast computers are available and more complex models can be analysed numerically. One of the approaches in this area is simulation.

In this section, we provide the basic techniques behind a simple simulation tool known as Monte Carlo simulation and look at the applications in options pricing.

Monte Carlo simulation relies on repeated random sampling to compute their results. It is widely used in different areas in mathematics and physics such as fluids, cellular structures, queueing theory and risk theory. Another application of Monte Carlo in financial mathematics, that we shall not explore, deals with the analysis of credit risk. We describe the Monte Carlo method and explain its limitations, hence the need of another variance reduction technique.

1.3.1 Standard Monte Carlo method

If you throw a fair coin many times, you expect that the fraction of heads will be about 50%. In other words, if we see the tossing of coins as Bernoulli experiments and we define random variables X_i to be equal to 1 if heads occur, and 0 if tails occur, we expect the sample mean

$$\overline{X}_n := \frac{(X_1 + X_2 + \cdots + X_n)}{n},$$

to be close to the theoretical mean $\mathbb{E}[X_i] = \frac{1}{2}$.

The law of large numbers states that this holds in a general setting: if X_1, X_2, \dots are independent, identically distributed (i.i.d.) random variables with mean $x := \mathbb{E}[X_1]$ and finite variance, the probability of the *sample mean* being close to x is large.

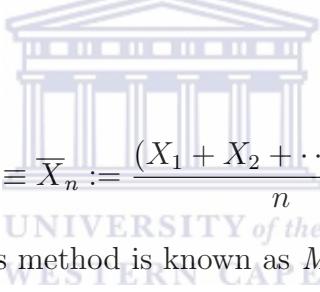
The general idea of the Monte Carlo method is that we have some quantity of interest, that we can not compute explicitly and thus for which we require an approximation. We write our quantity of interest as the expected value of some random variable. To approximate this quantity numerically, we take a number of samples of this random variable and compute the mean. The method builds upon the law of large numbers and thereby large samples of random numbers. The law of large numbers assures that a sample of independent, identically distributed (i.i.d.) random variables, converges to the sample mean as the sample size, n increases. More

precisely, if X is a random variable with its expected value and variance given by μ and σ^2 respectively and X_1, X_2, \dots is a sequence of independent and identically distributed random variables with expected value μ and variance σ^2 ; then by the law of large numbers,

$$\bar{X}_n = \frac{\sum_{i=1}^n X_i}{n} \longrightarrow \mu = \mathbb{E}[X] \text{ almost surely as } n \rightarrow \infty.$$

The proof and conditions can be found in literature.

Now suppose that we have a method to obtain i.i.d outcomes X_1, X_2, \dots, X_n while we do not know the value of x . Think for example of a coin with an unknown probability x of throwing heads. The discussion above suggests using the sample mean



$$\bar{X} \equiv \bar{X}_n := \frac{(X_1 + X_2 + \dots + X_n)}{n},$$

as an estimator for x . This method is known as *Monte Carlo simulation* (after the famous city with many casinos). As we will see, many quantities of interest can be expressed as an expectation and can therefore be estimated using Monte Carlo simulation.

In option pricing, the price is expressed by the *discounted* expected value of the option payoff at maturity, under the risk neutral measure. The Monte Carlo method was introduced to option pricing to evaluate this expected value, which is a function of random variables. The procedure now is to evaluate a function of random variables $f(X_i)$ until there is convergence of the sample mean.

As a simple example, let $S(T)$, the asset price at time T , be a random variable with known distribution. Now let $\phi(S(T)) = \max(S(T) - K, 0)$ denote the payoff of an European call option with strike price K . The expectation under the risk-neutral measure \mathcal{Q} denoted $\mathbb{E}^{\mathcal{Q}}[\phi(S(T))]$ can now be determined by simulating independent

trials of $S(T)$ and simultaneously computing $\phi(S(T))$ until the convergence of the sample mean is obtained, that is,

$$\frac{1}{n} \sum_{i=1}^n \phi(S_i(T)) \longrightarrow \mathbb{E}^Q[\max(S(T) - K, 0)] \text{ almost surely as } n \rightarrow \infty.$$

1.3.2 Confidence intervals and variance reduction

We establish the correlation between the Monte Carlo method and variance reduction. In other words, we explain why variance reduction methods are required for a good accuracy of Monte Carlo.

We are interested in estimating a certain quantity x , and we have a random variable X , such that $x = \mathbb{E}[X]$. Given independent realisations X_1, X_2, \dots, X_n of the random variable X , we already know that the sample mean $\bar{X}_n = \frac{(X_1 + X_2 + \dots + X_n)}{n}$, is an unbiased estimator of x . Although this is interesting in itself, we are often also interested in the quality of the estimator: if there is a large spread in the outcomes X_i , it is quite possible that the estimator \bar{X} is far off from the true value x .

We will often use confidence intervals centered around the estimator \bar{X} to give an estimate that expresses in some sense the spread of the estimate. A confidence interval is an interval in which we can assert that x lies with a certain degree of confidence.

Mathematically speaking, a $100(1 - 2\alpha)\%$ confidence interval is a random interval (usually depending on the outcomes X_i), such that the probability that the interval covers the true value of x equals $1 - 2\alpha$. In this thesis, we will focus on approximate confidence intervals, inspired by the central limit theorem. Better confidence intervals can be constructed using Student's t -distribution

Consider a sequence of i.i.d. random variables X_1, X_2, \dots, X_n , with common mean x and variance σ^2 . We have already introduced the sample mean \bar{X} . We will now

define the sample variance S^2 .

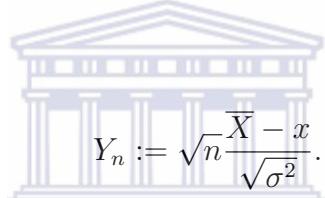
Definition 1.3.1. *The quantity S^2 defined by*

$$S^2 := \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2,$$

is called the sample variance of the random sample X_1, X_2, \dots, X_n and a justification for this name lies in the fact S^2 is an unbiased estimator of σ^2 , i.e. $\mathbb{E}[S^2] = \sigma^2$.

The *Central Limit Theorem*, often abbreviated as CLT, relates the sample mean of a sufficiently large number of random variables to the normal distribution.

Theorem 1.3.1. (*Central Limit Theorem*). *Let the sequence of random variables Y_n be defined by*



$$Y_n := \sqrt{n} \frac{\bar{X} - x}{\sqrt{\sigma^2}}.$$

Then Y_n converges in distribution to a standard normal random variable, or

$$\lim_{n \rightarrow \infty} \mathbb{P}(Y_n \leq y) = \Phi(y),$$

where $\Phi(\cdot)$ is the standard normal cumulative distributive function, given by

$$\Phi(y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^y e^{-\frac{1}{2}t^2} dt.$$

This theorem is often interpreted as $\mathbb{P}(Y_n \leq y) \approx \Phi(y)$ and we will use it to construct approximate confidence intervals.

Note that Y_n depends on an unknown parameter, σ^2 . Fortunately, by a theorem known as Slutsky's theorem, the theorem(CLT) still holds if we replace σ^2 by the sample variance, S^2 , so that the quantity

$$Y_n := \sqrt{n} \frac{\bar{X} - x}{\sqrt{S^2}},$$

still converges in distribution to a standard normal random variable.

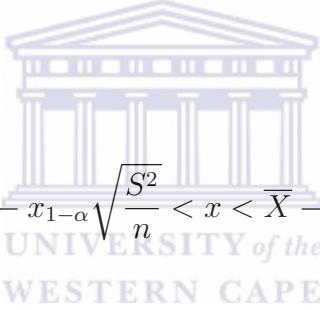
If x_α is the α quantile of the standard normal distribution, that is $\Phi(x_\alpha) = \alpha$, it follows that

$$\mathbb{P}(x_\alpha < Y_n < x_{1-\alpha}) \approx 1 - 2\alpha.$$

By rewriting the event $\{x_\alpha < Y_n < x_{1-\alpha}\}$, we find that

$$x_\alpha < \sqrt{n} \frac{\bar{X} - x}{\sqrt{S^2}} < x_{1-\alpha},$$

is equivalent to



$$\bar{X} - x_{1-\alpha} \sqrt{\frac{S^2}{n}} < x < \bar{X} - x_\alpha \sqrt{\frac{S^2}{n}},$$

UNIVERSITY of the
WESTERN CAPE

so that

$$\mathbb{P}\left(\bar{X} - x_{1-\alpha} \sqrt{\frac{S^2}{n}} < x < \bar{X} - x_\alpha \sqrt{\frac{S^2}{n}}\right) \approx 1 - 2\alpha.$$

From this, it follows that $\left(\bar{X} - x_{1-\alpha} \sqrt{\frac{S^2}{n}} < x < \bar{X} - x_\alpha \sqrt{\frac{S^2}{n}}\right)$ is an approximate $100(1 - 2\alpha)\%$ confidence for x .

There are many choices for the confidence level but here, a 95% confidence level is used because in 95% of cases, the true value lies in the confidence interval. This corresponds to $\alpha = 0.025$ and $x_{1-\alpha} = -x_\alpha \approx 1.96$. Our confidence interval is then

$$I = \left(\bar{X} - 1.96 \sqrt{\frac{S^2}{n}}, \bar{X} + 1.96 \sqrt{\frac{S^2}{n}}\right).$$

From the factor \sqrt{n} in the denominator, it can be seen that in order to obtain one

extra digit of the value x , the number of runs must be increased by a factor of 100.

Hence, worthy of note are the following:

1. One of the main disadvantages of the Monte Carlo method is that the size of the confidence interval grows smaller like the inverse of the square root of the number of samples n and if we are not careful, the actual parameter will not be included in the interval.
2. The size of I is directly proportional to the standard deviation of X which implies that the smaller the variance of X is, the better the Monte Carlo method works.

These two points lead to the need for what is called variance reduction. Since the size of the confidence interval is influenced by the value of $\sqrt{\text{Var}(X)}/\sqrt{n}$, we need to find another method to decrease the width of the interval other than increasing n . To do this, X could be replaced with a random variable Y such that $\mathbb{E}(Y) = \mathbb{E}(X)$, but $\text{Var}(Y) < \text{Var}(X)$. In this case, via the Monte Carlo method, we compute the expectation of X by using Y because since the variance of Y is lower, the results will be better. We need to find this Y however. There are two main methods to find Y :

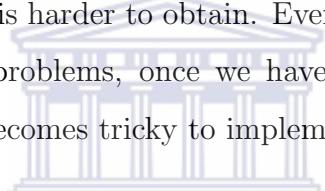
- a) The method of antithetic variates.
- b) The method of control variates.

We shall not describe these methods here but the interested reader is referred to [17]. We shall see later that the multilevel Monte Carlo introduces a control variate.

From [37] we see that a difficulty however occurs for Monte Carlo valuation of American options. Options that depend on multiple underlying securities or that involve path dependent features require that the Monte Carlo method is used. Since the determination of the optimal exercise time depends on an average over futuristic events, Monte Carlo simulation for an American option has a “Monte Carlo on Monte

Carlo” feature that increase the complexity of the computation.

The above analysis suggests an improvement be done on the classical Monte Carlo. The Monte Carlo method is more or less efficient for an European option. In fact, we have an analytical solution, for example, via the Black-Scholes formula. Of more import, the value is determined only by the terminal stock price if one assumes a given start point, time, constant interest rate and volatility. It is easily observed that the Monte Carlo simulation must work in a forward manner. However, for an American option, because of early exercise, in contrast to a partial differential equation, it becomes imperative to know the option value at the intermediate times between the simulation start time and the option expiry time. In Monte Carlo however, this information is harder to obtain. Even though it is simple and capable of handling multi-factor problems, once we have to solve a problem backwards, Monte Carlo simulation becomes tricky to implement (see [37]).



1.4 Improved Monte Carlo methods

This section briefly describes some improvement on the Monte Carlo method used to value options.

Least Squares Monte Carlo method

Here we describe briefly, the least squares Monte Carlo method as seen in [37]. There is basically one way to value American-style options, instead of determining the exercise boundary before simulation, this approach relies on the conditional expectation function; see for examples [8], Tsitsiklis and Roy (1999). Longstaff and Schwartz [14] suggested the least squares Monte Carlo (LSM) method, a simple way to implement this approach. Clemént et. al in [12], investigated related convergence issues. Tian and Burrage (2002) debated the accuracy of the LSM method and

Moreno and Navas (2003) further debated the robustness of the LSM with regard to the choice of the basis functions.

As seen in [14], Longstaff and Schwartz introduced the use of Monte Carlo simulation and least squares algorithm of [8] to compute the value of American options since we do not require anything more than simple least square. At each exercise time point, option holders compare the payoff for immediate exercise with the expected payoff for continuation. The options are exercised if the payoff for immediate exercise is higher. Otherwise, the options will be left alive. The expected payoff for continuation is conditional on the information available at that time point. The key insight underlying this approach is that this conditional expectation can be estimated from the cross-sectional information in the simulation by using least squares. This allows us to apply this method readily in path-dependent and multi-factor situations where traditional finite difference techniques cannot be used.

To derive the conditional expectation function, we regress the realized payoffs from continuation on a set of basis functions in the underlying asset prices. The fitted values are chosen as the expected continuation values. We now simply compare these continuation values with the immediate exercise values and make the optimal exercise decisions, then we obtain a complete specification of the optimal exercise strategy along each path. We use this algorithm recursively and discount the optimal payoffs to time zero. That gives the option price.

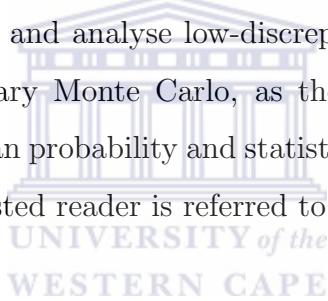
Quasi Monte Carlo method

This section discusses alternatives to Monte Carlo simulation known as quasi-Monte Carlo or low-discrepancy methods as seen in [35]. These methods differ from ordinary Monte Carlo in that they make no attempt to mimic randomness. Indeed, they seek to increase accuracy specifically by generating points that are too evenly distributed to be random. Applying these methods to the pricing of derivative se-

curities requires formulating a pricing problem as the calculation of an integral and thus suppressing its stochastic interpretation as an expected value. This contrasts with the variance reduction techniques which take advantage of the stochastic formulation to improve precision.

Low-discrepancy methods have the potential to accelerate convergence from the $\mathcal{O}(1/\sqrt{n})$ rate associated with Monte Carlo to nearly $\mathcal{O}(1/n)$ convergence: under appropriate conditions, the error in a quasi-Monte Carlo approximation is $\mathcal{O}(1/n^{1-\varepsilon})$ for all $\varepsilon > 0$. Variance reduction techniques, affecting only the implicit constant in $\mathcal{O}(1/\sqrt{n})$ are not nearly so ambitious. However, the ε in $\mathcal{O}(1/n^{1-\varepsilon})$ hides a dependence on problem *dimension*.

The tools used to develop and analyse low-discrepancy methods are very different from those used in ordinary Monte Carlo, as they draw on number theory and abstract algebra rather than probability and statistics. For a detailed understanding of this method, the interested reader is referred to [35].



1.5 Literature review

As seen in [37], option valuation is an important sub-area of research in the mathematical finance community. Earlier, due to the many problems that arise with simulation, the primary methods for pricing American options are binomial trees and other lattice methods such as trinomial trees, and finite difference methods to solve the associated boundary value partial differential equations (PDEs). Due to the complexity of the underlying dynamics, analytical models for option pricing suggest many restrictive hypothesis, so to fit in with reality, approximate numerical methods are used, especially for American options, these include the valuation of options, the estimation of their sensitivities, risk analysis. But in recent years there has been a huge increase in the complexity of numerical computation in financial theory and practice, laying more demands on computational speed and efficiency.

Many problems in mathematical finance involve the computation of a particular integral. These integrals can be valued analytically in many cases, and still in more cases, they can be valued using numerical integration, or computed using a partial differential equation (PDE). The Black-Scholes model, for instance, provides explicit closed form solutions for the values of certain call and put options (European). However when the dimension of the problem becomes large, PDEs and numerical integrals get out of control, the formulas exhibiting them are complicated and difficult to evaluate accurately by conventional methods. In these cases, Monte Carlo methods naturally give the answer, because they have proved to be valuable and flexible computational tools to calculate the value of options with multiple sources of uncertainty or with complicated features.

The multilevel Monte Carlo method has been employed in different fields to improve the complexity of computations in so many fields, some of which will be described briefly in this section.

It is shown in [9], how to extend the recently proposed multilevel Monte Carlo approach to the continuous time Markov chain setting, thereby greatly reducing the computational complexity needed to compute expected values of functions of the state of the system to an accuracy which has been specified. The extension is not trivial as it makes use of a coupling of the essential processes that is easy to simulate while providing a small variance for the estimator. Further, it is shown how to produce an unbiased estimator that is significantly less expensive computationally (this has nothing to do with other implementations of multilevel Monte Carlo) than the usual unbiased estimator arising from exact algorithms in conjunction with the crude Monte Carlo.

In a manner quantifiable, the basic computational complexity of prevalent approaches that have many names and forms all over scientific literature, including the Bortz-Kalos-Lebowitz algorithm, discrete event simulation, dynamic Monte Carlo, kinetic Monte Carlo, the n-fold way, the next reaction method, the residence-time al-

gorithm, the stochastic simulation algorithm, Gillespies algorithm, and tau-leaping, are dramatically improved. The new algorithm generically applies, but an example is also given, where even without a multilevel discretisation the coupling idea alone, can be used to improve efficiency by exploiting system structure. Stochastically modelled chemical reaction networks give an important application for this work.

The method also has been applied to elliptic PDEs with random coefficients as seen in [20] where the numerical solution of elliptic partial differential equations with random coefficients are considered. Such problems are seen, for example, in quantifying uncertainty for groundwater flow. The multilevel Monte Carlo method, a novel (variance reduction) technique for the standard Monte Carlo method is described and its superiority demonstrated numerically. The asymptotic cost of solving the stochastic problem with the multilevel method is always significantly lower than that of the standard method and grows only proportionally to the cost of solving the deterministic problem in certain situations. Numerical calculations showing the effectiveness of the method for one- and two-dimensional model problems evolving in groundwater flow are also presented therein.

In Kuznetsov et al. (2011) (see [6]), a new Monte Carlo simulation technique was presented for a large family of Lévy processes that is based on the Wiener-Hopf decomposition. Their technique was combined with this recently introduced multi-level Monte Carlo method. Moreover, for the first time a theoretical analysis of the new Monte Carlo simulation technique was provided and of its multilevel variant for finding expectations of functions depending on the historical trajectory of a Lévy process. Convergence rates for both methods are derived and it is shown that with respect to the “jump activity” (e.g. characterised by the Blumenthal-Getoor index), they are uniform. A modified version of the algorithm is also presented, which, when combined with the multilevel method gives the optimal convergence rate for general Lévy processes and Lipschitz functionals.

The work of Hakon et al. (see [15]) generalizes a multilevel forward Euler Monte

Carlo method presented in [31] for the approximation of expected values depending on the solution to an Itô stochastic differential equation. In 2008, Giles [31] proposed and analysed a forward Euler multilevel Monte Carlo method based on a ranking of uniform time discretisations and control variates to reduce the computational effort required by a standard, single level, forward Euler Monte Carlo method. This work as seen in [15], introduces an adaptive hierarchy of non-uniform time discretisations, generated by an adaptive algorithm introduced in [[4], [21], [22]]. This form of the adaptive algorithm generates stochastic, path dependent, timesteps and is based on a posteriori error expansion that was developed first in [3]. The numerical results obtained for a stopped diffusion problem, show savings in the computational cost to achieve a fine accuracy.

The paper [28] reviews the multilevel Monte Carlo path simulation method for estimating option prices in computational finance as seen in Giles [31], and further extends the method by combining it with quasi-Monte Carlo integration using a randomised rank-1 lattice rule. It is demonstrated using the Milstein discretisation of the stochastic differential equation, that the combination has considerably reduced computational cost than either one on its own for evaluating lookback, Asian, barrier, European and digital style of options.

Note that all work that has been done using this multilevel approach have successfully reduced computational cost as this is the main goal of the method. In this project however, we consider applications of this method to finance.

1.6 Outline of the thesis

This thesis is divided into six chapters. The first chapter presents background material on this research which includes some basic tools and notation used throughout this thesis.

CHAPTER 1. GENERAL INTRODUCTION

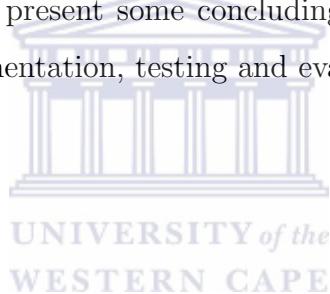
In Chapter 2, we review some applications of Monte Carlo method used for option pricing problems. We place emphasis on European, Asian and lookback options and give a quantitative analysis of the method used. This includes implementing the models and evaluating them to show the convergence of the Monte Carlo simulations.

Chapter 3 introduces the multilevel Monte Carlo approach which is an improvement of the standard Monte Carlo in terms of efficiency.

In Chapter 4, we present application of multilevel Monte Carlo method.

Chapter 5 gives a brief overview on how the multilevel Monte Carlo approach can be used in the computation of Greeks.

Finally, in Chapter 6, we present some concluding remarks and discuss the scope for future research implementation, testing and evaluation are carried out.



Chapter 2

Monte Carlo Simulation for Options Pricing

This chapter is devoted to option pricing using Monte Carlo simulation. Here, we look at the simulation of a geometric Brownian motion and then we look at how to price different option types with Monte Carlo.

2.1 Modelling the stock price and pricing options

A model that is often used for financial markets is the Black-Scholes model that is named after Fisher Black and Myron Scholes, described in their 1973 paper, *The pricing of options and corporate liabilities* (see [13]). Black and Scholes model the stock price as a geometric Brownian motion. That is, given the current stock price S_0 , the stock price process $(S_t)_{t \geq 0}$ is such that

$$\log \left(\frac{S_t}{S_0} \right)$$

is a Brownian motion.

An important problem in mathematical finance is the valuation or pricing of derivatives. At the moment that a derivative is taken out, it is not known what its value will be in the future, as it depends on the future value of the underlying stock.

2.2 Simulating a geometric Brownian motion

For the convenience of the reader, we will start off with the definition of the standard Brownian motion, which forms the basis of the Black-Scholes model. It is not necessary to recall the history of the Brownian motion, it is important to mention that Brownian motion is one of the most well-known and fundamental stochastic processes. It appeared heuristically around 1827 by observation of the botanist Robert Brown but was given a rigorous mathematical treatment by Norbert Wiener, which is why it is often called the Wiener process. Einstein's explanation of the Brownian motion explains why the Brownian motion is a popular model for stock prices. In this section, we will look at pricing options within this model. For much more information on Brownian motion, see [39].

Definition 2.2.1. *A standard Brownian motion (SBM) (or a standard Wiener process) is a stochastic process $(B_t)_{t \geq 0}$ (that is, a family of random variables B_t , indexed by nonnegative real numbers t , defined on a common probability space (Ω, \mathcal{F}, P)) with the following properties*

- (1) $B_0 = 0$.
- (2) *With probability 1, the function $t \rightarrow B_t$ is continuous in t .*
- (3) *The process $(B_t)_{t \geq 0}$ has stationary (for each $t > s > 0$, the distribution of $B_t - B_s$ depends only on $t - s$), independent increments (or each $t > s > 0$, $B_t - B_s$ is independent of the values $(B_u)_{0 \leq u \leq s}$).*
- (4) *The increments are stationary and normally distributed (for each $t \geq s \geq 0$, $B_t - B_s \sim \mathcal{N}(0, t - s)$).*

The property (3) shows that if $s < t$, we have $B_t = X_1 + X_2$, where $X_1 \sim \mathcal{N}(0, s)$ and $X_2 \sim \mathcal{N}(0, t - s)$. This inspires the following discrete approximation of the Brownian motion. Let T be our time horizon; we will approximate sample paths from the Brownian motion on the time interval $[0, T]$. Let M be large and let $\tau = T/M$. By definition, for each $i = 1, 2, \dots, M$, we have

$$W_i := B_{i\tau} - B_{(i-1)\tau} \sim \mathcal{N}(0, \tau),$$

and all these W_i are mutually independent. Also observe that

$$W_1 + W_2 + \dots + W_k = B_{k\tau} - B_0 = B_{k\tau}.$$

Thus, the above summation can be used as a discretized approximation to the process up to time $k\tau$. Increasing M makes the grid finer. The above discussion is formalised in the following theorem:

Theorem 2.2.1. *Let $0 = t_0 < t_1 < t_2 < \dots < t_M = T$ be a subdivision of the interval $[0, T]$, and $X_1, X_2, \dots, X_n \sim \mathcal{N}(0, \tau)$ be independent. Define*

$$Z_0 = 0, \quad Z_i = Z_{i-1} + \sqrt{t_i - t_{i-1}} X_i \quad (i = 1, 2, \dots, M)$$

Then (Z_0, Z_1, \dots, Z_M) is equal in distribution to $(B_{t_0}, B_{t_1}, \dots, B_{t_M})$ which is a standard Brownian motion.

More generally, let $(B_t)_{t \geq 0}$ be a standard Brownian motion. A stochastic process $(S_t)_{t \geq 0}$ is called a Brownian motion (also μ, σ^2 -Brownian motion) if it is equal in distribution to

$$S_t = \mu t + \sigma B_t. \tag{2.2.1}$$

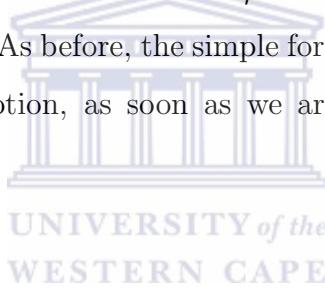
The parameters μ and $\sigma > 0$ are respectively called the drift and the volatility of the Brownian motion. For example, if $\mu > 0$, the process has the tendency to

increase over time, while a large value of σ increases the variability (volatility) of the process. The simple formula (2.2.1) shows that if we are able to simulate a standard Brownian motion, then we are also able to simulate a general Brownian motion.

Moreover, since Brownian motion would allow for negative stock prices, it is not really adequate to model stock markets. The Black-Scholes model uses a *geometric Brownian motion (GBM)* instead. If we denote, as before, by $(B_t)_{t \geq 0}$ a standard Brownian motion, then the process

$$S_t = S_0 \exp(\mu t + \sigma B_t). \quad (2.2.2)$$

is a geometric Brownian motion with drift μ and volatility σ . $(S_t)_{t \geq 0}$ will be the model of our stock prices. As before, the simple formula (2.2.2) allows us to simulate a geometric Brownian motion, as soon as we are able to simulate the standard Brownian motion.



2.3 Pricing options using geometric Brownian Motion

Pricing options in the GBM can be done using the risk-neutral measure. We recall quickly that *risk-neutral measure* is a probability measure under which the current value of all financial assets at time t is equal to the expected future payoff of the asset discounted at the risk-free rate, given the information structure available at time t . Under this measure, the stock price follows a geometric Brownian motion with drift $r - \sigma^2/2$ and volatility σ . Here, r is the continuously compounded interest rate. A general option has a payoff function that depends on the sample path of the underlying Brownian motion, say

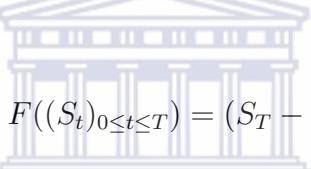
$$\text{payoff} = F((S_t)_{0 \leq t \leq T}) \quad (2.3.3)$$

for some function F . The price of this option then will be

$$S_t = \exp(-rT) \mathbb{E}[F(S_0 \exp(r - \sigma^2/2)t + \sigma B_t)_{0 \leq t \leq T}),$$

with $(B_t)_{t \geq 0}$ a standard Brownian motion. For details, see e.g. [5].

For simplicity, we shall only be interested in pricing European options. Pricing European options using Monte Carlo simulation is relatively easy, as the price of the option only relies on the stock price path $(S_t)_{t \geq 0}$ through the value S_T : the stock price at maturity. For now, we will focus on the European *call options*, but simulating *put options* works just the same. In the case of European *call options*, the function F from the previous paragraph will be



$$F((S_t)_{0 \leq t \leq T}) = (S_T - K)^+,$$

with K the strike price of the option. In other words: the exact dynamics of the underlying process are not important, we only care about the value

$$S_T = S_0 \exp((r - \sigma^2/2)T + \sigma B_T),$$

which is easily generated, once we are able to generate the value of the standard Brownian motion at time T , which is just an $\mathcal{N}(0, T)$ -distributed random variable!

The general version of Equation (2.3.3) shows that the payoff of an option may depend on the whole sample path, rather than only on the value of the stock at maturity, as is the case with European options. Pricing a general option therefore needs simulation of the whole sample path of a geometric Brownian motion rather than only its value at maturity.

One can adapt the “discretisation” strategy for simulating sample paths of the standard Brownian motion to simulate sample paths of the geometric Brownian motion. As an example, we will consider Asian call options. The value of an Asian call

option is based on the average price of the underlying over the time until maturity. Its payoff may be given as the maximum of the average value over the time until maturity minus the strike, and zero.

When we price options using Monte Carlo simulation, the approach can be summarised into three steps;

1. Simulate n sample paths of the underlying asset price over the time interval (there is really no need for this when dealing with vanilla options because only the price of the underlying asset at maturity is of concern).
2. Calculate the payoff of the option for each path.
3. Average the discounted payoffs over sample paths.

We give below an application of the above described method to valuation of some select options. The tables given below also serve in the analysis of the convergence test. The overall observation is that there is a significant decrease in the standard deviation of the Monte Carlo values as the value of M increases.

2.3.1 European options

For numerical experiments with the European call, we consider the parameters $r = 0.05, \sigma = 0.2, T = 1, S = 110, K = 100$. Note that for the European call option, we have $f(S) = \exp(-rT)(S - K)^+$, whereas for the European put option $f(S) = \exp(-rT)(K - S)^+$, where K is the strike price.

See Table 2.1 below summarising the number of paths, M simulated and the corresponding mean Monte Carlo (payoff) values for a number of timesteps.

Table 2.1: Payoffs of European options

M	10	100	1000	10000	100000
Value (payoff)	13.0839	19.8291	16.7551	17.6857	17.6988
Time elapsed	0.013868	0.016453	0.013841	0.018232	0.034084
$\frac{1.96\hat{b}}{\sqrt{M}}$	8.4881	3.7033	1.1502	0.3698	0.1170

2.3.2 Asian options

For numerical experiments with an Asian call option, we consider the parameters $r = 0.05, \sigma = 0.2, T = 1, S = 110, K = 100$. Note that for the Asian call option, we have $f(S) = \exp(-rT)(\text{mean}(S) - K)^+$ whereas for the Asian put option $f(S) = \exp(-rT)(K - \text{mean}(S))^+$ where K is the strike price.

See below tables summarising the number of paths simulated and the corresponding mean Monte Carlo values.

 Table 2.2: Payoffs of an Asian option when $n = 10$

M	10	100	1000	10000	100000
Value (payoff)	8.2933	12.8821	11.7380	12.0437	12.1829
Time elapsed	0.037312	0.035521	0.039802	0.045071	0.137534
$\frac{1.96\hat{b}}{\sqrt{M}}$	0.8212	0.4349	0.1006	0.0340	0.0110

 Table 2.3: Payoffs of an Asian option when $n = 50$

M	10	100	1000	10000	100000
Value (payoff)	17.8665	9.2448	12.7723	12.3116	12.0896
Time elapsed	0.033780	0.035011	0.047208	0.084150	0.583504
$\frac{1.96\hat{b}}{\sqrt{M}}$	3.7634	0.1493	0.1080	0.0327	0.0096

 Table 2.4: Payoffs of an Asian option when $n = 200$

M	10	100	1000	10000	100000
Value (payoff)	16.6372	10.4254	11.9339	12.2573	12.1187
Time elapsed	0.03780	0.036116	0.055408	0.229478	2.104143
$\frac{1.96\hat{b}}{\sqrt{M}}$	3.4160	0.2354	0.0821	0.0322	0.0095

2.3.3 Lookback options

For numerical experiments with a lookback call option with parameters $r = 0.05$, $\sigma = 0.2$, $T = 1$, $S = 110$, $K = 100$ where the payoff of the European lookback call is given by $c_T = \max(S_T - m_0^T, 0)$, where $m_0^T = \min_{0 \leq u \leq T} S(u)$ whereas that of the European lookback put is given by $p_T = \max(M_0^T - S_T, 0)$, where $M_0^T = \max_{0 \leq u \leq T} S(u)$.

See below tables summarising the number of paths simulated and the corresponding mean Monte Carlo values for a given number of timesteps.

Table 2.5: Payoffs of a lookback option when $n = 10$

M	10	100	1000	10000	100000
Value (payoff)	20.0258	26.8396	25.8916	26.4834	26.6198
Time elapsed	0.032148	0.032137	0.037908	0.092553	0.159515
$\frac{1.96\hat{b}}{\sqrt{M}}$	5.7269	3.3287	1.0546	0.3198	0.1024

Table 2.6: Payoffs of a lookback option when $n = 50$

M	10	100	1000	10000	100000
Value (payoff)	29.0076	31.5362	29.1882	28.8544	28.6651
Time elapsed	0.038412	0.036764	0.042313	0.086680	0.583472
$\frac{1.96\hat{b}}{\sqrt{M}}$	10.1863	3.5128	1.1193	0.3290	0.1037

Table 2.7: Payoffs of a lookback option when $n = 200$

M	10	100	1000	10000	100000
Value (payoff)	35.3644	30.5099	30.1925	29.5986	29.5705
Time elapsed	0.034705	0.063041	0.057638	0.243820	2.246785
$\frac{1.96\hat{b}}{\sqrt{M}}$	14.3213	3.6159	1.0111	0.3315	0.1045

Note that the above formula is valid only when the lookback minimum (or maximum) is sampled continuously. However, the continuous sampling is not infeasible in practice. (Strictly speaking, even the quotations of the stock price are not continuous). It is common to adopt the daily sampling rule in financial markets. It is straightforward to apply the Monte Carlo simulation to pricing discrete-sampling lookback options, whose value is the average present value of the payoff of the lookback option associated with each simulated path. When the sampling frequency increases, the results of the Monte Carlo simulation can approach the theoretical value based on the above analytic solution.

2.3.4 Convergence test

To find the approximation error, we first need to calculate the estimated variance. Let $a = E(X)$ and $b^2 = \text{Var}(X)$ be the expectation and the variance of X respectively. If we obtain M samples X_i for $i = 1, 2, \dots, M$, then the approximation of a is

$$\hat{a} = \frac{1}{M} \sum_{i=1}^M X_i.$$

Therefore, the estimated variance is

$$\hat{b} = \frac{1}{M} \frac{\sum_{i=1}^M (X_i - \hat{a})^2}{M-1}.$$

The central limit theorem implies that the error is approximately normally distributed with mean 0 and variance $\frac{b^2}{M}$, that is, with a standard deviation of $\frac{b}{\sqrt{M}}$. Therefore, the rate of convergence is $\frac{1}{\sqrt{M}}$. Based on the confidence interval built in [17], the expected value a should lie in the 95% interval

$$\left[\hat{a} - \frac{1.96\hat{b}}{\sqrt{M}}, \hat{a} + \frac{1.96\hat{b}}{\sqrt{M}} \right].$$

$\frac{1.96\hat{b}}{\sqrt{M}}$ is the measure of the error of the Monte Carlo simulation. We observe in Table (2.8) that the error bound decreases at a rate of $\frac{1}{\sqrt{M}}$ as M increases, therefore ideally we want an M big enough to reduce the error.

Table 2.8: Error Vs M

M	10	100	1000	10000	100000
$\frac{1.96\hat{b}}{\sqrt{M}}$	10.1863	3.5128	1.1193	0.3290	0.1037

Of course the simulation becomes computationally slower as the number of paths increases.

2.4 Quantitative analysis and complexity theorem for the standard Monte Carlo method

To be more specific, suppose that we are given an SDE with general drift and volatility terms

WESTERN CAPE

$$dS(t) = a(S, t)dt + b(S, t)dB(t), \quad 0 < t < T, \quad (2.4.4)$$

and given initial data S_0 we wish to calculate the expected value of $f(S(T))$, where $f(S)$ is a scalar function which is assumed to be uniformly Lipschitz and $S(T)$ is the final time solution. For simplicity of description, we will suppose that the SDE is scalar - the conclusions stand for general systems. For example, in the case where we wish to value an European call option, the SDE models the dynamics of an asset, under a risk-neutral measure, and has a piecewise linear payoff function $F(x) = \max(x - K, 0)$, where K is the exercise price.

Given a timestep h such that $kh = T$, we apply the Euler-Maruyama method N times to get approximate samples $S_k^i, i = 1, \dots, N$ from the distribution of $S(T)$. Here S_k^i denotes the final time Euler-Maruyama approximation from the i th path.

The computed approximation to $\mathbb{E}[S(T)]$ is then

$$\hat{S} = \frac{1}{N} \sum_{i=1}^N S_k^i. \quad (2.4.5)$$

It is trivial to see that $\mathbb{E}[\hat{S}] = \mathbb{E}[S_k]$. We measure the error of a single run of Monte Carlo with $e = \hat{S} - \mathbb{E}[S(T)]$. There are two error sources in this estimator (2.4.5), one related to the time discretisation and the other due to the fact that we replace the expected value with a finite sample average. More precisely, e splits into

$$\begin{aligned} \mathbb{E}[S(T)] - \hat{S} &= \mathbb{E}[S(T) - S_k + S_k] - \hat{S} \\ &= \mathbb{E}[S(T) - S_k] + \mathbb{E}[S_k] - \hat{S}. \end{aligned}$$

The part played by each of these errors is made clear when we use the *mean square error* $\mathbb{E}[e^2]$. We can express the mean square error as

$$\begin{aligned} \mathbb{E}[e^2] &= \mathbb{E}\left[\left(\hat{S} - \mathbb{E}[S(T)]\right)^2\right] \\ &= \mathbb{E}\left[\left(\hat{S} - \mathbb{E}[\hat{S}] + \mathbb{E}[\hat{S}] - \mathbb{E}[S(T)]\right)^2\right] \\ &= \mathbb{E}\left[\left(\hat{S} - \mathbb{E}[\hat{S}]\right)^2\right] + (\mathbb{E}[S_k] - \mathbb{E}[S(T)])^2. \end{aligned} \quad (2.4.6)$$

The first term in this expression is the variance of the estimator, and by

$$\text{Var}[\hat{S}] = \frac{1}{N^2} \text{Var}\left[\sum_{i=1}^N S_k^i\right] = \frac{1}{N} \text{Var}[S_k],$$

we see that it is proportional to $\frac{1}{N}$. The second term is the square of the *bias* of the approximation.

Ideally, we want the *discretisation error* $S_k - S(T)$ to go to 0 as $h \rightarrow 0$, i.e. the approximation converges. In this context, there are two ways of defining the convergence of the approximated path. The first is the notion of *strong convergence*,

which means that the absolute distance between the paths goes to zero as $h \rightarrow 0$.

We say that the order of strong convergence is m if

$$\mathbb{E} \left[\sup_{0 \leq j \leq K} |S_j - S_{t_j}| \right] = \mathcal{O}(h^m).$$

The other measure of convergence is *weak convergence*, which refers to the convergence of the difference between the expected estimated value and the expected actual value. The order of weak convergence is m if

$$\mathbb{E} [|S_k - S(T)|] = \mathcal{O}(h^m).$$

Returning to the mean square error (2.4.6), we see that the second term, that is, the square of the bias of the estimator, is related to the weak convergence of the discretisation method used. As the variance of the estimator is proportional to $\frac{1}{N}$ and, assuming order 1 weak convergence, the square of the bias is proportional to h^2 , we see that the mean square error is

UNIVERSITY of the
WESTERN CAPE

$$\mathbb{E} [e^2] = \mathcal{O} \left(\frac{1}{N} \right) + \mathcal{O}(h^2).$$

To ensure that the root mean square error is proportional to ε , we must have $\mathbb{E} [e^2] = \mathcal{O}(\varepsilon^2)$ and therefore $\frac{1}{N} = \mathcal{O}(\varepsilon^2)$ and $h^2 = \mathcal{O}(\varepsilon^2)$. This can be achieved by choosing $N = \mathcal{O}(\varepsilon^{-2})$ and $h = \mathcal{O}(\varepsilon)$. On the other hand, the computational cost of the algorithm which is quantified by the number of floating point operations that are needed to compute the estimator, is proportional to the multiple of N , the number of simulated sample paths, and $k = \frac{T}{h}$, the number of timesteps in each sample path. Therefore, the cost is $C_\varepsilon = \mathcal{O}(Nk) = \mathcal{O}(Nh^{-1}) = \mathcal{O}(\varepsilon^{-3})$. This means that the computational complexity, required to achieve a given accuracy ε using the Monte Carlo algorithm, is proportional to ε^{-3} . In other words, we need to take a number of samples N which is large enough, as well as a small enough value for h , so that we have a sufficiently accurate approximation of this expectation. It is concluded that

for Monte Carlo SDE simulations, the cost varies inversely with the third power of the required accuracy.

Another very important parameter in the quantitative analysis of the standard Monte Carlo is the discretisation method used. Discretisation is a method of approximating solutions to SDEs, which can be done in several ways, for example the use of Taylor series expansion to get time-discretised solutions. The approximate solution to a stochastic process X is assumed to follow the SDE

$$dX_t = a(X_t) dt + b(X_t) dB_t, \quad (2.4.7)$$

where a and b are coefficient functions satisfying the conditions for the existence and uniqueness of a solution to the SDE (see [35]). The solution X is approximated over a time interval of $[0, t_m]$ with a time step of $h = t_i/i$ for $i = 0, 1, \dots, m$. Integrating Equation (2.4.7) over $[t, t+h]$ produces

$$X_{t+h} = X_t + \int_t^{t+h} a(X_u) du + \int_t^{t+h} b(X_u) dB_u. \quad (2.4.8)$$

The simplest discretisation method is the *Euler scheme*. In this scheme, the integrals of Equation (2.4.8) are approximated as products of the integrands at time t and the size of the integration domain from t to $t+h$. An alternative method of discretisation that improves the speed of convergence from a strong order of 0.5 for the Euler scheme to 1 is the *Milstein scheme*, which was derived by Milstein (1976). This scheme adds a correction term to the Euler scheme. The idea is to improve the accuracy of the discretisation by considering expansions of order $\mathcal{O}(h)$ to the last term of the Euler scheme. This can be achieved by applying Itô's formula to that term. The Euler scheme has a weak convergence of order 1 and strong convergence of order 0.5. The Milstein scheme, on the other hand, has the same weak convergence but with an improved strong convergence of order 1 (see [18]).

An clear way to improve the complexity would be to use a numerical method with a higher weak order, and this could improve the computational complexity to $\mathcal{O}(\varepsilon^{-2.5})$, see Talay & Tubaro (1990), [11]. However, a revolutionary different approach that gives a complexity of $\mathcal{O}(\varepsilon^{-2}(\log \varepsilon)^2)$ was recently developed by Giles [31], and this technique we will describe in the next chapter.



Chapter 3

Multilevel Monte Carlo Simulation in Option Pricing: Theoretical Framework



In this chapter, we present the multilevel Monte Carlo algorithm and explain the improvements in performance that it offers. We give a precise bound on the cost of the multilevel Monte Carlo method, test the complexity and evaluate the root-mean-square (RMS) error, and confirm its advantage over the classical Monte Carlo method. As mentioned earlier, the basic idea was introduced by Heinrich[41], for two levels, to speed-up Monte Carlo computations of high-dimensional, parameter-dependent integrals and to solve integral equations and later expatiated by Giles [31] in computational finance. The multilevel MC approach which is based on the multi-grid idea for finding iterative solutions of PDEs. The objective is to reduce the computational cost of estimating the payoff value obtained using MC simulations from $\mathcal{O}(\varepsilon^{-3})$ to $\mathcal{O}(\varepsilon^{-2}(\log \varepsilon)^2)$ for the Euler discretisation with an RMS accuracy ε .

It is worth mentioning the following, noted by Higham[10]:

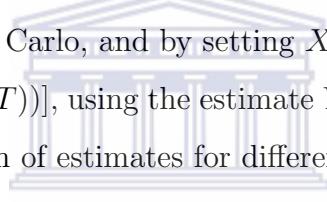
- The multilevel Monte Carlo method does not rely on a special SDE discreti-

sation - the standard Euler-Maruyama method can be used and the analysis just exploits its basic properties of weak and strong convergence .

- Though the aim is to compute an expected value, the method relies on both the weak and the strong convergence behaviour of the numerical method used.

The multilevel Monte Carlo method was developed for computational finance by Giles [31] and includes estimating the same value on different levels l , using a different timestep h_l for each level. In a standard MC simulation, the timestep is fixed at $h = TM^{-1}$, where M is the refinement factor. For the multilevel MC simulation, consider MC simulations with different timesteps $h_l = TM^{-l}$ for different levels of refinements, $l = 0, \dots, L$, with L being the finest refinement.

As in the standard Monte Carlo, and by setting $X = S(T)$ we are again estimating $\mathbb{E}[Y] = \mathbb{E}[f(X)] = \mathbb{E}[f(S(T))]$, using the estimate \hat{Y} , which, in the multilevel Monte Carlo algorithm is the sum of estimates for different levels:



$$\hat{Y} = \sum_{l=0}^L \hat{Y}_l$$

where

$$\hat{Y}_0 = \frac{1}{N_0} \sum_{i=1}^{N_0} \hat{P}_0^{(i)},$$

and

$$\hat{Y}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} \left(\hat{P}_l^{(i)} - \hat{P}_{l-1}^{(i)} \right).$$

Here, \hat{P}_l denotes the approximation $\hat{f}(\hat{X})$ of the value $f(X)$ for a single sample path, simulated using N_l timesteps of size h_l . Now we want to estimate the payoff value in a way that will minimise the variance for a given computational cost as indicated by the superscripts (i) , the approximations \hat{P}_l and \hat{P}_{l-1} are computed from the same sample path, using the same Brownian increments. This will later aid us in reducing the variance of the estimator \hat{Y} . Once we obtain the estimator, the

variance is calculated. This estimator has a variance

$\text{Var}[\hat{Y}_l] = N_l^{-1}V_l$ where $V_l = \text{Var}[\hat{P}_l - \hat{P}_{l-1}]$, which is the variance of a single sample. The combined variance for the combined estimator $\hat{Y} = \sum_{l=0}^L \hat{Y}_l$ is therefore

$$\text{Var}[\hat{Y}] = \sum_{l=0}^L N_l^{-1}V_l.$$

Next, the value of the estimator is tested for convergence. The criterion for convergence is

$$\max \left\{ M^{-1}|\hat{Y}_{L-1}|, |\hat{Y}_L| \right\} < \frac{1}{\sqrt{2}}(M-1)\varepsilon.$$

To minimise the variance, we must choose an optimal number of samples N_l to be proportional to $\sqrt{V_l h_l}$. Hence, the equation to determine optimal N_l [31] is given by

$$N_l = \left\lceil 2\varepsilon^{-2}\sqrt{V_l h_l} \left(\sum_{l=0}^L N_l^{-1}V_l \right) \right\rceil$$

which means that N_l is the smallest integer not less than $2\varepsilon^{-2}\sqrt{V_l h_l} \left(\sum_{l=0}^L N_l^{-1}V_l \right)$.

This optimal N_l takes into account the effect of the computational cost across all levels. Also note that we are simulating N_l paths on each level, and each sample path is simulated using $\frac{T}{h_l}$ timesteps, so the overall computational cost is given by the sum over all levels, $\sum_{l=0}^L N_l h_l^{-1}$.

We can see that $\mathbb{E}[\hat{P}_L]$ splits into

$$\mathbb{E}[\hat{P}_L] = \mathbb{E}[\hat{P}_0] + \sum_{l=1}^L \mathbb{E}[\hat{P}_l - \hat{P}_{l-1}]$$

and that for each $l = 1, \dots, L$, we trivially have

$$\mathbb{E}[\hat{Y}_L] = \frac{1}{N_l} \sum_{i=1}^L \mathbb{E}[\hat{P}_l - \hat{P}_{l-1}] = \mathbb{E}[\hat{P}_L - \hat{P}_{L-1}],$$

and hence,

$$\mathbb{E}[\hat{Y}] = \sum_{l=0}^L \mathbb{E}[\hat{Y}_l] = \mathbb{E}[\hat{P}_0] + \sum_{l=1}^L \mathbb{E}[\hat{P}_l - \hat{P}_{l-1}] = \mathbb{E}[\hat{P}_L].$$

So, though we are using different levels with different discretisation errors to estimate $\mathbb{E}[f(X)]$, the final accuracy depends on the accuracy of L (the finest level). In other words, the expectation of the finest level is equal to the expectation of the coarsest level with an additional sum of the difference between expectations with different levels.

To minimise the computational costs necessary for a given accuracy ε , we minimise the mean square error.

$$\begin{aligned} \mathbb{E}[e^2] &= \mathbb{E}[(\hat{Y} - \mathbb{E}[f(X)])^2] \\ &= \underbrace{\mathbb{E}[(\hat{Y} - \mathbb{E}[\hat{Y}])^2]}_A + \underbrace{(\mathbb{E}[\hat{P}_L] - \mathbb{E}[f(X)])^2}_B, \end{aligned} \quad (3.0.1)$$

where

$$A = \text{Var}[\hat{Y}] = \sum_{l=0}^L \text{Var}[\hat{Y}_l] = \sum_{l=0}^L \frac{1}{N_l} V_l.$$

For a fixed computational cost $C = \sum_{l=0}^L N_l h_l^{-1}$, minimising the variance of \hat{Y} leads us to a constrained optimisation problem for which the Lagrangian is therefore

$$\mathcal{L} = \sum_{l=0}^L \frac{1}{N_l} V_l + \lambda \left(\sum_{l=0}^L N_l h_l^{-1} - C \right).$$

The first order Euler condition is

$$\frac{\delta \mathcal{L}}{\delta N_l} = -\frac{1}{N_l^2} V_l + \lambda h_l^{-1} = 0,$$

which entails that

$$N_l = \lambda^{-\frac{1}{2}} \sqrt{h_l V_l}. \quad (3.0.2)$$

Assuming order 1/2 strong convergence and that the function f is Lipschitz, the variance $\text{Var}[\hat{P}_l - \hat{P}_{l-1}] = O(h_l)$ (see [31]), so the optimal N_l is proportional to h_l . The choice $N_l = O(\varepsilon^{-2} L h_l)$, produces an overall variance proportional to ε^2 . Indeed,

$$\text{Var}[\hat{Y}] = \sum_{l=0}^L \frac{V_l}{N_l} = \sum_{l=0}^L \mathcal{O}(\varepsilon^2 L^{-1} h_l^{-1}) \mathcal{O}(h_l) = \sum_{l=0}^L \frac{\mathcal{O}(\varepsilon^2)}{L} = \mathcal{O}(\varepsilon^2).$$

The term B appearing in (3.0.1) is the square of the bias due to weak error. Assuming $\mathcal{O}(h_l)$ weak convergence for an individual level l , we can see that the bias of the overall method is $\mathcal{O}(h_L) = \mathcal{O}(M^{-L})$. To achieve the desired accuracy, we would want the bias to be proportional to ε , so we set

$$L = \frac{\log \varepsilon^{-1}}{\log M} = \mathcal{O}(\log \varepsilon^{-1}).$$

Summarising the above estimates about the asymptotic behaviour of the multilevel Monte Carlo algorithm, we can see that the overall computational complexity is proportional to

$$\begin{aligned} \sum_{l=0}^L N_l h_l^{-1} &= \sum_{l=0}^L \mathcal{O}(\varepsilon^{-2} L h_l) h_l^{-1} \\ &= \mathcal{O}(\varepsilon^{-2} L^2) = \mathcal{O}(\varepsilon^{-2} (\log \varepsilon)^2). \end{aligned}$$

The algorithm begins by initially setting $L = 0$ and $N = 10^4$. Then we estimate V_l and find the optimal N_l . This optimal N_l is compared to the number of sample paths at that current level and if N_l is larger, then the extra samples are calculated. Then, we test for convergence. If $L < 2$, then go to the next level $L = L + 1$ and repeat the steps. If $L \geq 2$, then we take the convergence test which has the condition. If

the condition for convergence is not met, go to the next level and repeat the steps. Finally, when L converges, we calculate the estimated payoff \hat{P} . This is generally how the multilevel MC method works and it is applicable to different types of option styles.

3.1 Computational complexity

In order to increase efficiency in estimating the expected option value, a certain level of computational savings needs to be achieved. The computational complexity measure is governed by the complexity theorem, as stated by Giles in a general setting so that it can be applied to different option styles including path-dependent styles. The theorem gives bounds to the mean-square-error (MSE) and the computational complexity C of the multilevel estimator \hat{Y} . This theorem states that there exists a positive constant c_1 such that for any $\varepsilon < e^{-1}$, we have the bounds

WESTERN CAPE

$$MSE < \varepsilon^2 \quad \text{and} \quad C \leq c_1 \varepsilon^{-2} (\log \varepsilon)^2.$$

We already know that the computational cost is proportional to $\sum_{l=0}^L N_l h_l^{-1}$ and the cost is calculated as the total number of timesteps across all the levels, with each level performing the coarser and finer timesteps. Thus, we can write the computational cost as

$$C = N_0 + \sum_{l=1}^L N_l (M^l + M^{l-1}). \quad (3.1.3)$$

For the standard MC simulation, the computational cost is defined as

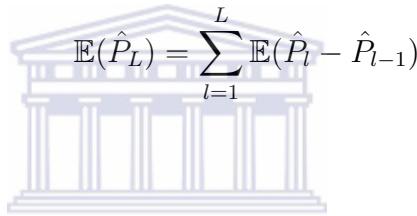
$$C^* = \sum_{l=1}^L N_l^* M^l, \quad (3.1.4)$$

where $N_l^* = 2\varepsilon^{-2}V[P_l]$. This is chosen so that the variance of the estimator is $\frac{1}{2}\varepsilon^2$ similar to that of the multilevel MC simulation.

3.2 Multilevel path simulation

We simulate paths at different levels of fineness as follows:

- At level l , $l = 0, 1, \dots, L$, 2^l timesteps of width $h_l = T/2^l$.
- Let \hat{P}_l be the payoff with level l 's discretisation with



$$\mathbb{E}(\hat{P}_L) = \sum_{l=1}^L \mathbb{E}(\hat{P}_l - \hat{P}_{l-1})$$

- With N_l samples, we estimate
- $$\mathbb{E}(\hat{P}_l - \hat{P}_{l-1}) \simeq \hat{Y}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} (\hat{P}_l^i - \hat{P}_{l-1}^i) \quad (3.2.5)$$

- We independently estimate the different \hat{Y}_l .
- We reuse the leading Brownian motion from \hat{P}_l in \hat{P}_{l-1} .

3.2.1 Complexity improvements

Variance of combined estimator:

$$\text{Var}\left(\sum_{l=1}^L \hat{Y}_l\right) = \sum_{l=1}^L (\text{Var}(\hat{Y}_l)) = \sum_{l=1}^L \left(\frac{1}{N_l} \text{Var}(\hat{P}_l - \hat{P}_{l-1}) \right)$$

Computational cost: $\sum_{l=1}^L (N_l h_l^{-1})$

Targeting an accuracy of ε ,

- Choose L so as to make the discretisation bias small enough.
- Take $N_l \sim \kappa \sqrt{\text{Var}(\hat{P}_l - \hat{P}_{l-1})h_l}$ to minimise the variance at a fixed computational cost.
- Choose κ big enough to have a $\mathcal{O}(\varepsilon^2)$ variance overall.

Here, we give the complexity theorem as stated by Giles [31] which generalises the analysis of the multilevel method.

Theorem 3.2.1 (Complexity theorem). *Let P denote a functional of the solution of the SDE (2.4.4) for a given Brownian path W_t , and let \hat{P}_l denote the corresponding approximation using a numerical discretisation with timestep $h_l = M^{-l}T$. If there exists independent estimators \hat{Y}_l based on N_l Monte Carlo samples, and positive constants $\alpha \geq \frac{1}{2}, \beta, c_1, c_2, c_3$ such that*

- i) $\mathbb{E}[\hat{P}_l - P] \leq c_1 h_l^\alpha,$
- ii) $\mathbb{E}[\hat{Y}_l] = \begin{cases} \mathbb{E}[\hat{P}_0], & l = 0 \\ \mathbb{E}[\hat{P}_l - \hat{P}_{l-1}], & l > 0 \end{cases}$
- iii) $\text{Var}[\hat{Y}_l] \leq c_2 N_l^{-1} h_l^\beta,$
- iv) C_l , the computational complexity of \hat{Y}_l , is bounded by $C_l \leq c_3 N_l h_l^{-1}$,

then there exists a positive constant c_4 such that for any $\varepsilon < e^{-1}$ there are values L and N_l for which the multilevel estimator

$$\hat{Y} = \sum_{l=0}^L \hat{Y}_l,$$

has a mean square error with bound

$$MSE := \mathbb{E}[(\hat{Y} - \mathbb{E}[P])^2] < \varepsilon^2,$$

with a computational complexity C with bound

$$C \leq \begin{cases} c_4 \varepsilon^{-2}, & \beta > 1 \\ c_4 \varepsilon^{-2} (\log \varepsilon)^2, & \beta = 1 \\ c_4 \varepsilon^{-2-(1-\beta)/\alpha}, & 0 < \beta < 1. \end{cases}$$

The theorem tells us that if $\beta > 1$, that is, if the variance of the estimators $\text{Var}[\hat{Y}_l]$ is decreasing faster than $\mathcal{O}(h_l)$ as $h_l \rightarrow 0$, then we can further reduce the computational complexity to $\mathcal{O}(\varepsilon^{-2})$.

3.3 Comments on the complexity theorem

Thanks to literature on weak convergence of discretisation schemes, α is known.

- For the Euler and Milstein discretisation: $\alpha = 1$, even with discontinuous payoffs[43].

The parameter β , related to strong convergence, determines the efficiency of the multilevel approach.

Note that

- For a Lipschitz payoff, Euler: $\beta = 1$, Milstein: $\beta = 2$,
- Not as okay for discontinuous payoffs.
- Generally, β is unknown a priori.

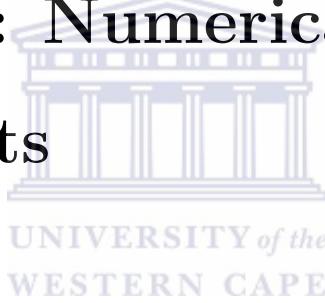
We have to create estimators \hat{Y}_l with β as large as possible.

- Pathwise sensitivities reduce the smoothness by order one.

In the next chapter, we describe the multilevel algorithm and carry out numerical experiments using the method.

Chapter 4

Multilevel Monte Carlo Simulation: Numerical Experiments



Here, we look at applications of the multilevel method to various options and compare results of the standard Monte Carlo method with the multilevel method.

4.1 Multilevel Monte Carlo algorithm

Let us now outline the multilevel Monte Carlo algorithm as executed by a numerical simulation. We begin with $L = 0$ and estimate V_L using an initial $N_L = 100$ samples. We then determine the optimal N_l for $l = 0, 1, \dots, L$ and generate additional samples as needed. Next, we estimate the accuracy of the method and check whether we have already converged. If we have not, we increase L by 1 and repeat the procedure.

To achieve the goal of root mean square error below ε , we must ensure that the estimator variance $\text{Var}[\hat{Y}] < \frac{1}{2}\varepsilon^2$ and that the bias $\mathbb{E}[\hat{P}_L - f(X)] < \frac{\varepsilon}{\sqrt{2}}$. We control the first by using the optimal number of samples on each level, and the second by

increasing the number of levels and thus decreasing the finest timestep.

From Equation (3.0.2), we see that $N_l = \lambda^{-1/2} \sqrt{h_l V_l} = \sqrt{h_l V_l} / \sqrt{\lambda}$ and therefore

$$\text{Var}[\hat{Y}] = \sum_{l=0}^L \frac{V_l}{N_l} = \sum_{l=0}^L \frac{\sqrt{\lambda}}{\sqrt{h_l V_l}} V_l \leq \frac{1}{2} \varepsilon^2.$$

It is easy to see that

$$\lambda \geq 2\varepsilon^{-2} \sum_{l=0}^L \sqrt{V_l/h_l},$$

and thus the optimal number of samples for level l is

$$N_l = \left\lceil 2\varepsilon^{-2} \sqrt{h_l V_l} \left(\sum_{l=0}^L \sqrt{V_l/h_l} \right) \right\rceil. \quad (4.1.1)$$

We can ensure that the bias $\mathbb{E}[\hat{P}_L - f(X)] < \frac{\varepsilon}{\sqrt{2}}$ by using small enough timestep h_L .

If we assume order one or better weak convergence, then $\mathbb{E}[\hat{P}_L - f(X)] = \mathcal{O}(h_L)$ and as $l \rightarrow \infty$, we can approximate the remaining bias with a linear function:

$$\mathbb{E}[f(X) - \hat{P}_l] = \alpha h_l.$$

Hence

$$\begin{aligned} \mathbb{E}[\hat{P}_L - \hat{P}_{L-1}] &= \mathbb{E}[\hat{P}_L - f(X) + f(X) - \hat{P}_{L-1}] \\ &= \alpha(M-1)h_L = (M-1)\mathbb{E}[f(X) - \hat{P}_l], \end{aligned}$$

and we can estimate the remaining bias with $\hat{Y}_L = \frac{1}{N_L} \sum_{i=1}^{N_L} (\hat{P}_L^i - \hat{P}_{L-1}^i)$, which is an estimator for $\mathbb{E}[\hat{P}_L - \hat{P}_{L-1}]$. Therefore, the basic test for convergence is

$$|\hat{Y}_L| \leq \frac{M-1}{\sqrt{2}} \varepsilon.$$

However, if we think of $\mathbb{E}[\hat{P}_L - \hat{P}_{L-1}]$ as a function of h_L that converges to zero

as $h_L \rightarrow 0$, and of \hat{Y}_L as a discrete sample of the value of this function, we see that the function might change signs before the desired accuracy is reached. If we sample the function at point h_L near its root, we will get \hat{Y}_L that will be misleadingly small. To avoid this possibility, Giles [31] suggests estimating the remaining bias with the estimates of the two finest timesteps, therefore using the following criterion for convergence instead:

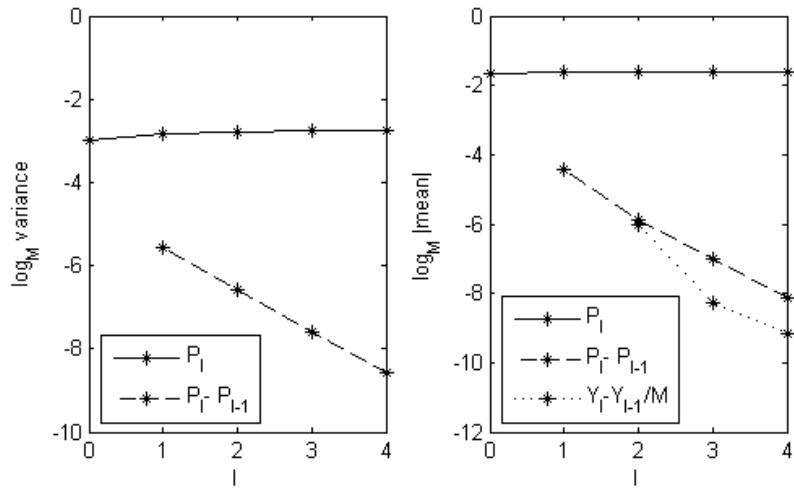
$$\max \left\{ \frac{1}{M} \left| \hat{Y}_{L-1} \right|, \left| \hat{Y}_L \right| \right\} \leq \frac{M-1}{\sqrt{2}} \varepsilon. \quad (4.1.2)$$

We now consider applications of this method to various options.

4.1.1 Application to the European option

Figure 4.1 shows the numerical results for parameters $S(0) = 1, K = 1, T = 1, \sigma = 0.2$ and $r = 0.05$ for a geometric Brownian motion with European option, which has a discounted payoff value of $P = e^{-rT} \max(S - K, 0)$.

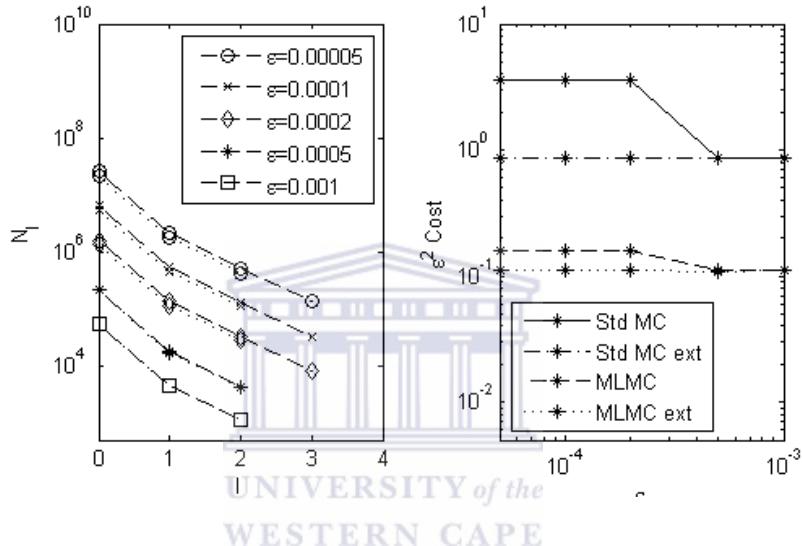
Figure 4.1: Geometric Brownian motion with European Option (option value ≈ 0.104476).



The left plot shows the behaviour of the logarithm base M variance of \hat{P}_l and $\hat{P}_l - \hat{P}_{l-1}$

at different grid levels. $\log |\hat{P}_l - \hat{P}_{l-1}|$ has a gradient of -1 , which indicates that the variance is proportional to M^{-1} , therefore $V_l = O(h_l)$. On the right is a plot of the log of mean to base M at different levels. The mean $\mathbb{E}[\hat{P}_l - \hat{P}_{l-1}]$ has a gradient of -1 , which again indicates a convergence of $O(h_l)$.

Figure 4.2: Effect of ε for Geometric Brownian motion with European Option



In the figure above, the plot at the left shows the influence of ε on the number of paths required to achieve the desired accuracy. The plot at the right is a result of obtaining five sets of multilevel calculations for different user-specified accuracies, ε . By observation, we can see a significant decrease in the computational cost for the multilevel MC method as compared to the standard MC method.

Table 4.1: $\varepsilon^2 C / (\log \varepsilon)^2$ for the European option.

$\varepsilon^2 C$	$(\log \varepsilon)^2$	ratio
0.1112388	47.71708299430558	2.331215e-03
0.10997845	57.773718199472874	1.903607e-03
0.16003196	72.54257985990712	2.206042e-03
0.1593146	84.83036976765435	1.878037e-03
0.159409975	98.07906570323802	1.625321e-03

The computational cost is calculated as $\varepsilon^2 C$ because from the complexity theorem, which states that $\varepsilon^2 C$ should be proportional to $(\ln \varepsilon)^2$. We check by evaluating the

ratio $\varepsilon^2 C / (\log \varepsilon)^2$ as seen in Table (4.1).

Table 4.2: MC costs, Multilevel MC (mlmc) costs and their respective savings.

ε	Standard cost	MC(mlmc) cost	savings
0.00100	8.734409e+005	1.112388e+005	7.851948
0.00050	3.493763e+006	4.399138e+005	7.941928
0.00020	9.070491e+007	4.000799e+006	22.671699
0.00010	3.628196e+008	1.593146e+007	22.773782
0.00005	1.451278e+009	63763990	22.760158

Table 4.2 shows the computational savings achieved from calculating the computational costs, for the standard MC simulation and the multilevel MC simulation. The costs are computed using the expressions (3.1.3) and (3.1.4). The savings are given by the ratio that is, savings = standard cost / multilevel cost. As the accuracy increases, the computational savings increases.

4.1.2 Application to the Asian option

The Asian option is an option style where the payoff is determined by the average asset price over a fixed period of time, unlike the European option where the payoff is determined by the asset price at the expiration date. Therefore, the Asian option has a payoff (discounted) of

$$P = e^{-rT} \max(\bar{S} - K, 0),$$

where $\bar{S} = T^{-1} \int_0^T S(t) dt$. The simplest approximation of \bar{S} is given by Giles in [31] as

$$\bar{S} \cong T^{-1} \sum_{n=1}^{n_T} \frac{1}{2} h (\hat{S}_n + \hat{S}_{n+1}),$$

where $n_T = T/h$ is the number of timesteps. Figure 4.3 shows the numerical results for parameters $S(0) = 1, K = 1, T = 1, \sigma = 0.2$ and $r = 0.05$ for a geometric Brownian motion with Asian option.

This approximation takes the average value of the coarser and finer values of the approximated prices found with the Euler discretisation. In the numerical calculation, the option prices for the coarser and finer grids are found separately and then the final payoff is found by subtracting the payoff at the finer grid and the payoff at the coarser grid discounted at e^{-rT} .

Figure 4.3: Geometric Brownian motion with Asian option (option value ≈ 0.057641).

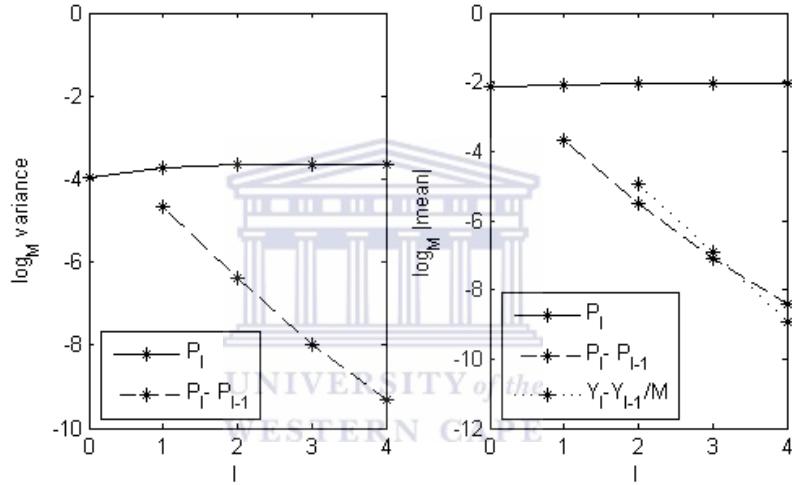


Figure 4.4: Effect of ε for Geometric Brownian motion with Asian Option

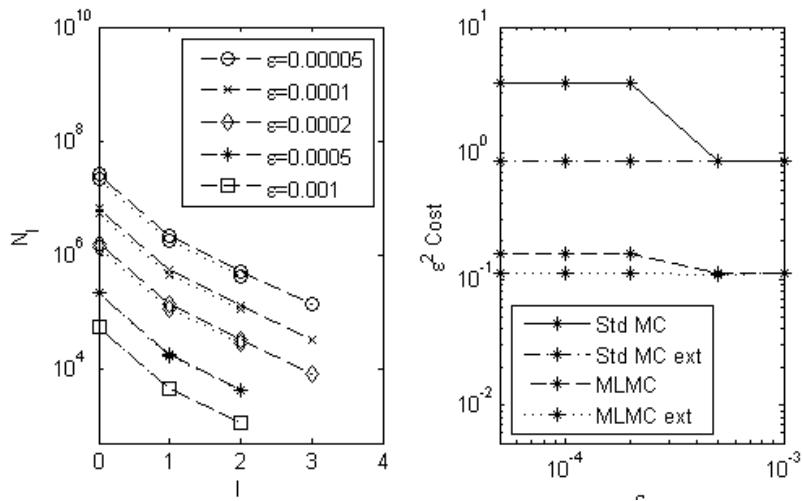


Figure 4.3 presents results from the multilevel MC simulation for the Asian option with Euler discretisation. The behaviours of the variance and the mean $\mathbb{E}[\hat{P}_l - \hat{P}_{l-1}]$

are similar to those of the European option. The slopes for the two plots for $\log |\hat{P}_l - \hat{P}_{l-1}|$ are both equal to -1 , hence the convergence is $O(h_l)$ in both cases. The plot (4.4) shows the result of $\varepsilon^2 C$ against ε . Similar to the European option, $\varepsilon^2 C$ should be proportional to $(\log \varepsilon)^2$. We observe this in Table 4.3.

Table 4.3: $\varepsilon^2 C / (\log \varepsilon)^2$ for the Asian option.

$\varepsilon^2 C$	$(\log \varepsilon)^2$	ratio
0.092335	47.71708299430558	1.935051227e-03
245.74815	57.773718199472874	4.253632235
0.1229084	72.54257985990712	1.694293203e-03
1230.129	84.83036976765435	14.50104489
0.153063137	98.07906570323802	1.560609661e-03

Table 4.4: MC costs, Multilevel MC (MLMC) costs and their respective savings.

ε	Standard cost	MC(MLMC) cost	savings
0.00100	2.532366e+005	92335	2.742585
0.0005	4.241193e+006	4.914963e+005	8.629146
0.0002	2.650746e+007	3072710	8.626736
0.0001	1.060298e+008	1.230129e+007	8.619410
0.00005	1.725695e+009	61225255	28.186005

Similarly, the savings for the Asian option are computed as standard cost/multilevel cost. The results indicate that there is a significant amount of savings as decreases.

4.1.3 Application to the Lookback option

The results below are for the lookback option.

$$P = \exp(-rT) \left(S(1) - \min_{0 < t < 1} S(t) \right).$$

The minimum value of $S(t)$ over the path is approximated numerically by

$$\hat{S}_{\min,l} = \left(\min_n \hat{S}_n \right) (1 - \beta^* \sigma \sqrt{h_l})$$

where $\beta \approx 0.5826$ is a corrective constant (see [31]).

As we can see, the results follow the same pattern as in the previous two cases.

Figure 4.5: Geometric Brownian motion with lookback option (option value ≈ 0.172175).

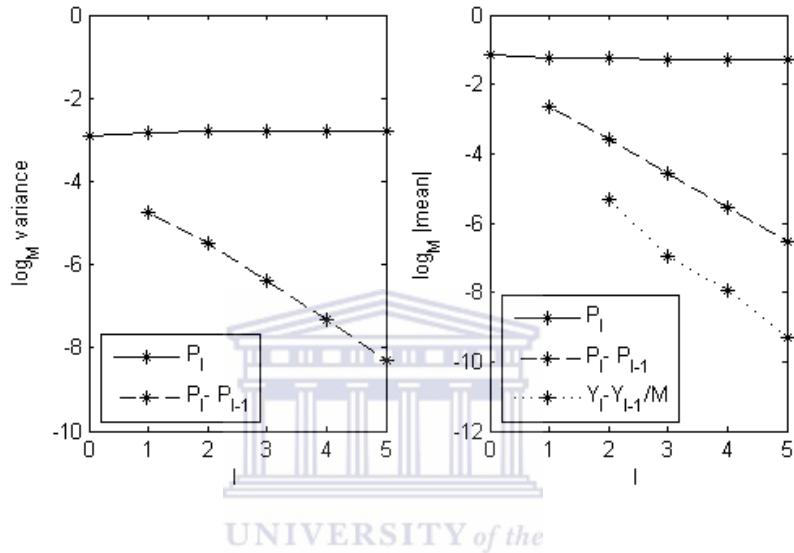


Figure 4.6: Effect of ε for Geometric Brownian motion with lookback option.

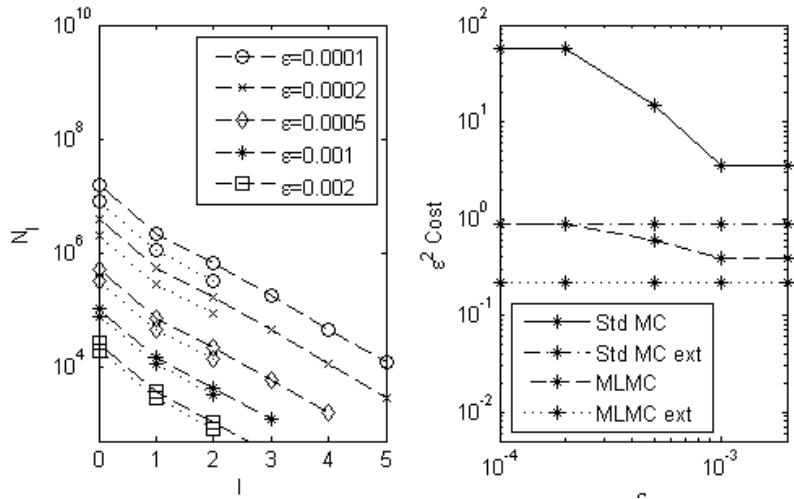


Table 4.5: $\varepsilon^2 C / (\log \varepsilon)^2$ for the lookback option.

$\varepsilon^2 C$	$(\log \varepsilon)^2$	ratio
0.09559625	47.71708299430558	2.003396771e-03
0.09608	57.773718199472874	1.663039926e-03
0.09625264	72.54257985990712	1.326843355e-03
0.2189451	84.83036976765435	2.580975429e-03
0.219998525	98.07906570323802	2.243073213e-03

Table 4.6: MC costs, Multilevel MC (MLMC) costs and their respective savings.

ε	Standard cost	MC(MLMC) cost	savings
0.00100	8.933272e+005	9.559625e+004	9.344793
0.0005	3.573309e+006	384320	9.297743
0.0002	5.786368e+007	2.406316e+006	24.046583
0.0001	1.451447e+009	2.189451e+007	66.292735
0.00005	5.805788e+009	8.799941e+007	65.975299

4.2 An improvement on the Multilevel approach

UNIVERSITY of the

WESTERN AUSTRALIA

The method of discretisation used in the previous section is the Euler scheme. In this section however, an alternative method that improves the speed of convergence from a strong order of 0.5 (via the Euler scheme) to order 1 is the Milstein scheme which is simply the addition of a correction term to the Euler scheme by the application of the Ito formula. The Milstein approximation for X is therefore given below

$$X_{t+h} \approx X_t + a(X_t)h + b(X_t)(B_{t+h} - B_t) + \frac{1}{2}b'(X_t)b(X_t)[(B_{t+h} - B_t)^2 - h], \quad (4.2.3)$$

where $B_{t+h} - B_t = \Delta B = \sqrt{h}Z$ with $Z \sim \mathcal{N}(0, 1)$. For a stock price S following a geometric Brownian motion,

$$S_{t+\delta t} \approx S_t + rS\delta t + \sigma S\sqrt{\delta t}Z + \frac{1}{2}\sigma^2 S[(\sqrt{\delta t}Z)^2 - \delta t], \quad (4.2.4)$$

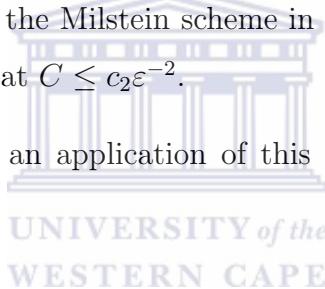
where from (4.2.3) we set $X_t = S_t, h = \delta t, a(X_t) = rS$ and $b(X_t) = \sigma t$.

The work of Giles, seen in [31] showed that a reduction in the computational cost from $\mathcal{O}(\varepsilon^{-3})$ to $\mathcal{O}(\varepsilon^{-2}(\log \varepsilon)^2)$ is possible. His next paper ‘Improved multilevel Monte Carlo convergence using the Milstein scheme’ [30], however, showed that this same computational cost can be improved to $\mathcal{O}(\varepsilon^{-2})$ using the Milstein scheme.

Using the same multilevel Monte Carlo algorithm in the previous section, we apply the Milstein scheme and choose the refinement factor M such that previous levels have half the number of timesteps as current levels that is, $M = 2$.

From Giles paper [30], a bound for the computational complexity C for a multilevel method was defined using the Milstein scheme in such a way that for any $\varepsilon < e^{-1}$, there exists $c_2 > 0$ such that $C \leq c_2 \varepsilon^{-2}$.

We will therefore look at an application of this scheme to one of the previously discussed options.



4.2.1 Application to the European option with the Milstein scheme

Using the same parameters used in the previous section, that is, $S = 1, K = 1, r = 0.05, \sigma = 0.2$ and $T = 1$, Figure 4.7 shows the plots of the results from the multilevel Monte Carlo simulation using the Milstein scheme. The left plot shows the behaviour of the logarithm base 2 variance of \hat{P}_l and $\hat{P}_l - \hat{P}_{l-1}$ at different grid levels. $\log |\hat{P}_l - \hat{P}_{l-1}|$ has a gradient of -2 approximately, which indicates that the variance is proportional to M^{-2} , therefore $V_l = O(h_l^2)$. The right plot shows the logarithm taken in base 2 of the absolute mean at different levels. The mean $\mathbb{E}[\hat{P}_l - \hat{P}_{l-1}]$ has a gradient of -1 approximately at $l = 8$ and $l = 4$, which indicates a convergence of $O(h_l)$.

In the above Figure 4.8, the left plot shows the influence of ε on the number of

Figure 4.7: Geometric Brownian motion with European option (option value ≈ 0.103495) using the Milstein scheme.

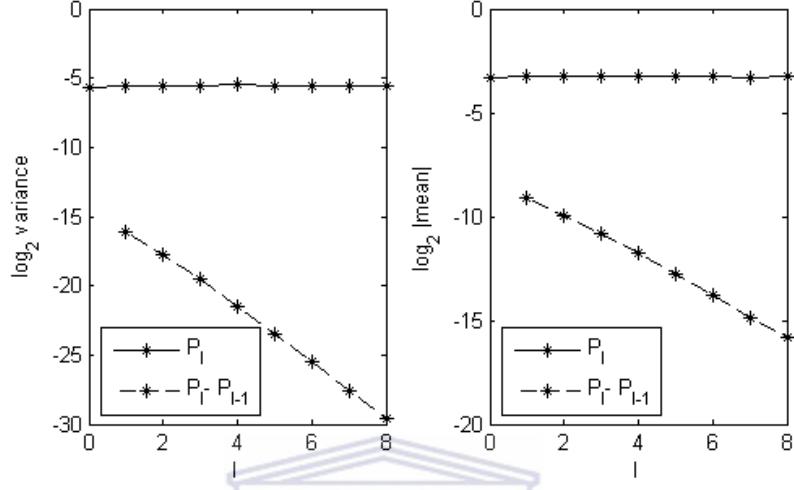
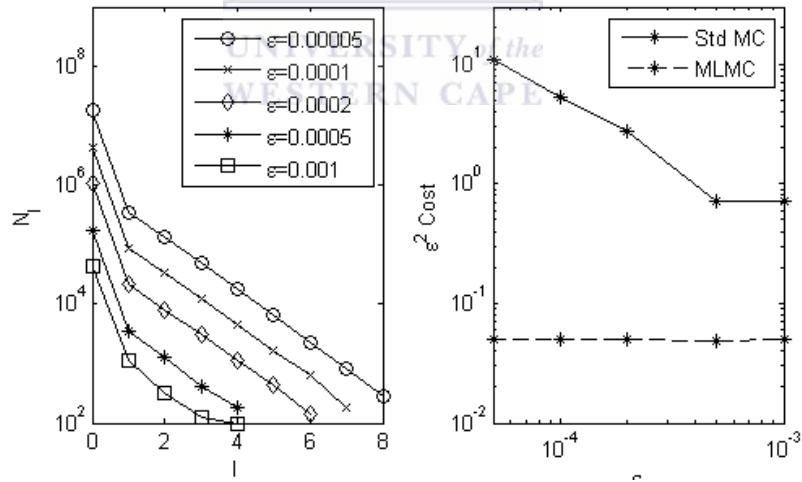


Figure 4.8: Effect of ε for Geometric Brownian motion with European option using the Milstein scheme.



paths required to achieve the desired accuracy. The plot at the right is a result of obtaining five sets of multilevel calculations for different user-specified accuracies, ε . By observation and as we expected, we can see a much significant decrease in the computational cost for the multilevel Monte Carlo method as compared to the standard Monte Carlo method. That is, for the finest ε , we see that the computational cost is reduced by almost eight times of the standard method.

Table 4.7: $\varepsilon^2 C / (\log \varepsilon)^2$ for the European option using the Milstein scheme.

$\varepsilon^2 C$	$(\log \varepsilon)^2$	ratio
0.723408	47.71708299430558	0.015160356
0.723404	57.773718199472874	0.012521333
2.75544576	72.54257985990712	0.03798384
5.24917376	84.83036976765435	0.061878473
10.88329	98.07906570323802	0.110964454

Table 4.8: MC costs, Multilevel MC (MLMC) costs and their respective savings using the Milstein scheme.

ε	Standard cost	MLMC cost	savings
0.00100	723408	50520	14.319240
0.0005	2893616	192234	15.052571
0.0002	68886144	1244999	55.330281
0.0001	524917376	5035415	104.245107
0.00005	4.353316e+009	20267964	214.788044

Table 4.8 shows the savings obtained from calculating the computational cost of the standard MC and that of the MLMC. These savings are given by the ratio of the standard cost to the multilevel cost and it is clear that the savings is higher at the finest accuracy.

In the next chapter, we briefly describe some common Greeks and also describe how this multilevel method is used to compute these Greeks.

Chapter 5

Computing Greeks Using Multilevel Monte Carlo Method



We have previously addressed some aspects of estimating expectations with a view to computing option prices. In this chapter, we investigate the extension of the multilevel Monte Carlo method to the calculation of the derivatives of derivative prices, also known as Greeks.

The Greeks are the quantities representing the sensitivity of the price of derivatives such as options, to a change in underlying parameters on which the value of an instrument or portfolio of financial instruments is dependent. 'Greeks' is used because the most common of these sensitivities are often denoted by Greek letters. Collectively these have also been called the risk sensitivities, risk measures or hedge parameters.

In mathematical finance, Monte Carlo methods are used to compute the price of an option by estimating the expected value $\mathbb{E}(P)$. Here, P is the payoff function that depends on an underlying asset's scalar price $S(t)$ which satisfies an evolution SDE

CHAPTER 5. COMPUTING GREEKS USING MULTILEVEL MONTE CARLO METHOD

of the form

$$dS(t) = a(S, t)dt + b(S, t)dB_t, \quad 0 \leq t \leq T, \quad S(0) \text{ given.} \quad (5.0.1)$$

This is just one use of Monte Carlo in finance. In practice the prices are often quoted and used to calibrate our market models; the option's sensitivities to market parameters, the so-called Greeks, reflect the exposure to different sources of risk. Computing these is essential to hedge portfolios and is therefore even more important than pricing the option itself. This is why the research focuses on getting fast and accurate estimates of Greeks through Monte Carlo simulations.

Derivative prices can often be observed in the market but their sensitivities cannot and so we can say that the accurate calculation of sensitivities is much more important than the calculation of prices itself. Estimation of these derivatives however, pose challenges (theoretically and practically) to Monte Carlo simulation.

To estimate derivatives by simulation, two direct methods are classically used. The first is the *pathwise* method which is based on the relationship between the security payoff and the parameter of interest. Differentiating this relationship leads, under appropriate conditions, to an unbiased estimator for the derivative of the security price. On the other hand, the *likelihood ratio* method is based on the relationship between the probability density of the price of the underlying security and the parameter of interest. These methods have been studied in the literature for discrete-event simulation, but have not been given much attention in financial applications.

The following is a list of the most common Greeks.

1. The **Delta** (Δ) of a financial derivative is the rate of change of the value with respect to the value of the underlying security, in symbols

$$\Delta = \frac{\partial V}{\partial S}.$$

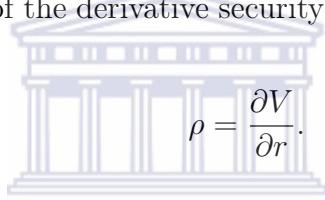
2. The **Gamma** (Γ) (or convexity factor) of a derivative is the sensitivity of (Δ) with respect to S , in symbols

$$\Gamma = \frac{\partial \Delta}{\partial S} = \frac{\partial^2 V}{\partial^2 S}.$$

3. The **Theta** (Θ) (or time decay) of a European claim with value function $V(S, t)$ is defined as

$$\Theta = \frac{\partial V}{\partial t}.$$

4. The **rho** (ρ) (or interest rate factor) of a derivative security is the rate of change of the value of the derivative security with respect to the interest rate, in symbols



$$\rho = \frac{\partial V}{\partial r}.$$

5. The **Vega** Λ (or volatility factor) of a derivative security is the rate of change of the value of the derivative security with respect to the volatility of the underlying asset, in symbols

$$\Lambda = \frac{\partial V}{\partial \sigma}.$$

5.1 Greek estimates for European call options

In this subsection, the pathwise method is developed for estimating Greeks. The method is introduced through an example. The likelihood ratio method will not be discussed here, the interested reader is referred to [29].

Consider the price p of a European call option on a dividend paying asset that follows a lognormal diffusion. In particular, assume that the risk neutralised price

of the underlying asset, S_t , satisfies the SDE

$$dS_t = S_t[(r - \delta)dt + \sigma dB_t] \quad (5.1.2)$$

where B is a standard Brownian motion process. In Equation (5.1.2), r is the riskless interest rate, δ is the dividend rate, and $\sigma > 0$ is the volatility parameter. Under the risk neutral measure, $\ln(S_T/S_0)$ is normally distributed with mean $(r - \delta - \sigma^2/2)T$ and variance $\sigma^2 T$. The option has a strike price of K and matures at time $T > 0$, with the current time taken to be $t = 0$. By Black-Scholes, the option price is given by

$$p = \mathbb{E}[e^{-rT} \max(S_T - K, 0)]. \quad (5.1.3)$$

\mathbb{E} is the expectation operator under the risk neutral measure.

To describe the application of the *pathwise* method, we consider the problem of estimating *vega*, which is $dp/d\sigma$. This we do by defining the discounted payoff

$$P = e^{-rT} \max(S_T - K, 0), \quad (5.1.4)$$

(so that $p = \mathbb{E}[P]$) and examining how changes in σ determine changes in P . Equations (5.1.2) and (5.1.3) tell us that σ affects P only through S_T , hence we begin by checking the dependence of S_T on σ .

The lognormal random variable S_T can be represented as

$$S_T = S_0 e^{(r-\delta-\sigma^2/2)T + \sigma\sqrt{T}Z}, \quad (5.1.5)$$

where Z is a standard normal random variable. Consequently,

$$\frac{dS_T}{d\sigma} = S_T(-\sigma T + \sqrt{T}Z) \frac{S_T}{\sigma} \left[\ln(S_T/S_0) - \left(r - \delta + \frac{1}{2}\sigma^2 \right) T \right]. \quad (5.1.6)$$

This tells us how a little variation in σ affects S_T . Now consider the effect of a little variation in S_T on P . If $S_T \geq K$, then the option is in the money and any increase Δ in S_T translates into an increase $e^{-rT}\Delta$ in P . If however, $S_T < K$, then $P = 0$, and P remains so for all sufficiently small changes in S_T . Indeed, $S_T < K \Leftrightarrow \exists \varepsilon > 0 : (S_T - \varepsilon, S_T + \varepsilon) \subset (-\infty, K)$. Thus we arrive at the formal expression

$$\frac{dP}{dS_T} = e^{-rT}1_{[S_T \geq K]}, \quad (5.1.7)$$

Combining (5.1.6) and (5.1.7) gives via the chain rule,

$$\frac{dP}{d\sigma} = \frac{dP}{dS_T} \frac{dS_T}{d\sigma} = e^{-rT}1_{[S_T \geq K]} \frac{S_T}{\sigma} \left[\ln(S_T/S_0) - \left(r - \delta + \frac{1}{2}\sigma^2 \right) T \right]. \quad (5.1.8)$$

Observe that each term in this expression is easily evaluated in a simulation, making the estimator $dP/d\sigma$ relatively easy to use. Moreover, it can be shown that this estimator is unbiased, that is,

$$\mathbb{E} \left[\frac{dP}{d\sigma} \right] = \frac{dp}{d\sigma}.$$

A similar argument leads to an estimator of *delta*, the derivative of the option price with respect to the initial price of the underlying asset. As before, we have

$$\frac{dP}{dS_0} = \frac{dP}{dS_T} \frac{dS_T}{dS_0} = e^{-rT}1_{[S_T \geq K]} \frac{dS_T}{dS_0}. \quad (5.1.9)$$

Furthermore, from Equation (5.1.5) we find that

$$\frac{dS_T}{dS_0} = e^{(r-\delta-\sigma^2/2)T+\sigma\sqrt{T}Z} = \frac{S_T}{S_0}.$$

Substituting this into Equation (5.1.9), we arrive at the estimator

$$\frac{dP}{dS_0} = e^{-rT} \mathbf{1}_{[S_T \geq K]} \frac{S_T}{S_0}. \quad (5.1.10)$$

The estimator is also unbiased, that is,

$$\mathbb{E} \left[\frac{dP}{dS_0} \right] = \frac{dp}{dS_0}.$$

Similar arguments can be used to develop derivative estimates for options with path dependencies (e.g. Asian options)

The above analysis and therefore the examination of each of the sensitivities could be more succinct under a reasonable restatement of the Black-Scholes formula for the value of a European call option. Most of the material described here is taken from [26, 16]. Set

$$d_1 = \frac{\ln(S/K) - (r + \sigma^2/2)(T - t)}{\sigma\sqrt{T-t}}$$

$$d_2 = \frac{\ln(S/K) - (r - \sigma^2/2)(T - t)}{\sigma\sqrt{T-t}}$$

and then

$$V_C(S, t) = S\Phi(d_1) - Ke^{-r(T-t)}\Phi(d_2).$$

It is not difficult to see that $d_1 - d_2 = \sigma^2\sqrt{T-t}$

- The Delta $\Delta = \frac{\partial V_C}{\partial S}$. An easy calculation with the necessary simplifications

gives

$$\begin{aligned}
 \Delta &= \frac{\partial V_C}{\partial S} \\
 &= \Phi(d_1) + S\Phi'(d_1)\frac{\partial d_1}{\partial S} - Ke^{-r(T-t)}\Phi'(d_2)\frac{\partial d_2}{\partial S} \\
 &= \Phi(d_1) + \frac{\exp(-d_1^2/2)}{\sqrt{2\pi}\sqrt{T-t}} \left[1 - \frac{K\exp(-r(T-t))}{S} \exp(\ln(S/K) + r(T-t)) \right] \\
 &= \Phi(d_1).
 \end{aligned}$$

Note that since $0 < \Phi(d_1) < 1$ (for all reasonable values of d_1), $\Delta > 0$, and so the value of a European call option is always increasing as the underlying security value increases.

Delta Hedging:

Notice that for any sufficiently differentiable function F ,



$$F(S_1) - F(S_2) \approx \frac{dF}{dS}(S_1 - S_2).$$

Therefore, for the Black-Scholes formula for a European call option, using the current notation $\Delta = \partial V / \partial S$,

$$(V(S_1) - V(S_2)) - \Delta(S_1 - S_2) \approx 0,$$

or equivalently for small changes in the security price from S_1 to S_2 ,

$$V(S_1) - \Delta(S_1) \approx V(S_2) - \Delta(S_2).$$

In financial language, we express this as: “long 1 derivative, short Δ units of the underlying asset is market neutral for small changes in the asset value.”

We say that the sensitivity Δ of the financial derivative value with respect to the asset value gives the hedge-ratio. The hedge-ratio is the number of

short units of the underlying asset which combined with a call option will offset immediate market risk. After a change in the asset value, $\Delta(S)$ will also change, and so we will need to dynamically adjust the hedge-ratio to keep pace with the changing asset value. Thus $\Delta(S)$ as a function of S provides a dynamic strategy for hedging against risk.

- The Gamma $\Gamma = \frac{\partial^2 V}{\partial^2 S}$. The concept of Gamma is important when the hedged portfolio cannot be adjusted continuously in time according to $\Delta(S(t))$. If Gamma is small then Delta changes very little with S . This means the portfolio requires only infrequent adjustments in the hedge-ratio. However, if Gamma is large, then the hedge-ratio Delta is sensitive to changes in the price of the underlying security. If Delta is the "speed" at which option prices change, one can think of Gamma as the "acceleration." Options with the highest gamma are the most responsive to changes in the price of the underlying stock.

According to the Black-Scholes formula, we have

WESTERN CAPE

$$\Gamma = \frac{1}{S\sqrt{2\pi}\sigma\sqrt{T-t}} \exp(-d_1^2/2).$$

Notice that $\Gamma > 0$, so the call option value is always concave-up with respect to S .

- The Theta $\Theta = \frac{\partial V}{\partial t}$ is the rate of change with respect to the real (or calendar) time, some other authors define the rate of change with respect to the time-to-expiration $T - t$. The Theta of a claim is sometimes referred to as the time decay of the claim. For a European call option on a non-dividend-paying stock,

$$\Theta = -\frac{S\sigma}{2\sqrt{T-t}} \frac{\exp(-d_1^2/2)}{\sqrt{2\pi}} - rK \exp(-r(T-t)) \Phi(d_2).$$

Note that Θ for a European call option is negative, so the value of a European

call option is decreasing as a function of time. Theta does not act like a hedging parameter as do Delta and Gamma. Although there is uncertainty about the future stock price, there is no uncertainty about the passage of time. It does not make sense to hedge against the passage of time on an option.

- The rho ρ of a derivative security is the rate of change of the value of the derivative security with respect to the interest rate. It measures the sensitivity of the value of the derivative security to interest rates. For a European call option on a non-dividend paying stock,

$$\rho = K(T - t) \exp(-r(T - t))\Phi(d_2),$$

so ρ is always positive. An increase in the risk-free interest rate means a corresponding increase in the derivative value.

- The Vega Λ of a derivative security is the rate of change of value of the derivative with respect to the volatility of the underlying asset. (Note, some authors also denote Vega by variously λ , κ and σ and refer to Vega by the corresponding proper Greek letter name). For a European call option on a non-dividend-paying stock,

$$\Lambda = S\sqrt{T - t} \frac{\exp(-d_1^2/2)}{\sqrt{2\pi}},$$

so the Vega is always positive. An increase in the volatility will lead to a corresponding increase in the call option value.

It would be incorrect to give the impression that traders continuously balance their portfolios to maintain Delta neutrality, Gamma neutrality, Vega neutrality, and so on as would be suggested by the continuous mathematical formulas presented above. In practice, transaction costs make frequent balancing expensive. Rather than try to eliminate every risk, an option trader usually concentrates on assessing risks and deciding whether they are acceptable. Traders tend to use Delta, Gamma, and Vega measures to quantify the different aspects of risk in their portfolios.

5.2 Monte Carlo Simulation of Greeks

Let us briefly recall two classic methods used to compute Greeks in a Monte Carlo setting.

5.2.1 Pathwise sensitivities

Let $\hat{S} = (\hat{S}_k)_{k \in [0, N]}$ be the simulated values of the asset at the discretisation times and $\hat{W} = (\hat{W}_k)_{k \in [1, N]}$ be the corresponding set of independent Brownian increments. The value of the option V is estimated by \hat{V} defined as

$$V = \mathbb{E}[P(S)] \approx \hat{V} = \mathbb{E}[P(\hat{S})] = \int P(\hat{S}) p(\theta, \hat{S}) d\hat{S}.$$

Assuming that the payoff $P(\hat{S})$ is Lipschitz, we can use the chain rule and write

$$\frac{\partial \hat{V}}{\partial \theta} = \frac{\partial}{\partial \theta} \int P(\hat{S}(\theta, \hat{W})) p(\hat{W}) d\hat{W} = \int \frac{\partial P(\hat{S})}{\partial \hat{S}} \frac{\partial \hat{S}(\theta, \hat{W})}{\partial \theta} p(\hat{W}) d\hat{W},$$

where $d\hat{W} = \prod_{k=1}^N d\hat{W}_k$ and $p(\hat{W}) = \prod_{k=1}^N p(\hat{W}_k)$ is the joint probability density function of the normally distributed independent increments $(\hat{W}_k)_{k \in [1, N]}$.

We obtain $\frac{\partial \hat{S}}{\partial \theta}$ by differentiating the discretisation of (5.0.1) with respect to θ and iterating the resulting formula. The limitation of this technique is that it requires the payoff to be Lipschitz and piecewise differentiable.

5.2.2 Likelihood Ratio method

The Likelihood Ratio Method consists in writing V as

$$V = \mathbb{E}[P(\hat{S})] = \int P(\hat{S}) p(\theta, \hat{S}) d\hat{S}.$$

The dependence on θ comes through the probability density function $p(\theta, \hat{S})$; assuming some conditions discussed in [35], we can write

$$\begin{aligned}\frac{\partial V}{\partial \theta} &= \int P(\hat{S}) \frac{\partial p(\hat{S})}{\partial \theta} d\hat{S} \\ &= \int P(\hat{S}) \frac{\partial \log p(\hat{S})}{\partial \theta} p(\hat{S}) d\hat{S} \\ &= \mathbb{E} \left[P(\hat{S}) \frac{\partial \log p(\hat{S})}{\partial \theta} \right],\end{aligned}\tag{5.2.11}$$

with $d\hat{S} = \prod_{k=1}^N \hat{S}_k$ and $p(\hat{S}) = \prod_{k=1}^N p(\hat{S}_k | \hat{S}_{k-1})$. The main limitation of the method is that the estimator's variance is $\mathcal{O}(N)$, becoming infinite as we refine the discretisation.

5.3 Valuation of Greeks using Multilevel Monte Carlo Simulation

By combining the elements of the previous sections together, we write

$$\frac{\partial V}{\partial \theta} = \frac{\partial \mathbb{E}(P)}{\partial \theta} \approx \frac{\partial \mathbb{E}(\hat{P}_L)}{\partial \theta} = \frac{\partial \mathbb{E}(\hat{P}_0)}{\partial \theta} + \sum_{l=1}^L \frac{\partial \mathbb{E}(\hat{P}_l - \hat{P}_{l-1})}{\partial \theta}.\tag{5.3.12}$$

Then we define the multilevel estimators

$$\hat{Y}_0 = N_0^{-1} \sum_{i=1}^M \frac{\partial \hat{P}_0^{(i)}}{\partial \theta} \quad \text{and} \quad \hat{Y}_l = N_l^{-1} \sum_{i=1}^M \left(\frac{\partial \hat{P}_l^{(i)}}{\partial \theta} - \frac{\partial \hat{P}_{l-1}^{(i)}}{\partial \theta} \right),\tag{5.3.13}$$

where $\frac{\partial \hat{P}_0}{\partial \theta}, \frac{\partial \hat{P}_{l-1}}{\partial \theta}, \frac{\partial \hat{P}_l}{\partial \theta}$ are computed with the techniques presented in the previous section.

Chapter 6

Concluding remarks and scope for future research



In this thesis, we presented applications of multilevel Monte Carlo method for pricing standard and non-standard options. In particular, we showed how this method can be used to improve the efficiency of the standard Monte Carlo method when applied to European, Asian and Lookback options. We then discussed how this method can be used to evaluate Greeks. Improved results are found in all cases. Computed result shows that a good accuracy can be achieved with a large number of samples. Furthermore, the computational convergence rate verifies the theoretical convergence rate presented in literature.

We tested the standard Monte Carlo simulation by initially running a simulation of few paths. We then changed the number of paths and observed how the results vary. We found that increasing the number of paths will reduce the sampling error at a rate of $1/\sqrt{m}$.

The result from the Euler scheme shows a strong convergence order of $1/2$. The refinement to this was the Milstein scheme which has a strong convergence order of 1 . The error is thus significantly reduced. The results presented for the improved

CHAPTER 6. CONCLUDING REMARKS AND SCOPE FOR FUTURE RESEARCH

Monte Carlo simulation show an increase in the efficiency of the model.

As far as the scope for future research is concerned, we intend to study more theoretical results on the development of multilevel Monte Carlo methods. We also intend to apply them to solve other classes of option pricing problems.



Appendix A

Multilevel Monte Carlo Algorithm

Below, we list the main steps of the multilevel Monte Carlo algorithm:

1. Begin with $L = 0$.
2. Calculate the initial estimate of V_L using 100 samples.
3. Determine optimal N_l using Equation (4.1.1).
4. Generate additional samples for $l = 0, 1, \dots, L$.
5. Test for convergence using Equation (4.1.2).
6. If not converged, set $L := L + 1$ and continue from step 2.

Bibliography

- [1] A. Brandt, M. Galun and D. Ron, Optimal multigrid algorithms for calculating thermodynamic limits, *Journal of Statistical Physics* **74(1-2)** (1994) 313-348.
- [2] A. Brandt and V. Ilyin, Multilevel Monte Carlo methods for studying large scale phenomena in fluids, *Journal of Molecular Liquids* **105(2-3)** (2003) 245-248.
- [3] A. Szepessy, R. Tempone and G. E. Zouraris, Adaptive weak approximation of stochastic differential equations, *Communications on Pure and Applied Mathematics* **54(10)** (2001) 1169-1214.
- [4] A. Dzougoutov, K.S. Moon, E. von Schwerin, A. Szepessy, and R. Tempone, Adaptive Monte Carlo algorithms for stopped diffusion, multiscale methods in science and engineering, *Computing in Science and Engineering* **44** (2005) 59-88.
- [5] A. Etheridge, A course in financial calculus, *Cambridge University Press* (2002).
- [6] A. Ferreiro-Castilla, A.E. Kyprianou, R. Scheichl and G. Suryanarayana, Multilevel Monte Carlo simulation for Lèvy processes based on the Wiener-Hopf factorisation, *Stochastic Processes and their Applications* **124(2)** (2014) 985-1010.

BIBLIOGRAPHY

- [7] C. J. Gittelson et al., The multilevel Monte Carlo finite element method for a stochastic Brinkman problem, *Numerische Mathematik* **125(2)** (2013) 347-386.
- [8] C. Joy, P. P. Boyle and K. S. Tan, Quasi-Monte Carlo methods in numerical finance, *Management Science* **42(6)** (1996) 926-938.
- [9] D. F. Anderson and D. J. Higham, Multilevel Monte Carlo for continuous time Markov chains with applications in biochemical kinetics, *Multiscale Modeling and Simulation* **10(1)** (2012) 146-179.
- [10] D. J. Higham, Stochastic ordinary differential equations in applied and computational mathematics, *IMA Journal of Applied Mathematics* **76** (2011) 449-474.
- [11] D. Talay and L. Tubaro, Expansion of the global error for numerical schemes solving stochastic differential equations, *Stochastic Analysis and Applications* **8** (1990) 94-120.
- [12] E. Clément, D. Lamberton and P. Protter, An analysis of a least squares regression method for American option pricing, *Finance and Stochastics* **6(4)** (2002) 449-471.
- [13] F. Black and M. Scholes, The pricing of options and corporate liabilities, *Journal of Political Economy* **81(3)** (1973) 637-654.
- [14] F. A. Longstaff and E. S. Schwartz, Valuing American options by simulation: a simple least-squares approach, *Review of Financial Studies* **14(1)** (2001) 113-147.
- [15] H. Hoel, E. Von Schwerin, A. Szepessy and R. Tempone, Adaptive multi-level Monte Carlo simulation, *Numerical Analysis of Multiscale Computations* (2012) 217-234.

BIBLIOGRAPHY

- [16] J. C. Hull, Options, futures and other derivative securities, *Pearson education* (2009).
- [17] J. Palczewski and A. Palczewski, Lecture notes on Monte Carlo simulation, http://mimuw.edu.pl/~apalczew/CFP_lecture4.pdf
- [18] J. Palczewski, Milstein scheme and convergence, *Computations in Finance, MATH 5350* Jan. (2009).
- [19] J. van Resenborg, E. Janse and G. M. Torrie, Estimation of multidimensional integrals: is Monte Carlo the best method? *Journal of Physics A: Mathematical and General* **26(4)** (1993) 943.
- [20] K.A. Cliffe, M.B. Giles, R. Scheichl and A.L. Teckentrup, Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients, *Computing and Visualization in Science* **14(1)** (2011) 3-15.
- [21] K. S. Moon, A. Szepessy, R. Tempone and G. E. Zouraris, Convergence rates for adaptive weak approximation of stochastic differential equations, *Stochastic Analysis and Applications* **23(3)** (2005) 511-558.
- [22] K. S. Moon, E. von Schwerin, A. Szepessy and R. Tempone, An adaptive algorithm for ordinary, stochastic and partial differential equations, recent advances in adaptive computation, *Contemporary Mathematics* **383** (2005) 325-343.
- [23] L. C. G. Rogers, Monte Carlo valuation of American options, *Mathematical Finance* **12(3)** (2002) 271-286.
- [24] L. Stentoft, Assessing the least squares Monte Carlo approach to American option valuation, *Review of Derivatives Research* **7(2)** (2004) 129-168.
- [25] L. Stentoft, Convergence of the least squares Monte Carlo approach to American option valuation, *Management Science* **50(9)** (2004) 1193-1203.

BIBLIOGRAPHY

- [26] M. Avellaneda and P. Laurence, Quantitative modeling of derivative securities, *CRC Press* (1999).
- [27] M. Broadie, P. Glasserman and G. Jain, Enhanced Monte Carlo estimates for American option prices, *The Journal of Derivatives* **5(1)** (1997) 25-44.
- [28] M. B. Giles and B. J. Waterhouse, Multilevel quasi-Monte Carlo path simulation, *Advanced Financial Modelling, Radon series on Computational and Applied Mathematics* (2009) 165-181.
- [29] M. Broadie and P. Glasserman, Estimating security price derivatives using simulation, *Management Science* (1996) 269-285.
- [30] M. B. Giles, Improved multilevel Monte Carlo convergence using the Milstein scheme, *Monte Carlo and quasi-Monte Carlo methods* (2006) Springer Berlin Heidelberg (2008) 343-358.
The logo of the University of Western Cape, featuring a classical building facade with four columns and a triangular pediment above the text "UNIVERSITY OF WESTERN CAPE".
- [31] M. B. Giles, Multilevel Monte Carlo path simulation, *Operations Research* **56(3)** (2008) 607-617.
- [32] M. B. Giles and C. Reisinger, Stochastic finite differences and multilevel Monte Carlo for a class of SPDEs in finance, *SIAM Journal on Financial Mathematics* **1(3)** (2012) 575-592.
- [33] P. P. Boyle, Options: a Monte Carlo approach, *Journal of Financial Economics* **4(3)** (1977) 323-338.
- [34] P. P. Boyle, M. Broadie and P. Glasserman, Monte Carlo methods for security pricing, *Journal of Economic Dynamics and Control* **21(8)** (1997) 1267-1321.
- [35] P. Glasserman, Monte Carlo methods in financial engineering, *Springer-Verlag* **53** (2004).
- [36] P. Jäckel and R. Bubley, Monte Carlo methods in finance, *J. Wiley* (2002).

BIBLIOGRAPHY

- [37] Q. Jia, Pricing American options using Monte Carlo methods, *U.U.D.M Project Report, Uppsala University* (2009) 1-31.
- [38] R. Caflisch, Monte Carlo and Quasi-Monte Carlo methods, *Acta Numerica* **7** (1998) 1-49.
- [39] S. M. Ross, Stochastic processes, *John Wiley & Sons* **2** (1996).
- [40] S. Dereich and F. Heidenreich; A multilevel Monte Carlo algorithm for Lévy-driven stochastic differential equations, *Stochastic Processes and their Applications* **121(7)** (2011) 1565-1587.
- [41] S. Heinrich, Multilevel Monte Carlo methods, *Lecture notes in Computer Science* **2179** (2001) 3624-3651.
- [42] S. K. Zaremba, The mathematical basis of Monte Carlo and quasi-Monte Carlo methods, *SIAM Review* **10(3)** (1968) 303-314.
- [43] V. Bally and D. Talay, The law of the Euler scheme for stochastic differential equations, *Probability Theory and Related Fields* **104(1)** (1996) 43-60.