MÄLARDALEN UNIVERSITY
SWEDEN

School of Education, Culture and Communication
Division of Mathematics and Physics

BACHELOR'S DEGREE PROJECT IN MATHEMATICS

Multilevel Monte Carlo Simulation for American Option Pricing

by

Sabina Colakovic and Viktor Ågren

MAA322 — Examensarbete i matematik för kandidatexamen

DIVISION OF MATHEMATICS AND PHYSICS
MÄLARDALEN UNIVERSITY
SE-721 23 VÄSTERÅS, SWEDEN

MÄLARDALEN UNIVERSITY
SWEDEN

School of Education, Culture and Communication
Division of Mathematics and Physics

MAA322 — Bachelor’s Degree Project in Mathematics

Date of presentation:
1 June 2021

Project name:
Multilevel Monte Carlo Simulation for American Option Pricing

Authors:
Sabina Colakovic and Viktor Ågren

Version:
19th September 2022

Supervisor:
Doghonay Arjmand

Reviewer:
Milica Rancic

Examiner:
Daniel Andrén

Comprising:
15 ECTS credits

Abstract

In this thesis, we center our research around the analytical approximation of American put options with the Multilevel Monte Carlo simulation approach. The focus lies on reducing the computational complexity of estimating an expected value arising from a stochastic differential equation. Numerical results showcase that the simulations are consistent with the theoretical order of convergence of Monte Carlo simulations. The approximations are accurate and considerately more computationally efficient than the standard Monte Carlo simulation method.

Keywords: Multilevel Monte Carlo simulation, Stochastic Differential Equations, Option pricing.

Acknowledgements

We would like to take the opportunity to show our gratitude to our supervisor, Doghonay Arjmand, for all his guidance during the completion of this thesis. We would also like to thank Milica Rancic for all the comments and suggestions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The pricing of American options is still an active field of research within the quantitative finance world. Dissimilar to European options, there is no analytical solution for the value of the option price [5]. American options can be exercised at any time up until the expiration date. The most important evaluation of an American option is to estimate the optimal exercising time correctly to get the highest possible payoff [19]. Hertz [20] introduced the first Monte Carlo simulation in finance in his article, where he advocated for the need of a new concept as the conventional methods were not satisfactory to give accurate estimates. This statement is a well-discussed topic regarding the future of simulation in pricing options. Tilley [31] preformed the first Monte Carlo simulation on American options and in his article he attempted to weaken the arguments against the use of simulations of pricing American options. His work included an eight-step description of the algorithm and it includes bundling parameters alongside with a backwards approach of processing each time step. Carriere [16] attempted to improve the algorithm by the theory of optimal stopping with a sequential regression algorithm. His result shows that there is no difference in the formula used for American and European options. The well documented algorithm used to approximate the price for American put options is showcased in the paper.

Barraquand and Martineau [6] focused on the numerical valuation in higher dimensions. This is something previous algorithms could not calculate because of the exponentially increasing memory demand. The approach is successful by combining the Monte Carlo method simulation with state aggregation. The aggregation works by introducing partitioning the state space into cells. Appropriate partings results in the approximation that would be close to optimal exercising. Stratifying these states makes it possible to determine the strategies that result in the optimal payoff. Raymar and Zwecher [26] continued to work on the model calculated by Barraquand and Martineau [6]. The improvements were done with a maximum price constraint and partitioning based on the second statistic. Their analysis of the applications showed that the two-factor approach is closer to the benchmark values except for cases where the option is too far out of the money. The use of the second statistic proved to be potent at making approximations for early exercising.

Broadie and Glasserman [9] found an issue when working with a single estimator. In order to work around the problem they created a model that calculates two estimates, where one estimate is biased high and the other is biased low. Subsequently, they should converge between the estimates resulting in a better true price approximation. They found that the relative errors were quite small without a large effort on computation. Broadie, Glasserman and Jain [10] followed up on their previous work [9] with advancements to the algorithm. They complemented it with a confidence interval bounded by the estimates to reduce the variance combined with the use of European options to reduce the branching process, in conjunction with other strategies to get faster convergence rates and shorter simulation times. Carr [15] developed an algorithm to calculate the American option value and the boundaries based on randomization. This included a three-step approach for calculating expected value for a fixed randomly generated parameter. Their results proved to be computationally fast and the option value was approached for a close approximation of the true value. Longstaff and Schwartz [25] in 2000 had a similar approach to [16] and [32] on valuing American options. The unique aspect of their work is the strategy of only calculating the paths where the options are in-the-money. As a results, their LSM algorithm produces an efficient computational process and a true price approximation.

The most naive way to compute the optimal American option value is to exploit Monte Carlo methods. Caflisch and Chaudharry [13] reviewed the basic properties of American options and the difficulties of applying Monte Carlo valuation to American options. This included Branching processes to obtain the upper and lower bounds on the American option price, a Martingale optimization and the Least Squares Monte Carlo (LSM), that provided a direct method for pricing American option. Quasi-random sequences was then used to improve the performance of the LSM method. Their results proved the difficulties involved in applying Monte Carlo evaluation to American options, as well as valuable results in improving Monte Carlo simulations. Birge [7] presented some empirical evidence that Quasi-random streams can produce more accurate results with fewer iterations, compared to the standard Monte Carlo. Quasi-Monte Carlo also offers less fluctuation than standard Monte Carlo. Adding to that, it also makes on-line error approximations possible. Birge's results showed that Quasi-Monte Carlo has an advantage in a simple model where analytical results are available. The advantage may be even greater in more complex models, such as Multilevel Monte Carlo. As Monte Carlo comes together with high computational burden, there have been other attempts in decreasing the computational cost by exploiting more interesting approaches based on Quasi-Monte Carlo methods. Longstaff and Schwartz [25] suggested that their method might be improved by the use of quasi-random points. However, Caflisch and Chaudharry [13] considered that there are two difficulties with this extension of their method. With this extension of the method, there are two potential difficulties. The issue is high dimensional and the prices along the various paths in LSM method, are correlated. Both of which can be troublesome for Quasi-Monte Carlo. Multilevel Monte Carlo, deals with a more advanced variant of Monte Carlo method. Giles [18] developed the Multilevel Monte Carlo for simulations of European option valuation. The results have shown that a Multilevel approach, using a geometric sequence of time steps, can reduce Monte Carlo's path simulations order of complexity, thus reducing the computational burden.

Options are a form of financial contract that grants the investor the right to purchase or sell the underlying assets on or before a certain future date, known as maturity, at a mutually negotiated exercise price, called the strike price. There are several kinds of options. The most common ones are European-, Asian- and American options. In this paper, we will focus on

American options, and the key feature of the American options is that they are continuous time instruments because of their availability to be exercised at any time. Early exercising is the decision to exercise the option prior to the expiration date. Thanks to its exercising time flexibility, the most common options that are traded on exchanges are American.

In the world of quantitative finance, the pricing of American options is still an active field of research. There are no general solutions for the valuation of American options, unlike European options. One often utilizes Monte Carlo simulation in option pricing. However, Multilevel Monte Carlo, as proposed by Giles in [18], was used as an alternative method to reduce complexity and computational cost.

Assume that the stock value $S(t)$ evolves by the following stochastic differential equation, called the Itô geometric Brownian motion [13],

$$dS = \mu S dt + \sigma S dW,$$

where μ and σ are the constant average growth rate and volatility, respectively, and $W = W(t)$ is a standard Brownian motion. The aim of the project is to implement the Multilevel Monte Carlo method for valuing American options. The research objective is to approximate,

$$f_A(t,s) \equiv \max_{t \leq \tau \leq T} E[e^{-r(\tau - t)} \max(K - S(\tau), 0) | S(t) = s],$$

by implementing the Multilevel Monte Carlo method, where K is the strike price, $S(t)$ is the spot price of the underlying asset, τ is the class of admissible stopping times with values in $[t, T]$, r is the risk-free interest rate, and s is the initial stock price. In the present thesis, we aim to extrapolate the ideas presented in [18] for American option valuation.

This report will start with an introduction to the theoretical preliminaries, stochastic differential equations, as well as examples of numerical methods for solving those. Followed by an explanation of the standard Monte Carlo method with error estimates and variance reduction techniques, and the Multilevel Monte Carlo method to price American options. We present and analyze the numerical results in the following chapter, where we investigate the option prices depending on the values of different variables, as well as discuss the convergence rates and complexity for the different methods. Finally, we conclude and review the results, as well as suggest additional methods and further research to improve the algorithm.

Chapter 2

Preliminaries

2.1 Time-stepping Methods for Ordinary Differential Equations

The most commonly utilized time-stepping methods are the *Forward Euler-* and *Backward Euler methods*. Forward Euler has the advantage of including an explicit update equation, making it simpler to incorporate in practice. Backward Euler, on the other hand, necessitates the solution of an implicit equation, so it is more expensive, but it has better stability properties in general.

There are two standard forms for expressing the solution of initial value problems in systems of ordinary differential equations, the first one being

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0, \tag{2.1}$$

where y is the solution assumed to be a differentiable function on the interval $[x_0, \bar{x}]$ to a finite dimensional Euclidean space $\mathbb{R}^N$.

2.1.1 Forward Euler Method

We begin our discussion of numerical methods for initial value ordinary differential equations (ODEs) with a review of the fundamental concepts. The discretization Forward Euler [2] is used to demonstrate these principles. The problem to be solved in the general form is,

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \quad 0 \le t \le b, \tag{2.2}$$

where the initial $\mathbf{y}(0) = \mathbf{c}$ is given. To approximate the equation (2.2), the interval of integration is first discretized by a mesh

$$0 = t_0 < t_1 < \cdots < t_{N-1} < t_N = b,$$

and let $h_n = t_n - t_{n-1}$ be the nth step size. The approximations can be constructed as

$$\mathbf{y}_0(= \mathbf{c}), \mathbf{y}_1, \ldots, \mathbf{y}_{N-1}, \mathbf{y}_N,$$

with $\mathbf{y}_n$ being an intended approximation of $\mathbf{y}(t_n)$. Since $\mathbf{y}_0$ is given and proceed to integrate the Ordinary Differential Equation (ODE) in distinct steps. Each $n \in [1, N]$ is an approximation $\mathbf{y}_{n-1}$ at t_{n-1}, and the variable of interest is $\mathbf{y}_n$ at t_n.

When constructing a discretization method, consider the Taylor's expansion

$$\mathbf{y}(t_n) = \mathbf{y}(t_{n-1}) + h_n\mathbf{y}'(t_{n-1}) + \tfrac{1}{2}h_n^2\mathbf{y}''(t_{n-1}) + \dots . \tag{2.3}$$

Using the big O notation, equation (2.3) can also be written as

$$\mathbf{y}(t_n) = \mathbf{y}(t_{n-1}) + h_n\mathbf{y}'(t_{n-1}) + O(h_n^2),$$

where the rightmost term of the Taylor's expansion in equation (2.3), the forward Euler method can be derived by replacing $\mathbf{y}'$ by $\mathbf{f}$, producing the scheme, [2]

$$\mathbf{y}_n = \mathbf{y}_{n-1} + h_n\mathbf{f}\left(t_{n-1}, \mathbf{y}_{n-1}\right).$$

2.1.2 Backward Euler Method

The backward Euler method [2] is derived from equation (2.2), just like the forward Euler method. The difference between Forward- and Backward Euler method is that using the Backward Euler method, it centers at t_n, rather than at t_{n-1}, which produces

$$\mathbf{y}_n = \mathbf{y}_{n-1} + h_n\mathbf{f}(t_n, \mathbf{y}_n). \tag{2.4}$$

In terms of geometry, the backward Euler method uses the tangent at the future point $(t_n, \mathbf{y}_n)$ rather than the tangent at $(t_{n-1}, \mathbf{y}_{n-1})$ as in the forward Euler method, which improves stability. The forward Euler method is explicit, while the backward Euler method is implicit. This implies that the unknown vector $\mathbf{y}_n$ at each step appears on both sides of the equation (2.4). Hence, at each step, a nonlinear system of algebraic equation has to be solved. One advantage of the backward Euler method is the method's stability. When obtaining equation (2.4) from equation (2.2), it produces

$$y_n = y_{n-1} + hcy_n,$$

i.e.,

$$y_n = (1 - hc)^{-1}y_{n-1},$$

where c is a constant. The backward Euler method requires fewer steps than the forward Euler method to solve a problem. However, each backward Euler step can be more costly in terms of computation time. There are several applications where the implicit method's total computational cost is much lower than the explicit Euler method.

2.1.3 Runge-Kutta Method

The *Runge-Kutta method*, in numerical applications, is a widely used method for its effective use of solving problems that include differential equations. The second-order Runge-Kutta methods employs two function evaluations and yields precision proportional to the order of

h^2. Runge-Kutta methods are "one step" in the sense that the outcome at the end of one step is only functionally dependent on the result at the end of the previous step. If y_n denotes a computed approximation to $y(x_n)$, then y_n can be solved by

$$y_n = y_{n-1} + h \sum_{i=1}^{s} b_i F_i,$$

where $F_1, F_2, \ldots, F_s$ are computed derivatives from approximations $Y_1, Y_2, \ldots, Y_s$ to the solution at $x_{n-1} + hc_1, x_{n-1} + hc_2, \ldots, x_{n-1} + hc_s$. This leads to $F_i = f(x_{n-1} + hc_i, Y_i), i = 1, 2, \ldots, s$ for the differential equation system in (2.1), $F_i = f(Y_i), i = 1, 2, \ldots, s$. The values of the approximations $Y_i, i = 1, 2, \ldots, s$ can be found from

$$y_i = y_{n-1} + h \sum_{j=1}^{s} a_{ij} F_j \quad i = 1, 2, \ldots, s.$$

It turns out that the c vector components are related to the A matrix elements by

$$c_i = \sum_{j=1}^{s} a_{ij}, \quad i = 1, 2, \ldots, s.$$

The number of stages in a method of this type is the number of Y vectors needed to compute the solution, and it is a measure of the particular method complexity [12].

Higher order methods results in larger stability region. This leads to that the higher order Runge-Kutta methods have larger stability than for example the Forward Euler method. A result in substantially larger stability leads to that larger step sizes can be used. There are also methods based on Taylor's decomposition on several points, which results in higher order methods for ordinary differential equations. For further research on higher-order methods, see [3, 4, 1].

2.2 Discretization Methods for Stochastic Differential Equations

The most simplistic method for approximating the simulation of stochastic differential equations is the Euler scheme. One can begin by discussing the properties of the Euler scheme, then undertake an expansion to refine the Euler scheme, and finally present criteria for comparing methods. An extension of the Euler method for ordinary differential equation to stochastic differential equation is the Euler-Maruyama method. The Euler–Maruyama [19], is a method to approximate a numerical solution of a stochastic differential equation.

2.2.1 The Euler Scheme

The processes X is satisfying the stochastic differential equation

$$dX(t) = a\left(X(t)\right)dt + b\left(X(t)\right)dW(t). \tag{2.5}$$

Consider that X take on values in $\mathbb{R}^d$ and W is an m-dimensional standard Brownian motion, were a takes values in $\mathbb{R}^d$ and b takes values in $\mathbb{R}^{d\times m}$.

Denote $\hat{X}$ as a time-discretized approximation of X, then the Euler approximation on a time grid $0 = t_0 < t_1 < \cdots < t_m$ is defined by $\hat{X}(0) = X(0)$, for $i = 0, \ldots, m-1$, and the time-discretized approximation

$$\hat{X}(t_{i+1}) = \hat{X}(t_i) + a\left(\hat{X}(t_i)\right)[t_{i+1} - t_i] + b\left(\hat{X}(t_i)\right)\sqrt{t_{i+1} - t_i}Z_{i+1},$$

with $Z_1, Z_2, \ldots$ m-dimensional independent standard normal random vectors, and a fixed spacing h implies that $t_i = ih$. This provides that the largest of the increments $t_{i+1} - t_i$ decreases to zero. For any fixed time-step $h > 0$, $\hat{X}(ih)$ may be written as $\hat{X}(i)$ and the corresponding Euler scheme as

$$\hat{X}(i+1) = \hat{X}(i) + a(\hat{X}(i))h + b(\hat{X}(i))\sqrt{h}Z_{i+1},$$

[19].

Theorem 1 (Theorem 3.1 in [14]). *Let $\bar{X}$ and $\bar{\bar{X}}$ be forward Euler approximations of the stochastic process $X : [0,T] \times \Omega \longrightarrow \mathbb{R}$, satisfying the stochastic differential equation*

$$dX(t) = a\left(t, X(t)\right)dt + b\left(t, X(t)\right)dW(t), \quad 0 \leq t < T,$$

with time steps

$$\{\bar{t}_n\}_{n=0}^{\bar{N}}, \quad \bar{t}_0 = 0, \bar{t}_{\bar{N}} = T,$$
$$\{\bar{\bar{t}}_m\}_{m=0}^{\bar{\bar{N}}}, \quad \bar{\bar{t}}_0 = 0, \bar{\bar{t}}_{\bar{\bar{N}}} = T,$$

respectively, and

$$\Delta t_{max} = \max\left[\max_{0\leq n\leq \bar{N}-1} \bar{t}_{n+1} - \bar{t}_n, \max_{0\leq m\leq \bar{\bar{N}}-1} \bar{\bar{t}}_{m+1} - \bar{\bar{t}}_m\right].$$

Suppose that there exists a positive constant C such that the initial data and the given functions $a, b : [0,T] \times \mathbb{R} \longrightarrow \mathbb{R}$ satisfy

$$E[|\bar{X}(0)|^2 + |\bar{\bar{X}}(0)|^2] \leq C,$$

$$E[\left(\bar{X}(0) - \bar{\bar{X}}(0)\right)^2] \leq C\Delta t_{\max},$$

and

$$|a(t,x) - a(t,y)| < C|x-y|,$$
$$|b(t,x) - b(t,y)| < C|x-y|,$$
$$|a(t,x) - a(s,x)| + |b(t,x) - b(s,x)| \leq C(1+|x|)\sqrt{|t-s|}.$$

Then there is a constant K such that

$$\max\left\{E[\bar{X}^2(t,\cdot)], E[\bar{\bar{X}}^2(t,\cdot)]\right\} \leq K(T+1), \quad t < T,$$

and

$$E[\left(\bar{X}(t,\cdot) - \bar{\bar{X}}(t,\cdot)\right)^2] \leq K\Delta t_{\max}, \quad t < T.$$

2.2.2 Euler–Maruyama Scheme

For the scalar Stochastic Differential Equation (SDE)

$$dS_t = a(S_t, t)dt + b(S_t)dW_t, \tag{2.6}$$

divide a time-discretization of the interval $[0, T]$ into time-steps $0 = t_0 < t_1 < \cdots < t_N = T$. The *Euler-Maruyama scheme* is the most intuitive way to discretize the scalar SDE in (2.6)

$$\begin{aligned} \hat{X}(t_{i+1}) &= \hat{X}(t_i) + a(\hat{X}(t_i))(t_{i+1} - t_i) + b(\hat{X}(t_i))\sqrt{t_{i+1} - t_i}\, Z_{i+1}, \\ \hat{X}(t_0) &= X(0), \end{aligned} \tag{2.7}$$

where $Z_i, \ldots, Z_n$ are standard normal random variables that are independent and identically distributed. Let $t_i := ih$, $h > 0$, and $\hat{X}(i) := \hat{X}(ih)$. Then,

$$\hat{X}(i+1) = \hat{X}(i) + a(\hat{X}(i))h + b(\hat{X}(i))\sqrt{h}\, Z_{i+1}.$$

Definition 1. A discretization $\hat{X}$ has *strong order of convergence* $\beta > 0$ if there exist positive constants c and h_0 such that for all h $\in (0, h_0)$ we have

$$E[|\hat{X}(nh) - X(T)|] \leq ch^{\beta}.$$

Definition 2. A function $g : \mathbb{R} \longrightarrow \mathbb{R}$ is called *polynomially bounded* if there exist positive constants k and q such that for all $x \in \mathbb{R}$ we have

$$|g(x)| \leq k(1 + |x|^q).$$

Consider $C_P^{2\beta+2}$ as the set of functions whose derivatives of orders up to $2\beta + 2$ are polynomially bounded.

Definition 3. A discretization $\hat{X}$ has *weak order of convergence* $\beta > 0$ if there exist positive constant h_0 and for any $f \in C_P^{2\beta+2}$ there exists a constant $c = c(f) > 0$ such that for all $h \in (0, h_0)$ we have

$$|E[f(\hat{X}(nh))] - E[f(X(T))]| \leq ch^{\beta}.$$

The approximation of the SDE in equation (2.7) can in certain instances be used to achieved a more refined discretization, called the Milstein scheme

$$\hat{X}(i+1) \approx \hat{X}(i) + a(\hat{X}(i))h + b(X(i))\sqrt{h}\, Z_{i+1} + \frac{1}{2}b'(\hat{X}(i))b(\hat{X}(i))h(Z_{i+1}^2 - 1).$$

The Milstein scheme has a weak convergence with order of 1 and a strong convergence of order 0.5 [19].

2.3 Generating Sample Paths

The most well-known method to simulate paths for a variety of stochastic process is the *Brownian motion*. We dedicate this section to methods of precisely simulating continuous time processes over a discrete set of dates. Additionally, we present the simulation of Brownian motion paths of one dimension and the Geometric Brownian motion.

2.3.1 Standard Brownian Motion

One of the most frequently asked questions is whether the random process reaches a limit when the phase size is finer and finer. We now move on to continuous random processes to see if the model makes sense to the point that the step size reaches zero when we use lattices to model asset prices. The models are based on the SDE:

$$dS = a(t, S)dt + b(t, S)dW$$

for the asset price $S(t)$, where the $a(t, S)dt$ term accounts for *deterministic motions*, and the other term $b(t, S)dW$ accounts for *random motions*. To develop such stochastic models, the first step is to define the Brownian motion $W(t)$. Randomly generated piece-wise process $W(t)$ is composed of pieces with identical statistical properties, no matter how it is sub-divided. This is because stock prices appear random on time scales. The following are the characteristics of the Brownian Motion:

(i) $W(t)$ have independent increments. For any date τ and for any $\Delta\tau > 0$, the value of $\Delta W = W(\tau + \Delta\tau) - W(\tau)$ is independent of $W(t)$ for all $t \leq \tau$. This means that the increments of Brownian motion are independent of everything that happened prior to or at the current date τ.

(ii) Increments $\Delta W \equiv W(t_2) - W(t_1)$ is Gaussian random variables with mean 0 and variance $\Delta t \equiv t_2 - t_1$. Consequently,

$$\Delta W \equiv W(t_2) - W(t_1) = \sqrt{t_2 - t_1}\xi,$$

where $\xi \sim \mathbb{N}(0, 1)$ is a Gaussian random variable with mean zero and variance 1. The aim is to represent ΔW with zero mean, as it follows the random part of the asset price movements. Any non-zero term would represent a deterministic piece, hence that could be put in the drift term $a(t, S)dt$.

(iii) It is easily proven that for any $\delta > 0$, $W(t)$ is a continuous random process by,

$$\text{prob}\{|W(t + \Delta t) - W(t)| > \delta\} = \text{prob}\left\{|\xi| > \frac{\delta}{\sqrt{\Delta t}}\right\} \xrightarrow[\Delta t \to 0]{} 0.$$

The above equation is the definition of a continuous stochastic process.

(iv) $W(t)$ is almost surely nowhere differentiable.

For any $K > 0$, we argue that

$$\text{prob}\left\{\left|\frac{W(t - \Delta t) - W(t)}{\Delta t}\right| < K\right\} = \text{prob}\left\{|\xi| > K\sqrt{\Delta t}\right\} \xrightarrow[\Delta t \to 0]{} 0.$$

Hence, the probability that the slope is bounded equals zero as $\Delta t \to 0$. The above property demonstrates that $W(t)$ is almost certainly not differentiable anywhere.

(v) The continuity and non-differentiability followed directly from the scaling of ΔW. Since $\Delta W = \sqrt{\Delta t \xi}$, where ξ is $\mathbb{N}(0, 1)$, and can be written as $\Delta W \sim O(\sqrt{\Delta t})$, or more concisely $dW \sim O(\sqrt{\Delta t})$.

To show that ΔW is Gaussian with variance Δt, which follows directly from our desire to have $W(t)$ to be sub-dividable into finer and finer intervals, each with identical properties. Consider

$$\Delta W = W(t_1) - W(t_0) \equiv \sum_{k=1}^{n-1} [W(\tau_{k+1}) - W(\tau_k)], \quad \tau_k = t_0 + \frac{k}{n}(t_1 - t_0),$$

where each $\delta W_k \equiv W(\tau_{k+1}) - W(\tau_k)$ are independent random variables with identical distributions. Since the variables are independent, the variances are

$$\mathrm{Var}[\Delta W] = \sum_{k=0}^{n-1} \mathrm{Var}[W(\tau_{k+1}) - W(\tau_k)] = n \cdot \mathrm{Var}[W(t_1) - W(t_0)].$$

Let $v(y) = \mathrm{Var}[W(t + y) - W(t)]$. Thus,

$$v(t_1 - t_0) = n \cdot v\left(\frac{t_1 - t_0}{n}\right)$$

for any t_1, t_0 and n. Therefore, ΔW is the sum of n independent, identically distributed variables with mean 0 and variance $(t_1 - t_0)/n$. As n approaches ∞, the central limit theorem guarantees that δW is Gaussian with mean zero and variance $t_1 - t_0$ [19, 28].

Definition 4. (The Wiener process). [Definition 2.8 in [14]] The one-dimensional *Wiener process* $W : [0, \infty) \times \Omega \to \mathbb{R}$, also known as the Brownian motion, has the following properties:

1. with probability 1, the mapping $t \mapsto W(t)$ is continuous and $W(0) = 0$;
2. if $0 = t_0 < t_1 < \cdots < t_N = T$, then the increments

   $$W(t_N) - W(t_{N-1}), \ldots, W(t_1) - W(t_0)$$

   are independent; and
3. for all $t > s$ the increment $W(t) - W(s)$ is normally distributed, with $E[W(t) - W(s)] = 0$ and $E[(W(t) - W(s))^2] = \mathrm{Var}[W(t) - W(s)] = t - s$.

To generate Brownian motion, determine W at finitely many time steps $\{t_n : n = 0, \ldots, N\}$ of the form $t_0 < t_1 < \cdots < t_N = T$. Definition 4 shows how to generate $W(t_n)$ by a sum of independent normally distributed random variables for computational method to generate independent normal distributed random variables. Denote these increments as $\Delta W_n = W(t_{n+1}) - W(t_n)$. By Definition 4, the Brownian motion $W(t)$ is itself a normally distributed random variable for fixed time t. To generate W for all $t \in \mathbb{R}$ is computationally infeasible, since it requires infinite computational work [14].

2.3.2 Geometric Brownian Motion

A geometric Brownian motion is an exponentiation of the Brownian motion. This means that a stochastic process $S(t)$ is a geometric Brownian motion if the term $\log S(t)$ is a Brownian motion with an initial value $\log S(0)$. Since the exponential function only takes positive values, geometric Brownian motion is always positive, compared to ordinary Brownian motion that can take on non-positive values. The percentage changes for geometric Brownian motion is

$$\frac{S(t_2)-S(t_1)}{S(t_1)}, \frac{S(t_3)-S(t_2)}{S(t_2)}, \dots, \frac{S(t_n)-S(t_{n-1})}{S(t_{n-1})},$$

where the changes are independent for any time-step $t_1 < t_2 < \cdots < t_n$, rather than the absolute changes $S(t_{i+1}) - S(t_i)$. These characteristics explain why geometric Brownian motion, rather than ordinary Brownian motion, is used to model asset prices.

Suppose W is a standard Brownian motion and X satisfies

$$dX(t) = \mu dt + \sigma dW(t),$$

so that $X \sim \mathrm{BM}(\mu, \sigma^2)$. Set $S(t) = S(0)\exp(X(t)) \equiv f(X(t))$, with an application of the Itô formula it shows that

$$\begin{aligned} dS(t) &= f'(X(t))dX(t) + \tfrac{1}{2}\sigma^2 f''(X(t))dt \\ &= S(0)\exp(X(t))\,[\mu dt + \sigma dW(t)] + \tfrac{1}{2}\sigma^2 S(0)\exp{+}(X(t))dt \\ &= S(t)(\mu + \tfrac{1}{2}\sigma^2)dt + S(t)\sigma dW(t). \end{aligned} \tag{2.8}$$

A stochastic differential equation of the form

$$\frac{dS(t)}{S(t)} = \mu dt + \sigma dW(t), \tag{2.9}$$

is often used to describe a geometric Brownian motion. In equation (2.9), $S(t)$ has the drift $\mu S(t)$, which implies the following equation that can be verified through Itô formula:

$$d\log S(t) = (\mu - \tfrac{1}{2}\sigma^2)dt + \sigma dW(t). \tag{2.10}$$

To show that S is a process of the type in equation (2.9), the notation $S \sim \mathrm{GBM}(\mu, \sigma^2)$ will be used. If S has the initial value $S(0)$, and from equation (2.10) one can see that $S \sim \mathrm{GBM}(\mu, \sigma^2)$ then

$$S(t) = S(0)\exp\left(\left[\mu - \tfrac{1}{2}\sigma^2\right]t + \sigma W(t)\right).$$

If $u < t$ then

$$S(t) = S(u)\exp\left(\left[\mu - \tfrac{1}{2}\sigma^2\right](t-u) + \sigma(W(t) - W(u))\right),$$

since the increments of W are independent and normally distributed, a simple recursive method for simulating S values at $0 = t_0 < t_1 < \cdots < t_n$:

$$S(t_{i+1}) = S(t_i)\exp\left(\left[\mu - \tfrac{1}{2}\sigma^2\right](t_{i+1} - t_i) + \sigma\sqrt{t_{i+1} - t_i}Z_{i+1}\right),$$

where $i = 0, 1, \dots, n-1$ and with $Z_1, Z_2, \dots, Z_n$ being standard normal random variables [19].

2.3.3 Random Walk Construction

When simulating the Brownian motion (see Fig. 2.1), one primarily concentrates on simulating values $(W(t_1), \ldots, W(t_n))$ at a fixed set of points $0 < t_1 < \cdots < t_n$. Simulating the $W(t_i)$ is straightforward since Brownian motion has independent normally distributed increments. Let $Z_1, \ldots, Z_n$ be independent standard normal random variable, for a Brownian motion with $t_0 = 0$ and $W(0) = 0$. As a result, the following values can be generated

$$W(t_{i+1}) = W(t_i) + \sqrt{t_{i+1} - t_i}\, Z_{i+1}, \quad i = 0, \ldots, n-1.$$

When $X \sim BM(\mu, \sigma^2)$ with given $X(0)$ and constant μ and σ, set

$$X(t_{i+1}) = X(t_i) + \mu(t_{i+1} - t_i) + \sigma\sqrt{t_{i+1} - t_i}\, Z_{i+1}, \quad i = 0, \ldots, n-1.$$

Let the coefficients be time dependent. Hence, the recursion becomes

$$X(t_{i+1}) = X(t_i) + \int_{t_i}^{t_{i+1}} \mu(s) ds + \sqrt{\int_{t_i}^{t_{i+1}} \sigma^2(u) du}\, Z_{i+1}, \quad i = 0, \ldots, n-1. \tag{2.11}$$

Replacing equation (2.11) with the Euler approximation one gets

$$X(t_{i+1}) = X(t_i) + \mu(t_i)(t_{i+1} - t_i) + \sigma(t_i)\sqrt{t_{i+1} - t_i}\, Z_{i+1}, \quad i = 0, \ldots, n-1.$$

Since the increments $t_1, \ldots, t_n$ mean and variance are no longer exact it will have a discretization error even at the increments [19].

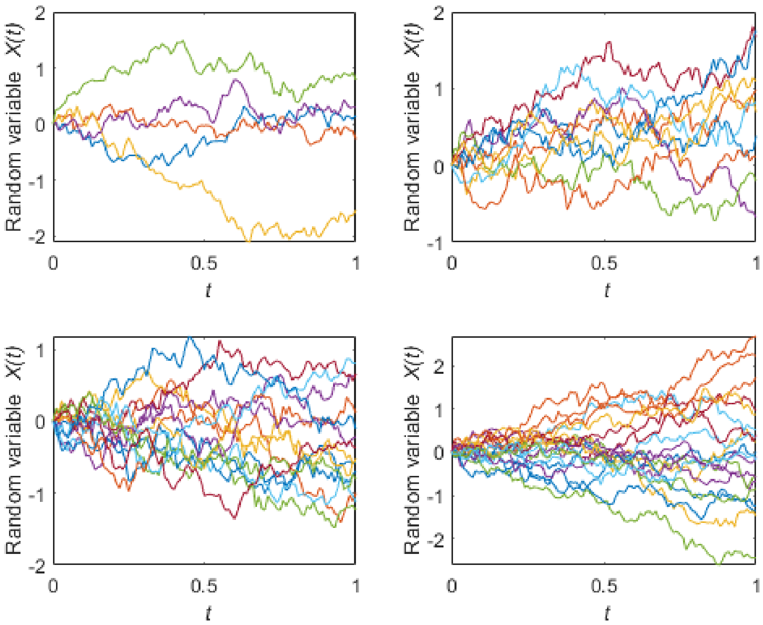


Figure 2.1: Simulated standard Brownian motion paths

2.4 The ITÔ Lemma

The *Itô Lemma* is one of the most important formulas in financial analysis and it explains how to distinguish stochastic process functions. Begin with a Taylor expansion to the lowest orders for a function of two variables $F(t, X)$ to understand the Ito formula in its most simple form

$$dF = \frac{\partial F}{\partial t}dt + \frac{\partial F}{\partial X}dX + \frac{1}{2}\frac{\partial^2 F}{\partial t^2}(dt)^2\frac{1}{2}\frac{\partial^2 F}{\partial X^2}(dX)^2 + \frac{\partial^2 F}{\partial t \partial X}dtdX + \ldots,$$

for the stochastic process X

$$dX = \mu dt + \sigma dW,$$

where μ is the deterministic drift and σ the volatility, and W is a Wiener process, also called a Brownian motion, with the property $dW^2 = dt$. For the lowest order it is simplified to

$$(dX)^2 = \mu^2 dt^2 + \sigma^2 dW^2 + 2\mu\sigma dtdW \to \sigma^2 dt,$$

$dtdW$ and $dtdt$ can be ignored in the lowest order. Hence, to the lowest order of dF:

$$dF = \left(\frac{\partial F}{\partial t} + \mu\frac{\partial F}{\partial X} + \frac{1}{2}\sigma^2\frac{\partial^2 F}{\partial X^2}\right)dt + \sigma\frac{\partial F}{\partial X}dW, \tag{2.12}$$

and, (2.12) is called the Ito's formula [28].

Theorem 2. *(Theorem 3.9 in [14]). Suppose that the assumption in Theorem* (1) *holds, and that X satisfies the stochastic differential equation*

$$\begin{aligned} &dX(s) = a\left(s, X(s)\right)ds + b\left(s, X(s)\right)dW(s), \quad s > 0 \\ &X(0) = X_0, \end{aligned}$$

and let $g : (0, +\infty) \times \mathbb{R} \to \mathbb{R}$ *be a given bounded function in* $C^2((0, \infty) \times \mathbb{R})$. *Then* $y(t) \equiv g(t, X(t))$ *satisfies the stochastic differential equation*

$$\begin{aligned} dy(t) = &\left(\partial_t g(t, X(t)) + a(t, X(t))\partial_x g(t, X(t)) + \frac{b^2(t, X(t))}{2}\partial_{xx} g(t, X(t))\right)dt \\ &+ b(t, X(t))\partial_x g(t, X(t))dW(t). \end{aligned}$$

Chapter 3

Standard Monte Carlo Method

Monte Carlo (MC) simulations are named after Monaco's well-known gambling destination, and it was first developed by Stanislaw Ulam. Ulam was a mathematician who worked on the Manhattan Project in 1939. In collaboration with John von Neumann they developed the Monte Carlo simulation in 1946 [23]. A Monte Carlo simulation works by constantly repeating random samples to receive a result, where it takes the uncertain variable and assigns it a random value, and it repeats this process while assigning the variable in question to different values. When the simulation is complete, the average results provide an estimate. Consider the integral

$$\alpha = \int_0^1 f(x)dx,$$

if U is a random variable that is uniformly distributed on $[0, 1]$. Then independent points can be drawn of U to evaluate the function f at $U_1, U_2, ..., U_n$, and take the mean of those values to get a Monte Carlo estimate

$$\hat{\alpha}_n = \frac{1}{n}\sum_{i=1}^{n} f(U_i).$$

If $\int_0^1 |f(x)|dx < \infty$, then by the strong law of large numbers,

$$P\{\lim_{n\to\infty} \hat{\alpha}_n = \alpha\} = 1.$$

Assume that $\int_0^1 [f(x)]^2 dx < \infty$, where,

$$\sigma_f^2 = \int_0^1 [f(x) - \alpha]^2 \, dx,$$

then by the Central Limit Theorem [14] (see Theorem 3), the distribution of the error $\alpha_n - \alpha$ is approximately normally distributed with mean 0 and standard deviation $\sigma_f/\sqrt{n}$, and the accuracy of this approximation is improving with an increase of n. The unknown parameter σ_f can be estimated using the sample standard deviation

$$s_f = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(f(U_i) - \hat{\alpha}_n)^2}.$$

In particular, from the values of the function $f(U_1), f(U_2), \ldots, f(U_n)$ an estimate is obtained of the integral α and a measure of the error. The standard error is proportional to $1/\sqrt{n}$, and decreasing it twice requires increase of n by a factor of 4 [19].

Monte Carlo simulation is used in finance to value and analyse financial instruments, portfolios, investments, and the method can evaluate and price options [18]. It works by simulating the sources of uncertainty that affect their value, The price of the underlying asset is first simulated by random generation for several paths. The value of the option can be approximated by calculating the average of discounted returns over all paths. The option is priced under risk-neutral measure, since the discount rate is equal to the risk-free interest rate. The variance of the estimator should approach zero, in order to get a good estimate from this simulation [30].

In Monte Carlo path simulation, one is interested in the expected value of a quantity, and it is a functional of the solution to a stochastic differential equation. Consider the SDE

$$dS(t) = a(S,t)dt + b(S,t)dW(t), \quad 0 < t < T, \tag{3.1}$$

given the initial data S_0 to compute the expected value of $f(S(T))$, where $f(S)$ is a scalar function with a uniform Lipschitz bound. Hence, there exists a constant c such that

$$|f(U) - f(V)| \leq c \parallel U - V \parallel, \quad \forall U, V. \tag{3.2}$$

Euler discretization of this SDE with time-step h is

$$\hat{S}_{n+1} = \hat{S}_n + a(\hat{S}_n, t_n)h + b(\hat{S}_n, t_n)\Delta W_n.$$

The estimate for $E[f(S_T)]$ is the mean of the payoff values $f(\hat{S}_{T/h})$ from N independent path simulations,

$$\hat{Y} = N^{-1} \sum_{i=1}^{N} f\left(\hat{S}_{T/h}^{(i)}\right).$$

Provided that $a(S,t)$ and $b(S,t)$ satisfy certain conditions, the expected mean-square-error (MSE) in the estimate $\hat{Y}$ is asymptotic to the form

$$MSE \approx c_1 N^{-1} + c_2 h^2, \tag{3.3}$$

where c_1 and c_2 are positive constants. In Monte Carlo sampling, the first term in (3.3) corresponds to the variance in $\hat{Y}$, and the second term introduced by the Euler discretization is the square of the $O(h)$ bias [18].

Consider the SDE

$$dX(t) = a\left(t, X(t)\right) dt + b\left(t, X(t)\right) dW(t)$$

on the interval $t_0 \leq t \leq T$. To compute the value $E[g(X(T))]$, Monte Carlo method can be used based on the following approximation,

$$E[g(X(T))] \simeq \sum_{j=1}^{N} \frac{g(\bar{X}(T;\omega_j))}{N},$$

where $\bar{X}$ is an approximation of X. The error in the Monte Carlo method is given by

$$\begin{aligned} & E[g(X(T))] - \sum_{j=1}^{N} \frac{g(\bar{X}(T;\omega_j))}{N} \\ =& E[g(X(T)) - g(\bar{X}(T))] - \sum_{j=1}^{N} \frac{g(\bar{X}(T;\omega_j)) - E[g(\bar{X}(T))]}{N}. \end{aligned} \tag{3.4}$$

The first term of the error representation in (3.4) is the time discretization error and the second term is the statistical error [14]. To understand the statistical error of Monte Carlo methods, The *Central Limit Theorem* is the fundamental result.

Theorem 3. *(The Central Limit Theorem).[Theorem 5.2 in [14]] Assume ξ_n, $n = 1, 2, 3, \ldots,$ are independent, identically distributed and $E[\xi_n] = 0, E[\xi_n^2] = 1$. Then*

$$\sum_{n=1}^{N} \frac{\xi_n}{\sqrt{N}} \rightharpoonup \nu, \tag{3.5}$$

where $\nu \sim \mathbb{N}(0, 1)$ and $\rightharpoonup$ denotes convergence of the distribution. The convergence of (3.5) *means $E[g(\sum_{n=1}^{N} \xi_n/\sqrt{N})] \to E[g(\nu)]$ for all bounded and continuous functions g.*

Theorem 4 (Theorem 5.8 in [14])**.** *Assume that a, b and g are differentiable to any order and these derivatives are bounded, then there holds*

$$E[g(X(T)) - g(\bar{X}(T))] = O(\max \Delta t).$$

3.1 Error Estimate for the Monte Carlo Method

Let y be the output and N be the sample size for the Monte Carlo simulation. Let the individual values from each trail be labelled y_i where $i = 1, \ldots, N$. Consider the error made when estimating the expected value of y when using a sample that has N trials. If the probability density function of y is $f(y)$, then the expected value of y is,

$$\mu_y = E[y] = \int_{-\infty}^{\infty} y f(y) dy.$$

For N trials, the sample mean, would be a reasonable estimator for μ_y,

$$\bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i.$$

Monte Carlo simulation involves pseudo-random draws of the inputs. Hence, one will obtain different results each time the simulation is performed, and slightly different results for $\bar{y}$. The variability in the results of $\bar{y}$ depends on N, the number of trials in each Monte Carlo simulation, also known as the moments of the error. The first question to ask is how accurate is $\bar{y}$ as an estimate of μ_y. To answer this, calculate the expectation of $\bar{y} - \mu_y$,

$$\begin{aligned} E[\bar{y} - \mu_y] &= E[\bar{y}] - \mu_y \\ &= E[\frac{1}{N}\sum_{i=1}^{N} y_i] - \mu_y \\ &= \frac{1}{N}\sum_{i=1}^{N} E[y_i] - \mu_y. \end{aligned}$$

In the Monte Carlo method, the y_i term occur from a random sampling of the inputs, $E[y_i] = \mu_y$. Thus,

$$E[\bar{y} - \mu_y] = \frac{1}{N}N\mu_y - \mu_y = 0.$$

This results shows that on average, the error in using $\bar{y}$ to approximate μ_y is zero, also known as an unbiased estimator. To quantify the variability in $\bar{y}$, use the variance of $\bar{y} - \mu_y$. Since the term μ_y is a constant, the variance is zero. Therefore,

$$\begin{aligned} &\mathrm{Var}[\bar{y} - \mu_y] = \mathrm{Var}[\bar{y}] - \mathrm{Var}[\mu_y] \\ &= \mathrm{Var}[\bar{y}] \\ &= \mathrm{Var}\left[\frac{\sum_{i=1}^{N} y_i}{N}\right] \\ &= \frac{1}{N^2}\mathrm{Var}\left[\sum_{i=1}^{N} y_i\right]. \end{aligned}$$

The Monte Carlo method draws independent random samples; Therefore, the variance of the sum from samples y_i is the sum of their variance. Hence,

$$\begin{aligned} \mathrm{Var}[\bar{y} - \mu_y] &= \frac{1}{N^2}\sum_{i=1}^{N} \mathrm{Var}[y_i] \\ &= \frac{1}{N^2}N\sigma_y^2 = \frac{\sigma_y^2}{N}. \end{aligned}$$

Therefore, the following equations holds:

$$\mu_y \equiv E[\bar{y}] = \mu_y, \quad \sigma_{\bar{y}}^2 \equiv E[(\bar{y} - \mu_y)^2] = \frac{\sigma_y^2}{N}.$$

The quantity $\sigma_{\bar{y}}$ is known as the standard error of the estimator, and as the square root of the sample size $\sqrt{N}$ increases, the standard error decreases. To reduce the uncertainty in the mean estimate by a factor of ten, the number of Monte Carlo trials must be increased by a factor of 100.

To approximate the distribution of $\bar{y}$, the central limit theorem can be applied for large sample size N. The central limit theorem says for large N, the distribution of $\bar{y} - \mu_y$ will approach a normal distribution with mean 0 and variance $\frac{\sigma_y^2}{N}$,

$$f(\bar{y} - \mu_y) \to \mathcal{N}(0, \sigma_{\bar{y}}) = \mathcal{N}\left(0, \frac{\sigma_y^2}{N}\right).$$

In a practical situation one cannot calculate the above error estimate of the confidence intervals because they depend on σ_y, an unknown value. Therefore, an unbiased estimate of σ_y^2 is utilized

$$s_y^2 \equiv \frac{1}{N-1} \sum_{i=1}^{N} (y_i - \bar{y})^2.$$

This introduces additional uncertainty in the quality of the estimate, and for small sample sizes it could be significant [14, 17].

3.2 Variance Reduction

Monte Carlo methods are simulation algorithms that are used to approximate a numerical quantity in real-world mathematical applications, often programmed in languages such as Matlab, R, and Python. Although computer speeds and computational times has increased dramatically over the years, one still needs *variance reduction* techniques for increasing the efficiency of Monte Carlo simulation by reducing the variance of simulation estimates. Simple variance reduction techniques are often surprisingly successful and simple to implement. It provides possibilities to what would otherwise be difficult in certain applications, such as rare event simulation and quantum chemistry. Different variance reduction techniques are used in the most advanced Monte Carlo simulations, such as monitor variates, partial integration, systematic sampling, and significance sampling. If we can simplify a version of the problem, it can sometimes be solved directly with control variates approach. This is frequently the case in simple problems, such as quantitative finance pricing problems.

The error variance of Monte Carlo integration is usually in the form σ^2/n, where n denotes the number of samples. By increasing the value of n, the approximation of the result is better, but the computational time increases. Instead, one will often find a way to minimize the variance σ^2. To accomplish this, create a new Monte Carlo problem with the same solution as our previous one but a lower variance.

The *control variates method* is one of the most efficient and widely used techniques for increasing the Monte Carlo simulations performance, by reducing the variance. It uses knowledge about the errors in known-quantity estimates to reduce the error in an unknown-quantity estimate. Let $Y_1, \ldots, Y_n$ be the outputs from n replications of a simulation. Suppose that Y_i, $i = 1, \ldots, n$ are independent and identically distributed, where the aim is to estimate $E[Y_i]$. The sample mean $\bar{Y} = (Y_1 + \cdots + T_n)/n$ is the most common estimator for practical use. The estimator is unbiased and converges as $n \to \infty$ with a probability of 1. Suppose that another output X_i along with Y_i could be calculated on each replication. Assume that the pairs

(X_i, Y_i), $i = 1, \dots, n$ are independent and identically distributed the expected value, $E[X]$ of the output X_i are known. For any fixed b

$$Y_i(b) = Y_i - b(X_i - E[X]),$$

would be calculated using the ith replication and thereafter compute the sample mean

$$\bar{Y}(b) = \bar{Y} - b(\bar{X} - E[X]) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - b(X_i - E[X])). \tag{3.6}$$

The observed error $\bar{X} - E[X]$ is provided as a control in estimating $E[Y]$, and known as the control variate estimator [24].

Lemma 1. *The control variate estimator* (3.6)*, is unbiased and consistent, [19].*

Proof.

$$E[\bar{Y}(b)] = E[\bar{Y} - b(\bar{X} - E[X])] = E[\bar{Y}] = E[Y]$$

This is consistent with probability of 1 because

$$\begin{aligned}\lim_{n\to\infty}\sum_{i=1}^{n} Y_i(b) &= \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}(Y_i - b(X_i - E[X])) \\ &= E[Y - b(X - E[X])] \\ &= E[Y].\end{aligned}$$

The variance or each $Y_i(b)$ is

$$\begin{aligned}\mathrm{Var}[Y_i(b)] &= \mathrm{Var}[Y_i - b(X_i - E[X])] \\ &= \sigma_Y^2 - 2b\sigma_X\sigma_Y\rho_{XY} + b^2\sigma_X^2 \equiv \sigma^2(b),\end{aligned}$$

where $\sigma_X^2 = \mathrm{Var}[X], \sigma_Y^2 = \mathrm{Var}[Y]$ and ρ_{XY} is the correlation coefficient between X and Y.

Since the control variate estimator $\bar{Y}(b)$ has a variance of $\sigma^2(b)/n$ and the sample mean $\bar{Y}$ has a variance of σ_Y^2/n. This implies that the control variate estimator has a smaller variance than the standard estimator as long as the constraint $b^2\sigma_X < 2b\sigma_Y\rho_{XY}$ holds. To minimize the variance (3.2) the optimal coefficient b^* can be used,

$$b^* = \frac{\sigma_Y}{\sigma_X}\rho_{XY} = \frac{\mathrm{Cov}[X,Y]}{\mathrm{Var}[X]}.$$

Substitute the value of b^* from (3.2) into the equation (3.6) and simplify, which produce the ratio of the variance from the optimally controlled estimator to that of the uncontrolled estimator

$$\frac{\mathrm{Var}[\bar{Y} - b^*(\bar{X} - E[X])]}{\mathrm{Var}[\bar{Y}]} = 1 - \rho_{XY}^2.$$

Several potential control variates can often be combined into a single control variate, although this is not always possible. Since there are several different control variates, the issue of multiple controls is intriguing. The previous findings can easily be generalized to several control variables. Let $\bar{\mathbf{X}}$ denote the vector of sample means of the controls. For a fixed $b \in \mathbb{R}^d$ the control variate estimator $\bar{Y}(b)$ is

$$\bar{Y}(b) = \bar{Y} - b^\top(\bar{\mathbf{X}} - E[X]),$$

is an unbiased and consistent estimator of $E[Y]$; see Lemma (1). The optimal parameter values in the vector $\mathbf{b}$ are given by

$$\mathbf{b}^* = \Sigma_X^{-1}\Sigma_{XY},$$

where $\mathbf{\Sigma}_X^{-1}$ and $\mathbf{\Sigma}_{XY}$ corresponds to the covariance matrix of X and the vector of covariances between (Y, X). The estimation of $\mathbf{b}^*$ is given by

$$\hat{\mathbf{b}}_n^* = \mathbf{S}_X^{-1}\mathbf{S}_{XY},$$

where S_X is the $d \times d$ matrix with j, k entry and S_{XY} is the d-vector with jth entry.

$$S_{X^{(j)}X^{(k)}} = \frac{1}{n-1}\left(\sum_{i=1}^{n} X_i^{(j)}\bar{X}_i^{(k)} - n\bar{X}^{(j)}\bar{X}^{(k)}\right), \quad j, k = 1, \dots, d,$$

$$S_{XY^{(j)}} = \frac{1}{n-1}\left(\sum_{i=1}^{n} X_i^{(j)}Y_i - n\bar{X}^{(j)}\bar{Y}\right), \quad j = 1, \dots, d,$$

are the sample counterparts to Σ_X^{-1} and Σ_{XY} [27]. See [19] for further reading.

Chapter 4

Multilevel Monte Carlo for pricing American options

As seen in previous parts of this study, standard Monte Carlo provides a reasonably straightforward method for computing random variable expectations. In the context of financial analysis, this entails combining traditional Monte Carlo with a numerical method for SDEs to calculate the estimated value of the payout that corresponds to the option value. The only disadvantage is that standard Monte Carlo has a slow convergence rate, which results in the high computational cost. As Giles shows in [18] it is possible to build on this convergence rate and significantly reduce simulation costs. This method, or approach to simulation, is known as *Multilevel Monte Carlo* (MLMC). In the following sections the most basic version of MLMC will be explained and using that knowledge we price the American option.

4.1 Black-Scholes Model

In the year 1973, Myron Scholes and Fisher Black published their paper *The Pricing of Options and Corporate Liabilities* [8]. Robert C. Merton later expanded the mathematical understanding of the option pricing model, later named *Black-Scholes option pricing model*. The Black-Scholes partial differential equation is a Nobel Prize winning result in economics (1997) and one of the most famous models in finance.

Let $f(t, S(t))$ be the price of an European put option where $S(t)$ is the price of a stock. The price of a stock satisfies the stochastic differential equation $dS = \mu S dt + \sigma S dW$, where the drift μ and the volatility σ are constants. Assume that there exist a risk-free paper, B, which follows $dB = rBdt$, where r is the constant risk-free rent. First, consider the portfolio $I = -f + \alpha S + \beta B$ for $\alpha(t), \beta(t) \in \mathbb{R}$ to find the partial differential equation of the price, and $f(t, S(t))$ of an option. Then by Itô's formula

$$\begin{aligned} dI &= -df + \alpha dS + \beta dB \\ &= -(f_t + \mu S f_S + \tfrac{1}{2}\sigma^2 S^2 f_{SS})dt - f_{S\sigma} SdW + \alpha(\mu S dt + \sigma S dW) + \beta r B dt \\ &= \Big(-(f_t + \mu S f_S + \tfrac{1}{2}\sigma^2 S^2 f_{SS}) + (\alpha \mu S + \beta r B)\Big)\, dt + (-f_S + \alpha)\sigma S dW. \end{aligned}$$

Now choose α such that the portfolio I becomes risk-less. Hence, $\alpha = f_S$,

$$\begin{aligned} dI &= \left(-(f_t + \mu S f_S + \tfrac{1}{2}\sigma^2 S^2 f_{SS}) + (f_S \mu S + \beta r B)\right) dt \\ &= \left(-(f_t + \tfrac{1}{2}\sigma^2 S^2 f_{SS}) + \beta r B\right) dt. \end{aligned} \tag{4.1}$$

Assume the existence of an arbitrage opportunity is disallowed. Implying that $dI = rIdt$, where r is the interest rate for risk-less investment,

$$\begin{aligned} dI &= r(-f + \alpha S + \beta B)dt \\ &= r(-f + f_S S + \beta B)dt. \end{aligned} \tag{4.2}$$

Equations (4.1) and (4.2) show that

$$f_t + rsf_s + \tfrac{1}{2}\sigma^2 s^2 f_{ss} = rf, \quad t < T,$$

at maturity time T the contract value is given by the definition,

$$f(T, s) = \max(K - s, 0).$$

Equation (4.1) is a deterministic partial differential equation and is called the Black-Scholes equation.

Theorem 5. *(Theorem 4.1 in [14]). Suppose that a, b, and g are differentiable to any order and these derivatives are bounded. Let X be the solution of the stochastic differential equation,*

$$dX(t) = a\,(t, X(t))\,dt + b\,(t, X(t))\,dW(t),$$

and let $u(x, t) = E[g(X(T))|X(t) = x]$. Then u is the solution of the Komogorov backward equation

$$\begin{aligned} &L^* u \equiv u_t + au_x + \tfrac{1}{2}b^2 u_{xx} = 0, \quad t < T \\ &u(x, T) = g(x). \end{aligned}$$

Theorem 6. *(Feynman-Kač)(Theorem 4.7 in [14]). Suppose that a, b, g, h and V are bounded smooth functions. Let X be the solution of the stochastic differential equation $dX(t) = a(t, X(t))dt + b(t, X(t))dW(t)$ and let*

$$\begin{aligned} u(x, t) =& E[g(X(T)) \exp(\int_t^T V(s, X(s))ds)|X(t) = x] \\ &+ E[-\int_t^T h(s, X(s))e^{\int_t^s V(\tau, X(\tau))d\tau}, ds|X(t) = x]. \end{aligned}$$

Then u is the solution of the partial differential equation

$$L_V^* u \equiv u_t + au_x + \tfrac{1}{2}b^2 u_{xx} + V_u = h, \quad t < T \tag{4.3}$$

$$u(x, T) = g(x).$$

Comparing equation (4.1) with equation (4.3), one can reasonably conclude that u corresponds to f, X to $\tilde{S}$, $a(t,x) = rx$, $b(t,x) = \sigma x$, $V = -r$ and $h = 0$. Using Feynman-Kač (Theorem 6),

$$f(t, \tilde{S}(t)) = E[e^{-r(T-t)}\max(K - \tilde{S}(T), 0)]$$

with

$$d\tilde{S} = r\tilde{S}dt + \sigma\tilde{S}dW.$$

This creates a crucial connection between the approximation based on the Monte Carlo method and the partial differential equations [14].

4.2 Pricing American Options

The price of American options can be obtained by maximizing over all the stopping times $\tau \in [0, T]$ strategies. This depends on the stock price up to the stopping time [14, 29],

$$f_A(t, s) \equiv \max_{t \le \tau \le T} E[e^{-r(\tau-t)} \max(K - S(\tau), 0)|S(t) = s]. \tag{4.4}$$

For calculating the optimal selling strategy for an American option, assume that the option is only allowed to be exercised at the discrete time levels $0, \Delta t, 2\Delta t, \ldots, T$. Consider the small time step $(T - \Delta t, T)$ and by assumption the option is not exercised in that step. Therefore, the value $f(t, s)$ holds, where $f(T, s) = \max(K - s, 0)$ and for $T - \Delta < t < T$

$$f_t + rSf_S + \tfrac{1}{2}\sigma^2 S^2 f_{SS} = rf. \tag{4.5}$$

If a fixed stock price $s = S(T - \Delta t)$, holds the expression $f(T - \Delta t, s) < \max(K - s, 0)$. When keeping the option, the option gives the expected value $f(T - \Delta t, s)$. Observe that the expected value when keeping the options is less than the value $\max(K - s, 0)$, which is obtained by selling the option at time $T - \Delta t$. Therefore, one can conclude that it is optimal to sell if $f(T - \Delta t, s) < \max(K - s, 0) \equiv f_F$. By modifying the initial data at $t = T - \Delta t$ to $\max(f(T - \Delta t, s), f_F)$ and repeat the step (4.5) for $(T - 2\Delta t, T - \Delta t)$. The price of an American option can be obtained as the limit of this solution as $\Delta t \to 0$ [14].

Consider an American put option, which gives the option holder the right to sell the underlying asset at the exercise price at any time before the option expires. Let K be the exercise price, T the maturity time, r the risk free interest rate, and σ to be the volatility. The price $S(t)$ of the option can be obtained from the solution of the linear complementary problem

$$\begin{aligned}
&\lambda = \frac{\partial S(t)}{\partial t} + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 S(t)}{\partial x^2} + rx\frac{\partial S(t)}{\partial x} - rS, \quad x > 0 \quad \text{and} \quad t \in [0, T],\\
&[S(t) - (K - x)]\lambda = 0, \quad x > 0 \quad \text{and} \quad t \in [0, T],\\
&S(t) - (K - x) \ge 0, \lambda \ge 0, \quad x > 0 \quad \text{and} \quad t \in [0, T],\\
&S(t) = \max(K - x, 0), \quad x > 0 \quad \text{and} \quad t = T,\\
&S(t) = K, \quad x = 0 \quad \text{and} \quad t \in [0, T],\\
&S(t) \to 0, \quad x \to \infty \quad \text{and} \quad t \in [0, T],
\end{aligned}$$

where t denote the time and x the value of the underlying asset [22].

Consider a general class of continuous-time American option pricing by specifying a process $U(t), 0 \leq t \leq T$ that represents the discounted payoff from exercise at time t, and a class of admissible stopping time τ with values in the interval [0,T]. The optimal expected discounted payoff is

$$\sup_{\tau \in \mathcal{T}} E[U(\tau)].$$

The payoff to the holder of the option from exercise at time t is $\tilde{h}(X(t))$, where $\tilde{h}$ denotes a non negative payoff function. Suppose the existence of an instantaneous short rate process $\{r(t), 0 \leq t \leq T\}$, the pricing problem are denoted by

$$\sup_{\tau \in \mathcal{T}} E\left[e^{-\int_0^\tau r(u)du} \tilde{h}(X(\tau))\right].$$

The assumption is taken with respect to the risk-neutral measure, which is implied in the form of discounting in this expression. By making the discount factor a component of $X(t)$ it can absorb the discounting into the function $\tilde{h}$. It is normal to consider the admissible stopping rules τ to be functions of the current state in this Markovian context, augmenting the state vector if necessary. Suppose the option expires at T. The value of the option at time 0 is

$$\sup_{\tau \in \mathcal{T}} E\left[e^{-r\tau}(K - S(\tau))^+\right],$$

where τ are stopping times with respect to S, with values in the interval [0,T]. To achieve this supremum, denote τ^* as the optimal stopping time with the form

$$\tau^* = \inf\{t \geq 0 : S(t) \leq b^*(t)\},$$

with an optimal exercise boundary b^* (see Fig.4.1).

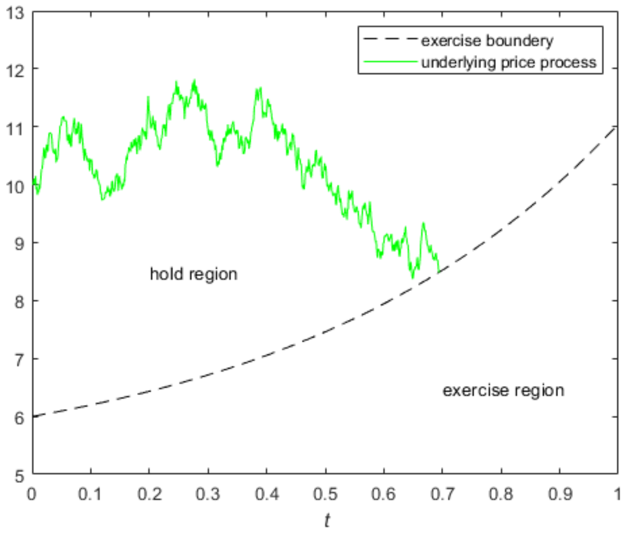


Figure 4.1: Hold and exercise region for a sample path which hits the optimal exercise boundary

Using simulation methods for pricing American options, limit the options so that it can only be exercised at a defined collection of exercise opportunities $t_1 < t_2 < \cdots < t_m$. The discrete-time process $X_0 = X(0), X_1, \ldots, X_m$ is a Markov chain on $\mathbb{R}^d$. Consider American option pricing to be based on simulation of this Markov chain, and simulating X_{i+1} from X_i may imply simulating $X(t)$ values with $t_i < t < t_{i+1}$ time steps. To apply a time discretization using a time step smaller than the intervals $t_{i+1} - t_i$ between the exercise dates, might be necessary [19].

4.3 Multilevel Monte Carlo

Consider Monte Carlo simulation with different time-steps $h_\ell = M^{-\ell}T$, with $\ell = 0, 1, \ldots, L$, number of simulations M, and time to maturity T, for a given the Brownian motion $W(t)$. Let P denote the payoff $f(S(T))$, $\hat{S}_{\ell,M^\ell}$ denote the approximations of $S(T)$ and $\hat{P}_\ell$ denote the approximations of P. This is done by using a numerical discretization with time-step h_ℓ. The idea is to write the expected payoff with a fine discretization using 2^ℓ uniform time-steps as a telescopic sum. Let $\hat{P}^\ell$ be the simulated payoff with a discretization using 2^ℓ uniform time-steps,

$$E[\hat{P}_L] = E[\hat{P}_0] + \sum_{\ell=1}^{L} E[\hat{P}_\ell - \hat{P}_{\ell-1}].$$

The Multilevel Monte Carlo method estimates each of the expectations on the right-hand side independently, in a way to minimize the computational complexity. For N_0 number of samples, let $\hat{Y}_0$ be an estimator for $E[\hat{P}_0]$, and $\hat{Y}_\ell$ for $\ell > 0$ be an estimator for $E[\hat{P}_\ell - \hat{P}_{\ell-1}]$ using N_ℓ paths. The estimator used, is a mean of N_ℓ independent samples. For $\ell > 0$,

$$E[\hat{P}_\ell - \hat{P}_{\ell-1}] \approx \hat{Y}_\ell = N_\ell^{-1} \sum_{i=1}^{N_\ell} \left(\hat{P}_\ell^{(i)} - \hat{P}_{\ell-1}^{(i)} \right). \tag{4.6}$$

The quantity $\hat{P}_\ell^{(i)} - \hat{P}_{\ell-1}^{(i)}$ comes from two discrete approximations with different time-steps, a fine and a coarse, but the same Brownian motion paths. This is done with various selections of simulations and time-steps for each level. The magnitude depended on the strong convergence properties of the scheme. Let V_ℓ denote the variance of a single sample $\hat{P}_\ell^{(i)} - \hat{P}_{\ell-1}^{(i)}$. The Complexity Theorem (see Theorem 7) shows that the convergence rate of V_l as $l \longrightarrow \infty$ determines the efficiency of the Multilevel approach. To ensure a better efficiency, (4.6) may be modified, and different estimators of $\hat{P}$ could be used on the fine and coarse discretization levels of $\hat{Y}_\ell$,

$$E[\hat{P}_\ell] = E[\hat{P}_0] + \sum_{\ell=1}^{L} E\left[\hat{P}_\ell^f - \hat{P}_{\ell-1}^c\right],$$

where $\hat{P}_\ell^f, \hat{P}_{\ell-1}^c$ are the estimators using 2^ℓ and $2^{\ell-1}$ steps respectively in the computation of $\hat{Y}_\ell$. Provided that

$$E[\hat{P}_\ell^f] = E[\hat{P}_\ell^c], \tag{4.7}$$

the telescoping sum property is maintained. To implement this, the Brownian motion increments are first constructed for the simulation of the discrete path leading to the evaluation of the term $\hat{P}_{\ell}^{(i)}$, continuing to sum them in groups of size M to give the discrete Brownian motion increments for the evaluation of $\hat{P}_{\ell-1}^{(i)}$. The variance of this estimator is $V[\hat{Y}_\ell]$=$N_\ell^{-1}V_\ell$, where V_ℓ is the variance of a single sample. Using stratified sampling or a zero-mean control variate to reduce the variance, the same inverse dependence on N_ℓ would apply. The Variance of the combined estimator $\hat{Y} = \sum_{\ell=0}^{L} \hat{Y}_\ell$ is,

$$V[\hat{Y}] = \sum_{\ell=0}^{L} N_\ell^{-1} V_\ell.$$

The cost of computation is equal to

$$\sum_{\ell=0}^{L} N_\ell h_\ell^{-1}.$$

Assume that $L \gg 1$ and consider the behaviour of V_ℓ as $l \longrightarrow \infty$. In the particular case of the Euler discretization and the Lipschitz payoff function, provided that the term $a(S,t)$ and $b(S,t)$ satisfy certain conditions, there is a $O(h)$ weak convergence and $O(h^{1/2})$ strong convergence. Hence, as $\ell \longrightarrow \infty$,

$$E\left[\hat{P}_\ell - P\right] = O(h_\ell), \tag{4.8}$$

and

$$E\left[||\hat{S}_{\ell,M^\ell} - S(T)||^2\right] = O(h_\ell). \tag{4.9}$$

From the Lipschitz property (3.2), it follows that

$$\mathrm{V}[\hat{P}_\ell - P] \leq E[(\hat{P}_\ell - P)^2] \leq c^2 E[||\hat{S}_{\ell,M^\ell} - S(T)||^2]. \tag{4.10}$$

Combining (4.10) with (4.9) gives $V[\hat{P}_\ell - P] = O(h_\ell)$. Further,

$$\begin{aligned} & (\hat{P}_\ell - \hat{P}_{\ell-1}) = (\hat{P}_\ell - P) - (\hat{P}_{\ell-1} - P) \\ \longrightarrow \quad & \mathrm{V}[\hat{P}_\ell - \hat{P}_{\ell-1}] \leq \left((\mathrm{V}[\hat{P}_\ell - P])^{1/2} + (\mathrm{V}[\hat{P}_{\ell-1} - P])^{1/2}\right)^2. \end{aligned}$$

Hence, for the simple estimator (4.6), the single sample variance V_ℓ is $O(h_\ell)$ and the optimal choice for N_ℓ is asymptotically proportional to h_ℓ. Setting $N_\ell = O(\epsilon^{-2}Lh_\ell)$, the variance of the combined estimator $\bar{Y}$ is $O(\epsilon^2)$. If L is now chosen such that

$$L = \frac{\log\epsilon^{-1}}{\log M} + O(1), \tag{4.11}$$

as $\epsilon \longrightarrow 0$, then $h_L = M^{-L} = O(\epsilon)$. Therefore the bias error $E[\hat{P}_L - P]$ is $O(\epsilon)$, due to (4.6). Consequently, a MSE which is $O(\epsilon^2)$ are obtained, with a computational complexity which is $O(\epsilon^{-2}L^2) = O(\epsilon^{-2}(\log\epsilon)^2)$ [18].

4.3.1 Complexity Theorem

A way to compare the computational cost of the standard Monte Carlo with the Multilevel Monte Carlo is by computational complexity theory. This is the cumulative number of times the random number generator is called for a certain method. The cost of regular Monte Carlo is proportional to the number of paths multiplied by the cost of simulating a single path. To measure the computational complexity of the standard Monte Carlo process, an exact analytical solution to an SDE is rarely feasible in most applications. This means one has to use a combination of regular Monte Carlo and a numerical scheme for solving SDEs, such as the Euler-Maruyama scheme or Milsteins schemes. Let $\hat{Y}$ be the numerical scheme's approximation of Y. The mean square error gives us the equation:

$$\mathrm{MSE} \equiv E[(\hat{Y} - Y)^2]. \tag{4.12}$$

Rewriting equation (4.12) yields

$$\begin{aligned} E[(\hat{Y} - [Y])^2] &= E[(\hat{Y} + E[\hat{Y}] - E[\hat{Y}] - Y)^2) \\ &= E[E[(\hat{Y} - E[\hat{Y}])^2) + E[(E[\hat{Y}] - Y)^2], \end{aligned}$$

given the first term to be the statistical error of the Monte Carlo variance. Since when approximating a time continuous process with a discrete process, the second term is the bias of the numerical discretization scheme [21].

It is often difficult to precisely simulate the random variable of interest, in this case, the payoff P in certain applications. This is often the case when an application that involves the simulation of SDEs is performed. Consider Y to be an approximation to $E[P]$, the mean square error is given by

$$\mathrm{MSE}(Y) = E[(Y - E[P])^2] = \mathrm{Var}[Y] + (E[Y] - E[P])^2. \tag{4.13}$$

Let $\hat{Y}$ be the Multilevel estimator

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

where

$$\begin{aligned} E[\hat{Y}] &= E[P_L], \\ \mathrm{Var}[\hat{Y}] &= \sum_{l=0}^{L} N_l^{-1} V_l. \end{aligned}$$

From this, the mean square error in (4.13) can be written as

$$E[(\hat{Y} - E[P])^2] = \mathrm{Var}[\hat{Y}] + (E[\hat{Y}] - E[P])^2 = \mathrm{Var}[\hat{Y}] + (E[P_L - P])^2.$$

This leads to the following complexity theorem [11].

Theorem 7 (Theorem 4.3 in [11]). *Let P be a function of a solution to*

$$dS(t) = a(S,t)dt + b(S,t)dW(t), \quad 0 < t < T, \tag{4.14}$$

for a given Brownian motion path $W(t)$; let $\hat{P}_\ell$ be the corresponding approximation using the discretization at level ℓ, i.e. with 2^ℓ steps of width $h_\ell = 2^{-\ell}T$. If there exist independent estimators $\hat{Y}_\ell$ of computational complexity C_ℓ based on N_ℓ samples and there are positive constants $\alpha \geq \frac{1}{2}, \beta, c_1, c_2, c_3$ such that

1. $E[\hat{Y}_\ell] = \begin{cases} E[\hat{P}_0], & \text{if } \ell = 0 \\ E[\hat{P}_\ell - \hat{P}_{\ell-1}], & \text{if } \ell > 0 \end{cases}$

2. $|E[\hat{P}_\ell - P]| \leq c_1 h_\ell^\alpha$

3. $V[\hat{Y}_\ell] \leq c_2 h_\ell^\beta N_\ell^{-1}$

4. $C_\ell \leq c_3 N_l h_l^{-1}$

Then there is a constant c_4 such that for any $\varepsilon < e^{-1}$, there are values for L and N_ℓ resulting in a Multilevel estimator

$$\hat{Y} = \sum_{\ell=0}^{L} \hat{Y}_\ell$$

with a mean square error with bound

$$MSE \equiv E\left[\left(\hat{Y} - E[P]\right)^2\right] < \varepsilon^2$$

with a computational complexity bound

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

The theorem shows the importance of the parameter β which defines the convergence of the variance V_ℓ as $\ell \longrightarrow \infty$. The optimal N_ℓ is proportional to $\sqrt{V_\ell h_\ell} = O(h_\ell^{(\beta+1)/2})$ in this limit, implying that the computational effort $N_\ell h_l^{-1}$ is proportional to $O(h_\ell^{(\beta-1)/2})$. This indicates that for $\beta > 1$ the computational effort is primarily expended on the coarsest levels, for $\beta < 1$ it is primarily expended on the finest levels, and for $\beta = 1$ it is spread across all levels. If the payoff function $f(S(T))$ has a discontinuity, the convergence is degraded. This implies that, for a given time-step h_ℓ, a fraction of the paths of size $O(h_\ell^{1/2})$ will have a final $\hat{S}_{\ell,M^\ell}$, which is $O(h_\ell^{1/2})$ from the discontinuity. With the use of the Euler discretization, this fraction of the paths have an $O(1)$ probability of $\hat{P}_\ell - \hat{P}_{\ell-1}$ being $O(1)$. This is due to the fact that $\hat{S}_{\ell,M^\ell}$ and $\hat{S}_{\ell,M^{\ell-1}}$ being on opposite sides of the discontinuity, and leads to $V_\ell = O(h_\ell^{1/2})$ and $\beta = \frac{1}{2}$. The overall complexity is $O(\varepsilon^{-2.5})$, which is better than the $O(\varepsilon^{-3})$ complexity of the standard Monte Carlo method with Euler discretization. Consequently, due to that the weak order of convergence is still $O(h_\ell)$, leading to $\alpha = 1$, see [11] for further details.

Chapter 5

Numerical Results

In this section, we present and analyze the Multilevel Monte Carlo simulation (MLMC) in comparison with Monte Carlo simulation (MC). We approximate numerous simulations of various input variables, such as number of simulations M, number of time-steps N, and levels L. Furthermore, we showcase the differences in simulations with initial stock price s, admissible starting times t, variance σ and interest rate r. We will represent the analysis in tables and figures to visualize the convergence rates and price comparisons between the algorithms. An important feature of the differences in the methods is the computational times, which are also visualised. The aim of this research is to approximate

$$f_A(t,s) \equiv \max_{t \leq \tau \leq T} E[e^{-r(\tau-t)} \max(K - S(\tau), 0) | S(t) = s]. \tag{5.1}$$

by implementing the Multilevel Monte Carlo method.

By using Multilevel Monte Carlo together with a numerical scheme, Theorem 7 shows that when compared to the standard Monte Carlo with the same numerical scheme, the computational cost can be reduced. The Theorem 7 stated that with the Multilevel Monte Carlo computational cost of $O(\epsilon^{-2})$ can be achieved when a numerical scheme with $\beta > \gamma$ is used. On the other hand, the standard Monte Carlo has a computational cost of $O(\epsilon^{-3})$. We wanted to analyse the computational times for different methods further.

We present the first results in Table 5.1 and Table 5.2, where we compare the convergence rate for Multilevel Monte Carlo method to standard Monte Carlo method. The tables are also showcasing the computational time for the respective method of simulation used, for Table 5.1 it utilizes constant numbers of time-step increments N, and tests for different number of simulations M with a factor of 2. We decide to test our method with the following option parameters and initial values:

$$s = 100,\ K = 95,\ L = 3,\ N = 800,\ \Delta t = 0.00125,\ T = 1,\ r = 0.05, \text{ and } \sigma = 0.03.$$

We performed this to analyze the expected differences in computational time for the separate methods as the number of simulations increased.

Number of simulations M	American Put Price		Computational time	
	MLMC	MC	MLMC	MC
2^{10}	7.6691	7.1433	1.113176	4.545195
2^{11}	7.9827	7.2198	1.346433	7.315279
2^{12}	7.9882	7.7166	1.733293	12.649382
2^{13}	7.7486	7.5617	2.732281	22.601926
2^{14}	7.3689	7.5900	4.682595	39.858427
2^{15}	7.4999	7.5338	7.360962	66.943823

Table 5.1: Approximation of put option price as M increases and compassion of computational time.

From Table 5.1, we can observe that the as M increases the price for both methods converge to an approximate of a similar value. Furthermore, the computational cost for each method is increasing separately. The MC method quickly becomes ineffective and requires a higher computational burden to get similar convergence rates. This results in an advantage to the MLMC method as the more time efficient method. The results exhibit that the MLMC method has similar convergence rate and can cause significant reductions in the computational time required to achieve the desired absolute error.

Table 5.2 comprises a constant number of simulations M, where we double the number of time-steps N for each consecutive experiment. We decide to test our method with the following option parameters and initial values:

$$s = 100,\ K = 95,\ L = 3,\ M = 2^{10},\ T = 1,\ r = 0.05,\ \text{and } \sigma = 0.03.$$

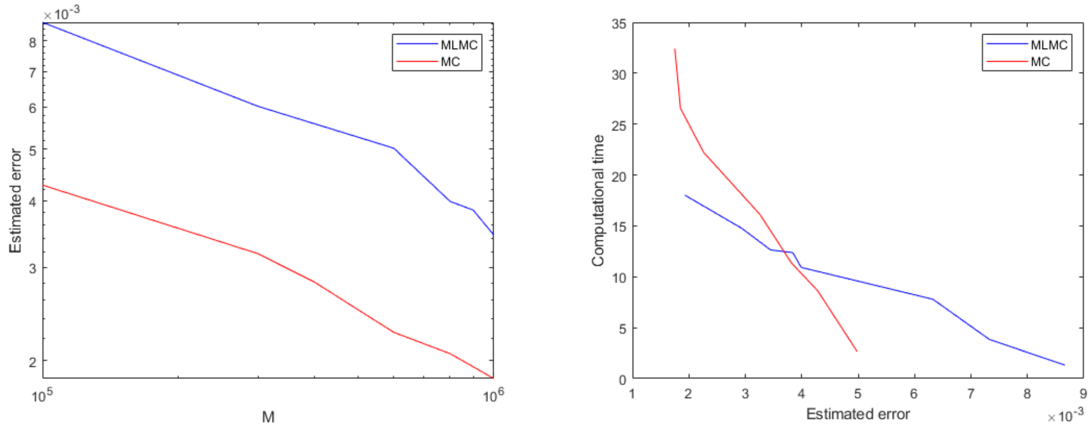
We performed this to analyze the expected differences in computational time for the separate methods as the number of steps increases. An important factor is that the number of steps changes with each level of the multilevel simulation. The course and fine level of time-steps varies depending on the current level (see Section 4.3). This is implemented to get a more efficient computational process and still achieve similar convergence rates.

Number of time-steps N	American Put Price		Computational time	
	MLMC	MC	MLMC	MC
200	8.1480	6.9468	0.081593	0.300559
400	6.7251	7.1449	0.312090	1.138500
800	7.6691	7.1433	1.113176	4.545195
1600	6.5109	7.2512	4.489613	19.672390
3200	7.0686	7.7480	18.354861	81.264753
6400	7.2991	7.4566	75.127351	327.282364

Table 5.2: Approximation of put option price as N increases and compassion of computational time.

From Table 5.2, we can infer similar results as in Table 5.1, that there comprises various differences in computational time for the two separate methods, while both MLMC and MC estimate an approximate value as the number of time-steps N increases. As for analysing Table 5.1, Table 5.2 also exhibits that the MC method quickly becomes ineffective and requires more time to get similar convergence rates. This results in an advantage to the MLMC method as the more computationally cost efficient method. The results exhibit that the MLMC method provides faster convergence and can cause significant reductions in the number of time-steps required.

The next section specializes to analyze how the methods MLMC and MC converge (see Sections 4.3 and 4.3.1). We have two separate instances of comparing the estimated error and computational time for MLMC and MC. We created Fig. 5.1 before we had to estimate the price. Therefore, we had the exact solution, and the estimated error should approach exactly 0. The subplot (a) shows the estimated error as the number of simulations increases. Subplot (b) shows the computational time as an effect of the estimated error converging.



(a)

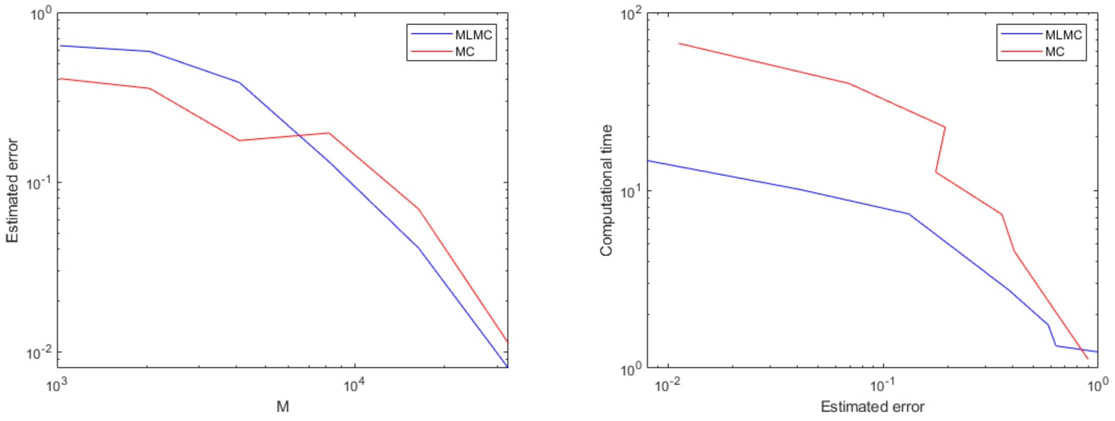
(b)

Figure 5.1: Evolution of (a) the absolute error as M increases, and (b) computational time as estimated error converges

One can observe from Fig. 5.1 (a) that as the amount of simulations increases the rates of convergence are similar. We analyzed from the similar slope patterns and as expected the models are approaching zero. Fig. 5.1 (b) shows that the models have different computational time performances when running the large number of simulations required to get accurate results. The next set of Figs. 5.2 (a) and (b) are log-log plots for estimating American option pricing. We used the codes from Appendix A with an increasing amount of simulation to test how close it can approach the true price. The problem is that a true analytical solution can not be calculated for American put options. Therefore, we ran an approximation with an extensive amount of simulations and time-steps to get as close as possible. This was performed with the initial values:

$$s = 100,\ K = 95,\ L = 3,\ M = 2^{15},\ \Delta t = 6.25 \times 10^{-5},\ T = 1,\ r = 0.05,\ \text{and } \sigma = 0.03,$$

where we obtained the approximation of $f_A(100, 0) = 7.570447994773782$, and we will use this as a true price for the preceding comparisons.



(a)

(b)

Figure 5.2: Evolution of (a) the absolute error as M increases, and (b) the computational time as estimated error decreases compared with estimated price

Fig. 5.2 (a) further proves that the MLMC and MC approaches the estimated true value similarly as Fig. 5.1, in other words, as we increase the amount of simulation the methods converge. Figure 5.2 shows that the MC method requires more computational time to reach the same level of absolute error. This is when we increase the number of simulations large enough to get an appropriate approximation.

We dedicate this next section to analyze the payoff evolution of the separate methods. Figure 5.3 shows different payoff functions depending on the initial time point $t \in [0, T]$. The different plots indicate the selections of variance σ for each of the separate algorithms.

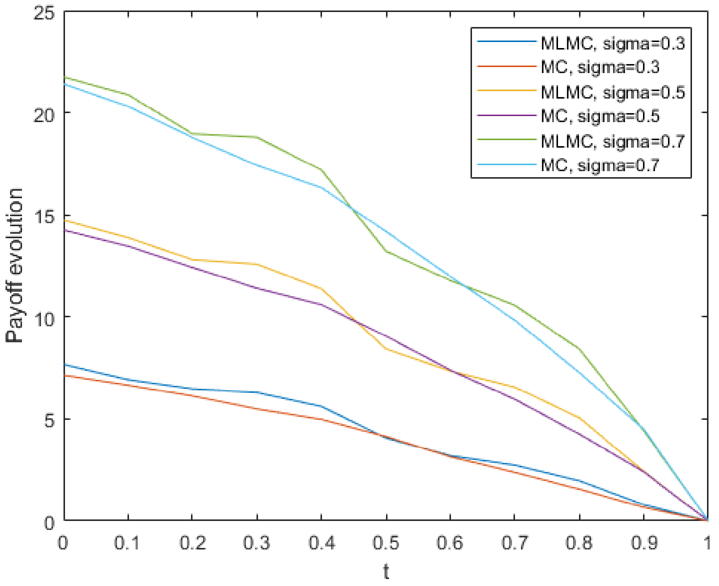


Figure 5.3: Payoff evolution for every initial time point t

Figure 5.3 shows that the MLMC simulation and the MC simulation have similar payoff evolution as the initial time point $t \longrightarrow T\ \forall t \in [0, T]$. The differences in variance do not show the tendency of more inaccuracy as variables change. Furthermore, it proves that the MLMC method is a valid method of approximating the American put option price when compared to MC. We want to know if the MLMC method can accurately determine the payoff where the number of time-steps N, number of simulations M, sigma $\sigma = 0.3$ and rate $r = 0.05$ are consistent. From (4.4) we can determine the payoff of all admissible stopping times $\tau \in [t, T]$, where method calculated the payoff for a given $s \in [80, 150]$.

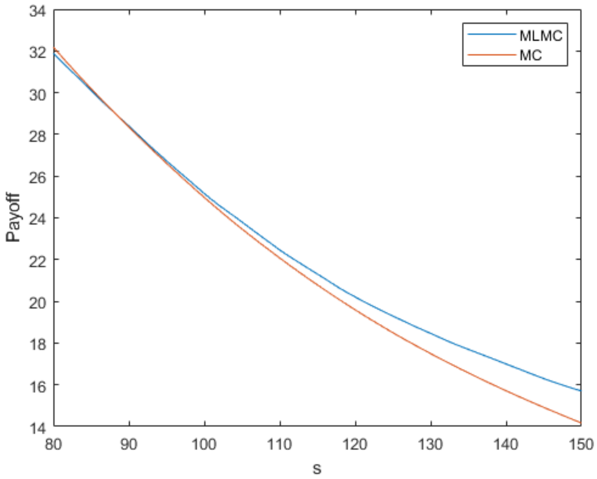


Figure 5.4: Payoff evolution for changes in initial stock price s

Figure 5.4 shows highly accurate approximations for starting price more realistically priced, i.e. closer to the strike price K. The further away, the more difficult it is to get a good estimated payoff; furthermore, the two methods provide similar estimates and even though there is no exact solution we believe these estimations are sufficient at pricing the American put option.

We began this chapter with discussing the expected convergence rates and differences in computational time between Multilevel Monte Carlo and standard Monte Carlo simulation. The convergence rates when comparing the two methods are consistent with the expectations provided in Chapter 4. When provided with the same initial values, the two algorithms approximate similar estimates and when either the step-sizes or the amount of simulations increase we get the same expected convergence rates. We showcased the results in numerous tables and figures to accurately provide necessary information to analyze the two methods. The analysis provided regarding each method are done in a way to showcase the expected results. The second reason to utilize the MLMC method is the more efficient computational process. The method quickly becomes more effective at providing reasonable estimates at less computational time. This is due to the levels usage of different amounts of simulations and time-steps. It is less taxing if we compute fewer simulations at the time-step size and utilize larger approximations at larger time-step sizes. We proved, showcased, and analyzed the data in figures to accurately, represent the differences in computational time.

Chapter 6

Conclusion

The aim of the project was to implement the Multilevel Monte Carlo method for valuing American options. We started by introducing the reader to the necessary and relevant literature of the subject, in order to research and provide critical evaluation of the method. Most importantly, using different numerical methods for solving ordinary differential equations and stochastic differential equations. Along with critically assessing the differences of the standard Monte Carlo and the Multilevel Monte Carlo, evaluation of the primary benefits of each method has been performed. We performed implementations and combinations of numerical methods and simulations. The main theoretical result of this research is that by using Multilevel Monte Carlo together with a numerical scheme, we can reduce the computational cost significantly more than by using the standard Monte Carlo with the same numerical scheme. We could also observe that as the number of simulations and/or number of time-steps increase, the rate of convergence is following the same rate as expected. Furthermore, it has been observed that the standard Monte Carlo method requires a higher computational burden to reach the same level of absolute error. Consequently, it explains the advantages of repeating samples when taken on various levels of accuracy. From the results observed in this research, we can conclude that the Multilevel Monte Carlo method drastically reduces the computational cost at the same level of convergence, leading to an advantage for simulating American option pricing.

Finally, further research in this topic can be applied in a variety of directions:

1. Consider the Multilevel Monte Carlo method to be extended to more complex financial models than being researched so far.

2. Allow models with volatility jumps or models that can predict the convergence rates.

3. Improved methods of the standard Monte Carlo method, such as Quasi-Monte Carlo and Multi-Index Monte Carlo.

Bibliography

[1] Doghonay Arjmand. *Highly Accurate Difference Schemes for the Numerical Solution of Third-Order Ordinary and Partial Differential Equations*. TRITA-CSC-E 2010:012. Skolan för datavetenskap och kommunikation, Kungliga Tekniska högskolan, 2010.

[2] U. M. (Uri M.) Ascher. *Computer methods for ordinary differential equations and differential-algebraic equations*. Society for Industrial and Applied Mathematics, Philadelphia, 1998.

[3] Allaberen Ashyralyev and Doghonay Arjmand. A note on the Taylor's decomposition on four points for a third-order differential equation. *Applied Mathematics and Computation*, 188:1483–1490, 05 2007.

[4] Allaberen Ashyralyev and Doghonay Arjmand. Taylor's decomposition on four points for solving third-order linear time-varying systems. *Journal of the Franklin Institute*, 346:651–662, 09 2009.

[5] Giovanni Barone-Adesi and Robert E. Whaley. Efficient Analytic Approximation of American Option Values. *The Journal of finance (New York)*, 42(2):301–320, 1987.

[6] Jérôme Barraquand and Didier Martineau. Numerical Valuation of High Dimensional Multivariate American Securities. *Journal of financial and quantitative analysis*, 30(3):383–405, 1995.

[7] R. J. Birge. Quasi-Monte Carlo Approaches to Option Pricing. *Department of Industrial and Operations Engineering, The University of Michigan*, 1995.

[8] Fischer Black and Myron Scholes. The Pricing of Options and Corporate Liabilities. *The Journal of political economy*, 81(3):637–654, 1973.

[9] Mark Broadie and Paul Glasserman. Pricing American-style securities using simulation. *Journal of economic dynamics & control*, 21(8):1323–1352, 1997.

[10] Mark Broadie, Paul Glasserman, and Gautam Jain. Enhanced Monte Carlo estimates for American option prices. *The Journal of derivatives*, 5(1):25–, 1997.

[11] Sylvestre Burgos and Mike Giles. Computing Greeks Using Multilevel Path Simulation. *Springer Proceedings in Mathematics and Statistics*, 23, 01 2012.

[12] J. C. (John Charles) Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, Ltd, Chichester, West Sussex, England, third edition. edition, 2016 - 2016.

[13] E.R. Caflish and S. Chaudhary. Monte Carlo Simulation for American Options. *Division of physical sciences, University of California Los Angeles*, 2005.

[14] J. Carlsson, K. Moon, A. Szepessy, R. Tempone, and G. Zouraris. Stochastic Differential Equations: Models and Numerics. *Kungliga Tekniska Högskolan*, 2018.

[15] Peter Carr. Randomization and the American Put. *The Review of financial studies*, 11(3):597–626, 1998.

[16] Jacques F. Carriere. Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance, mathematics & economics*, 19(1):19–30, 1996.

[17] M.J. Flegal. Monte Carlo Standard Errors for Markov Chain Monte Carlo. *Faculty of the graduate school of the university of Minnesota*, 2008.

[18] Michael B Giles. Multilevel Monte Carlo Path Simulation. *Operations research*, 56(3):607–617, 2008.

[19] Paul Glasserman. *Monte Carlo methods in financial engineering*. Applications of Mathematics ; 53. Springer, New York, 2003.

[20] David B. Hertz. Risk analysis in capital investment. *Harvard Business Review*, 1979.

[21] John Hull. *Options, futures, and other derivatives*. Pearson, 9.ed edition, 2014.

[22] S Ikonen and J Toivanen. Operator splitting methods for American option pricing. *Applied mathematics letters*, 17(7):809–814, 2004.

[23] Malvin H Kalos and Paula A Whitlock. *Monte Carlo methods*. John Wiley & Sons, 2009.

[24] Jack P.C Kleijnen, A.A.N Ridder, and R.Y Rubinstein. Variance Reduction Techniques in Monte Carlo Methods. *ssrn*, 2010-117, 2010.

[25] Francis A. Longstaff and Eduardo S. Schwartz. Valuing American Options by Simulation: A Simple Least-Squares Approach. *The Review of financial studies*, 14(1):113–147, 2001.

[26] Steven B Raymar and Michael J Zwecher. Monte Carlo estimation of American call options on the maximum of several stocks. *The Journal of derivatives*, 5(1):7–, 1997.

[27] Sheldon M. Ross. *Simulation*. Academic Press, Amsterdam, 5th ed. edition, 2013.

[28] Jan R. M. Röman. *Analytical Finance: Volume I The Mathematics of Equity Derivatives, Markets, Risk and Valuation*. Springer International Publishing, Cham, 1st ed. 2017. edition, 2017.

[29] Dmitrii S. Silvestrov. *American-type options : stochastic approximation methods. 1*. Studies in mathematics 56. De Gruyter, Göttingen, 2014.

[30] Nick Thomopoulos. *Essentials of Monte Carlo simulation: Statistical methods for building simulation models*. Springer-Verlag, 11 2013.

[31] James A. Tilley. Valuing American Options in a Path Simulation Model. *Transactions of the Society of Actuaries*, 1993.

[32] J.N Tsitsiklis and B van Roy. Optimal stopping of Markov processes: Hilbert space theory, approximation algorithms, and an application to pricing high-dimensional financial derivatives. *IEEE transactions on automatic control*, 44(10):1840–1851, 1999.

Appendix A

Algorithm

A.1 Basic Monte Carlo for Calculating Simulated Prices

Monte Carlo function for calculating average estimated simulated stock prices.

```
function [ep] = MCCode(M,N,dW,k,tau,t,s)
%Basic Monte Carlo for calculating simulated prices
%
%   [ep] = MCCode(M,N,dW,k,tau,t,s)
%
%   Inputs:
%   M          - Amount of simulations
%   N          - Amount of time-steps
%   dW         - Brownian motion path
%   k          - Level specific variable for the amount of brownian steps
%   tau        - Admissible stopping times
%   t          - Admissible starting times
%   s          - Current price of the underlying asset
%
%   Outputs:
%   ep         - Mean values of simulated prices
%
%   Notes:
%    (1) Default strike price K = 95, dt = 0.01, and L = 4. Edit function
%    to change desired parameters.

    K     = 95;
    T     = 1;
    dt    = T/N;
    r     = 0.05;
    sigma = 0.3;
    S     = s*ones(M,1); % S(0) for all realizations
   for j=1:N
       if j == 1 % Brownian path index
```

```
            kindex = 1:k;
        else
            kindex = kindex(end):j*k;
        end
        S        = S + S.*(r*dt+sigma*sum(dW(1:M,kindex),2)); % Processes at
            next time step
        ep(j) = mean((exp(-r*(tau-t))*max(K-S(:),0))); % Mean calculation of
            the estimated prices
    end
end
```

A.2 Monte Carlo Simulation

Monte Carlo function for estimating optimal American put option payoff.

```
function fA = American_Price_MC(t,s)
%Estimate American put option price using a Monte Carlo simulation
%
%   [fA] = American_Price_MC(t,s)
%
%   Inputs:
%   t          - Vector of admissible starting times [0,T]
%   s          - Vector of current price of the underlying asset.
%
%   Outputs:
%   fA         - Optimal estimated payoff
%
%   Notes:
%    (1) Inputs for t should be decimal values between [0,T], where T = 1.
%    (2) Default strike price dt = 0.01, and L = 4. Edit function
%    to change desired parameters.

randn('state',0)
dt = 0.01; % step-size
N0 = 1 / dt; % Amount of time-steps
M0 = 2^10; % Amount of simulations

dW = sqrt(dt) * randn(M0,N0); % Brownian motion
for x = 1 : size(s,2) % Loop for initial stock price
    fa = zeros(1,size(t,(2)));
    for j = 1 : size(t,(2)) % Loop for admissible starting times
        N   = round((1 - t(j)) / (1 / N0)); % Amount of time-steps for [t,T]
        tau = 1/N0 + t(j) : 1 / N0 : 1; % Admissible stopping times
        est = zeros(1,size(t,(2)));
        for i = 1 : size(tau,2) % Loop for stopping times
```

```
            Ep   = MCCode(M0,N,dW,1,tau(i),t(j),s(x)); % Expected value for
                Monte Carlo simulation
            est(i) = max(Ep);
        end
        fa(j) = max(est);
    end
    fA(x) = max(fa);
end
end
```

A.3 Multilevel Monte Carlo Simulation

Multilevel Monte Carlo function for estimating optimal American put option payoff.

```
function fA = American_Price_MLMC(t,s)
%Estimate American put option price using a Multilevel Monte Carlo
%simulation
%
%   [fA] = American_Price_MLMC(t,s)
%
%   Inputs:
%   t          - Vector of admissible starting times [0,T]
%   s          - Vector of current price of the underlying asset.
%
%   Outputs:
%   fA         - Optimal estimated payoff
%
%   Notes:
%    (1) Inputs for t should be decimal values between [0,T], where T = 1.
%    (2) Default strike price M0 = 2^10, and L = 4. Edit function
%    to change desired parameters.

N0 = 100; % Amount of time-steps for L = 0
M0 = 2^10; % Amount of simulations
L  = 4; % Amount of levels
dW = sqrt(1 / (2^(L) * N0)) * randn(M0,2^(L) * N0); % Brownian motion
for x = 1 : size(s,2) % Loop for initial stock price
    fa = zeros(1,size(t,(2)));
    for j = 1 : size(t,(2)) % Loop for admissible starting times
        N   = round((1 - t(j)) / (1 / N0)); % Amount of time-steps for [t,T]
        tau = 1 / N0 + t(j) : 1 / N0 : 1; % Admissible stopping times
        t0  = linspace(0,1,N);
        est = zeros(1,size(t,(2)));
        for i = 1 : size(tau,2) % Loop for stopping times
            ep0 = MCCode(M0,N,dW,(2^(L)),tau(i),t(j),s(x)); % Expected value
            when L = 0
          sum = ep0;
          if L == 0
              est(i) = max(sum);
          end
          for k = 1 : L % Loop for levels
              tf = linspace(0,1,2^k*N); % Time-steps for fine simulation
              tc = linspace(0,1,2^(k-1)*N); % Time-steps for course simulation

             ep1f   = ...
             MCCode(M0/2^(k),2^(k)*N,dW,(2^(L)*N)/(2^(k)*N),tau(i),t(j),s(x));
             % Expected course value when L = k
             ep0c   = ...
             MCCode(M0/2^(k),2^(k-1)*N,dW,(2^(L)*N)/ ...
             (2^(k-1)*N),tau(i),t(j),s(x));
             % Expected fine value when L = k-1
             ep1fInt = interp1(tf,ep1f,t0);
             ep0cInt = interp1(tc,ep0c,t0);
             sum     = sum + ep1fInt - ep0cInt; % Estimate the expectation for
                 all levels
             est(i) = max(sum);
          end
      end
      fa(j) = max(est);
  end
  fA(x) = max(fa);
end
end
```

Appendix B

Bachelor Degree Objectives

In this section, we present the Swedish National Agency for Higher Education's criteria for Bachelor theses, which include five key thesis objectives. After each objective, there is a brief explanation of how we intend to achieve it.

B.1 Objective 1: Knowledge and understanding in the major field of the research

The student should demonstrate knowledge and understanding in the major field of study, including knowledge of the field's scientific basis, knowledge of applicable methods in the field, specialization in some part of the field and orientation in current research questions.

We believe our thesis fulfill this aim with the presented description, implementation and the numerical analysis of the current research. Chapter 2 demonstrates our knowledge and understanding in different numerical methods for solving ordinary differential equations and stochastic differential equations. Chapter 3 and Chapter 4 shows the understanding and knowledge of simulation methods in order to solve problems using random variable, and understanding the impact on how uncertainty affect the forecasting of models. Chapter 5 demonstrates the practical usage of previously mentioned knowledge.

B.2 Objective 2: Ability to search, collect and evaluate information

For Bachelor degree, the student should demonstrate the ability to search, collect, evaluate and critically interpret relevant information in a problem formulation and to critically discuss phenomena, problem formulations and situations.

We searched, collected, and assessed information not only from previous research but also from research in similar topics and other types of simulation methods. The introduction section describes the necessary and relevant literature of the subject to our research. It also provides description and critical evaluation of this related research topic. Chapter 5 also demonstrated the critical interpretation were we analysed the model implementation and its results.

B.3 Objective 3: Identify, formulate and solve problems

For Bachelor degree, the student should demonstrate the ability to independently identify, formulate and solve problems and to perform tasks within specified time frames.

By the content of our research paper, we fulfil this objective. The aim of our project was to implement the Multilevel Monte Carlo method for valuing American options. We identified, analyzed, and presented the methods successfully, and identified and understand the topic. We also presented standard Monte Carlo method for evaluating the benefits of the primary method that is being researched, in order to analyse the accuracy and advantages of this improved method. To perform the research question, we implemented a combination of numerical methods to simulate American option pricing models in the computer software Matlab. We have respected the specified time frames for the thesis during the entire process of working on this research.

B.4 Objective 4: Ability to present orally and writing problems and solutions in dialogue to different groups

For Bachelor degree, the student should demonstrate the ability to present orally and in writing and discuss information, problems and solutions in dialogue with different groups.

Our thesis presents the aim of our research and the methodology. We present the results and critically analyze them in order to make the results significantly more reliable. The presentations of the thesis will verify our knowledge that were acquired during this research, as well as gaining important opinions from other individuals and groups for further research and improvements.

B.5 Objective 5: Ability in the major field to make judgements in respect to scientific, societal and ethical aspects

For Bachelor degree, the student should demonstrate ability in the major field of study make judgments with respect to scientific, societal and ethical aspects.

This research contributes to valuable research in the field of financial engineering by producing new research in the topic, and suggesting future research for further findings. The scientific aspect is respected by using earlier research and findings in order to improve the specific field of research.