MIDDLE EAST TECHNICAL UNIVERSITY Ankara, Turkey

INSTITUTE OF APPLIED MATHEMATICS
http://iam.metu.edu.tr

Monte Carlo Methods in Computational Finance: From Variance Reduction to American Option Pricing

Fulya Seray Yıldırım

Abstract. This study develops a simulation framework for derivative pricing, progressing from the foundations of the Monte Carlo method to the Multilevel Monte Carlo (MLMC) technique, with American option pricing and Brownian bridge path construction as central themes. The theoretical development covers variance reduction techniques including antithetic variates, control variates, importance sampling, and stratified sampling applied to European and Asian options under the Black–Scholes model. The early exercise problem of American options is addressed through the Longstaff–Schwartz regression-based backward induction algorithm. The Brownian bridge is introduced as a path construction tool whose conditional interpolation formula provides the natural mechanism for coupling simulations across discretization levels in the MLMC framework. MLMC is developed from the telescoping identity, reducing computational complexity from $\mathcal{O}(\epsilon^{-3})$ to $\mathcal{O}(\epsilon^{-2}(\ln \epsilon)^2)$ for a target accuracy ϵ. Numerical experiments for European call, Asian call, and American put options confirm the theoretical complexity advantage of MLMC in the asymptotic regime, establish that the Brownian bridge contributes primarily through path coupling across discretisation levels rather than as a standalone variance reduction technique, validate the Longstaff–Schwartz algorithm through a correctly shaped early exercise boundary and stable price estimates, and reveal that naive integration of LSMC and MLMC fails to produce the variance decay required for multilevel efficiency.

Keywords. Multilevel Monte Carlo, Option Pricing, Variance Reduction, Brownian Bridge, Longstaff–Schwartz Algorithm

⋆ Submitted to IAM, *Financial Mathematics* in June 2026.

TERM PROJECT

Student : Fulya Seray Yıldırım
Advisor : Ömür Uğur

Contents

1 Introduction **4**
- 1.1 Motivation and Background . . . 4
- 1.2 The Computational Challenge . . . 4
- 1.3 The Multilevel Monte Carlo Framework . . . 5
- 1.4 Scope and Objectives . . . 5
- 1.5 Contributions and Outline . . . 5

2 Monte Carlo Methods for Option Pricing **7**
- 2.1 Introduction . . . 7
- 2.2 Theoretical Foundations . . . 7
  - 2.2.1 The Strong Law of Large Numbers . . . 7
  - 2.2.2 The Crude Monte Carlo Method . . . 8
  - 2.2.3 First Numerical Results . . . 8
- 2.3 Variance Reduction Techniques . . . 9
  - 2.3.1 Antithetic Variates . . . 9
  - 2.3.2 Control Variates . . . 10
  - 2.3.3 Importance Sampling . . . 13
  - 2.3.4 Stratified Sampling . . . 14

3 American Option Pricing via Monte Carlo **16**
- 3.1 Introduction . . . 16
- 3.2 American and Bermudan Contingent Claims . . . 17
  - 3.2.1 The Optimal Stopping Problem . . . 17
  - 3.2.2 Bermudan Approximations . . . 17
- 3.3 Dynamic Programming and Backward Induction . . . 18
  - 3.3.1 The Central Difficulty . . . 18
  - 3.3.2 The Dynamic Programming Principle . . . 19
- 3.4 The Longstaff–Schwartz Algorithm . . . 19
  - 3.4.1 Regression-Based Continuation Values . . . 19
  - 3.4.2 Algorithm and Implementation . . . 20
  - 3.4.3 Convergence Properties . . . 21
- 3.5 Dual Upper Bounds . . . 22
  - 3.5.1 The Dual Representation . . . 22

3.5.2 The Andersen–Broadie Algorithm . . . 23
3.5.3 Confidence Intervals . . . 24
3.6 Summary . . . 24

4 The Brownian Bridge **25**
4.1 Introduction . . . 25
4.2 Definition and Distributional Properties . . . 25
4.3 Simulation Algorithms . . . 26
4.4 Path Coupling and MLMC Connection . . . 27
4.5 Graphical Illustrations . . . 28
4.6 Summary . . . 30

5 The Multilevel Monte Carlo Method **31**
5.1 Introduction . . . 31
5.2 The Telescoping Identity . . . 31
5.3 The MLMC Estimator . . . 32
5.4 Optimal Sample Allocation . . . 33
5.5 Complexity Analysis . . . 33
5.6 Adaptive Implementation . . . 34
5.7 Application to Option Pricing . . . 34
5.8 Numerical Illustration . . . 35
5.9 Summary . . . 37

6 Numerical Experiments and Results **38**
6.1 Overview . . . 38
6.2 European Call: Standard MC vs MLMC . . . 38
6.2.1 Setup . . . 38
6.2.2 Results . . . 39
6.2.3 Discussion . . . 39
6.3 Asian Call: Standard MC vs MLMC . . . 42
6.3.1 Setup . . . 42
6.3.2 Results . . . 42
6.3.3 Discussion . . . 43
6.4 Brownian Bridge Variance Reduction . . . 46
6.4.1 Setup . . . 46
6.4.2 Results . . . 47
6.4.3 Discussion . . . 48
6.5 American Put: Longstaff–Schwartz Pricing . . . 49
6.5.1 Setup . . . 49
6.5.2 Results . . . 49
6.5.3 Discussion . . . 50
6.6 LSMC with MLMC for American Options . . . 52
6.6.1 Setup . . . 52

6.6.2 Results 53
6.6.3 Discussion 55
6.7 Remark on the Dual Upper Bound 56
6.8 Summary and Comparison 56

7 Conclusion **59**
7.1 Summary of Contributions 59
7.2 Key Findings 60
7.3 Limitations and Future Directions 60
7.4 Closing Remarks 61

Chapter 1

Introduction

1.1 Motivation and Background

The evolution of financial markets over the last few decades has led to an unprecedented increase in the complexity of financial instruments. Derivatives, which are once limited to simple vanilla contracts, now encompass a vast array of "exotic" products whose payoffs depend on intricate path-dependent triggers or multi-asset correlations. The valuation of these instruments is a fundamental task for risk management, regulatory compliance, and speculative trading.

At the heart of derivative pricing lies the Fundamental Theorem of Asset Pricing, which suggests that the price of an option is the discounted expected value of its future payoff under a risk-neutral measure. While the Black Scholes Merton framework provides a closed-form solution for simple European options, it falls short when confronted with real-world complexities such as stochastic volatility, jumps, or American-style early exercise features. In these cases, we turn to numerical methods.

1.2 The Computational Challenge

Traditionally, two main pillars have supported numerical pricing: Partial Differential Equation (PDE) methods and Monte Carlo (MC) simulations. While PDE methods are efficient for low-dimensional problems, they suffer from the "curse of dimensionality" as the number of underlying assets increases.

Monte Carlo methods, by contrast, offer a convergence rate of $\mathcal{O}(N^{-1/2})$ that is independent of the number of dimensions. However, this rate is frustratingly slow. To reduce the estimation error by a factor of 10, one must increase the number of paths by a factor of 100. Furthermore, when dealing with path-dependent options, one must discretize the time dimension (e.g., using the Euler–Maruyama scheme), introducing a discretization error that competes with the statistical sampling error. The computational burden of achieving a high-precision estimate often becomes a bottleneck in time-sensitive financial environments.

1.3 The Multilevel Monte Carlo Framework

This study focuses on a transformative advancement in simulation technology: the Multilevel Monte Carlo (MLMC) method. First introduced by Giles [9], and extensively detailed in the financial context by Korn et al. [15] and Uğur [20], MLMC revolutionizes the standard approach by performing simulations across a hierarchy of discretization levels.

The core intuition is to perform most of the simulations on "coarse" levels (which are computationally cheap but less accurate) and only a few simulations on "fine" levels (which are expensive but accurate). By cleverly combining these levels, MLMC can achieve the accuracy of a fine-grid simulation at a total cost that is significantly lower — often reducing the complexity from $\mathcal{O}(\epsilon^{-3})$ to $\mathcal{O}(\epsilon^{-2})$ for a given error tolerance ϵ [9].

Beyond financial applications, MLMC has been successfully applied to a wide range of problems including elliptic partial differential equations with random coefficients [7, 21], stochastic biological systems such as biochemical reaction networks [16], atmospheric dispersion modelling via stochastic Lagrangian particle methods [14], nested expectation problems such as the estimation of the expected value of sample information [12], and risk estimation in large financial portfolios [11], demonstrating the generality and practical relevance of the multilevel framework across scientific disciplines. Further extensions combining MLMC with sparse grids and Quasi-Monte Carlo methods have also been explored [10], with the potential to reduce computational complexity below $\mathcal{O}(\epsilon^{-2})$.

1.4 Scope and Objectives

Despite its success in European and Asian option pricing, the application of MLMC to American options presents a unique challenge: the "early exercise" problem. Determining the optimal time to exercise an option requires backward induction or regression-based approaches (like the Longstaff–Schwartz method), which do not always integrate seamlessly with the multilevel framework.

Furthermore, the construction of the underlying stochastic paths is critical. This study explores the use of Brownian Bridge techniques to enhance path generation. Unlike the standard random walk, the Brownian Bridge allows for "filling in" the path between two known points, which is particularly useful for multilevel refinement and for pricing barrier or lookback options where the maximum or minimum value during an interval is crucial.

1.5 Contributions and Outline

The objective of this research is to evaluate the synergy between MLMC and Brownian Bridge techniques in the context of complex option pricing. Specifically, we aim to:

1. Formulate a rigorous mathematical framework for MLMC as applied to finance.
2. Address the difficulties of the early exercise problem in American options within a multilevel context.

3. Provide numerical evidence that MLMC, supported by Brownian Bridge path construction, outperforms standard Monte Carlo methods in both speed and variance reduction.

The remainder of this study is organised as follows. Chapter 2 introduces the Monte Carlo method and its theoretical foundations, covering the Strong Law of Large Numbers as the basis for simulation-based estimation and a comprehensive treatment of variance reduction techniques including antithetic variates, control variates, importance sampling, and stratified sampling, with numerical results for European and Asian option pricing. Chapter 3 addresses the early exercise problem that distinguishes American options from their European counterparts, developing the optimal stopping framework, the dynamic programming principle, and the Longstaff–Schwartz regression-based algorithm as the primary Monte Carlo pricing method for American and Bermudan options. Chapter 4 introduces the Brownian bridge as a path construction tool, deriving its key distributional properties and presenting both forward and backward simulation algorithms, with particular emphasis on the conditional interpolation formula that underpins path coupling in the multilevel framework. Chapter 5 develops the Multilevel Monte Carlo method from the telescoping identity, establishing the optimal sample allocation across levels, the complexity theorem, and the adaptive implementation strategy. Chapter 6 presents five numerical experiments covering European call, Asian call, and American put options. The experiments compare Standard Monte Carlo and MLMC across a range of accuracy targets, examine the variance reduction properties of the Brownian bridge construction, price an American put option via the Longstaff–Schwartz algorithm, and investigate the numerical behaviour of the LSMC and MLMC combination. Chapter 7 concludes the study with a summary of contributions, key findings, a discussion of limitations, and directions for future research.

Chapter 2

Monte Carlo Methods for Option Pricing

2.1 Introduction

The Monte Carlo method represents a class of computational algorithms that rely on repeated random sampling to obtain numerical results [5, 15]. In financial mathematics, it serves as a primary tool for numerical integration, specifically for valuing contingent claims whose payoffs are path-dependent or high-dimensional [15, 19]. This chapter establishes the theoretical foundations of the method and explores techniques to enhance its efficiency through variance reduction.

2.2 Theoretical Foundations

The mathematical legitimacy of Monte Carlo simulations is anchored in the asymptotic behavior of random variables.

2.2.1 The Strong Law of Large Numbers

As defined in [15], the Strong Law of Large Numbers (SLLN) states that for a sequence $\{X_i\}_{i=1}^{\infty}$ of independent and identically distributed (i.i.d.) random variables with a finite mean μ, the sample average $\bar{X}_n$ converges to the expected value almost surely:

$$\mathbb{P}\left(\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}X_i=\mu\right)=1 \tag{2.1}$$

In the context of option pricing, this result has a direct and important interpretation. Let $\{X_i\}_{i=1}^{\infty}$ represent a sequence of independent and identically distributed discounted payoffs of an option, each simulated under the risk-neutral probability measure $\mathbb{Q}$. That

is, for a European-style option with payoff $h(S_T)$ at maturity T, each X_i takes the form

$$X_i = e^{-rT} h(S_T^{(i)}), \tag{2.2}$$

where $S_T^{(i)}$ denotes the i-th simulated terminal stock price under $\mathbb{Q}$ and r is the risk-free rate. By the Fundamental Theorem of Asset Pricing, the fair price of the option is the expected value of this discounted payoff under $\mathbb{Q}$:

$$V = \mathbb{E}^{\mathbb{Q}}\big[e^{-rT} h(S_T)\big] = \mu. \tag{2.3}$$

The SLLN then guarantees that the Monte Carlo estimator

$$\hat{V}_n = \frac{1}{n} \sum_{i=1}^{n} X_i \tag{2.4}$$

converges to the true option price V almost surely as $n \to \infty$. This principle is the mathematical foundation that legitimises the use of simulation for option pricing: by increasing the number of simulated paths, our numerical estimate will eventually approach the true theoretical price of the option under the risk-neutral measure.

2.2.2 The Crude Monte Carlo Method

The crude Monte Carlo approach is the most straightforward application of the SLLN [15, 20]. For a European option with a discounted payoff H, the estimator $\hat{V}$ is simply the arithmetic mean of N simulated payoffs [15]. In our numerical experiment, we priced a standard European call option using $N = 10,000$ simulations. The result yielded a price of **10.3606** with a standard error of **0.1445**.

2.2.3 First Numerical Results

To visualize the convergence properties of the crude method, Figure 2.1 illustrates the estimated price as a function of the number of simulations N.

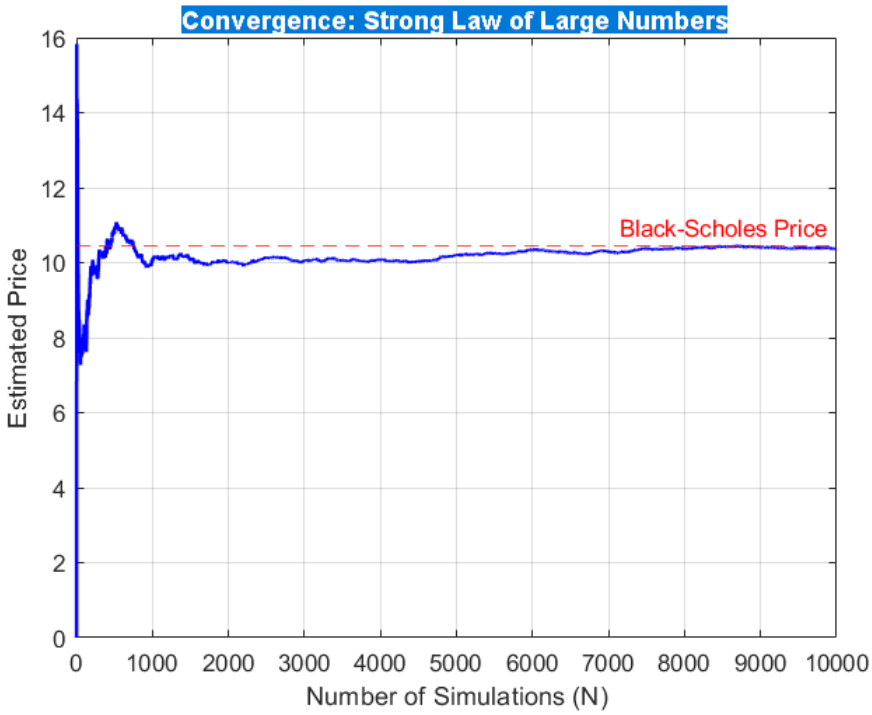

Figure 2.1: Convergence of the Monte Carlo estimator toward the Black-Scholes price (10.4506). Note the high volatility in the estimate at low values of N, which stabilizes as N approaches $10,000$.

2.3 Variance Reduction Techniques

While the crude method is robust, its convergence rate, $mathcalO(1/\sqrt{N})$ is relatively slow. To improve efficiency without simply increasing N, we utilize variance reduction methods.

2.3.1 Antithetic Variates

Antithetic variates is one of the simplest yet most effective variance reduction techniques. The core idea is to introduce a negative correlation between simulated paths. If we generate a random sample Z from a standard normal distribution $\mathcal{N}(0,1)$, its negative counterpart $-Z$ is also a valid sample from the same distribution.

As described in [5], let $\hat{V}_1$ be the estimate using Z and $\hat{V}_2$ be the estimate using $-Z$. The antithetic estimator is the average:

$$\hat{V}_{AV} = \frac{\hat{V}_1 + \hat{V}_2}{2} \tag{2.5}$$

The variance of this combined estimator is:

$$\mathrm{Var}(\hat{V}_{AV}) = \frac{1}{4}\left[\mathrm{Var}(\hat{V}_1) + \mathrm{Var}(\hat{V}_2) + 2\,\mathrm{Cov}(\hat{V}_1, \hat{V}_2)\right] \tag{2.6}$$

Since $\mathrm{Var}(\hat{V}_1) = \mathrm{Var}(\hat{V}_2)$, the equation simplifies to:

$$\mathrm{Var}(\hat{V}_{AV}) = \frac{1}{2}\left[\mathrm{Var}(\hat{V}_1) + \mathrm{Cov}(\hat{V}_1, \hat{V}_2)\right] \tag{2.7}$$

If the covariance $\mathrm{Cov}(\hat{V}_1, \hat{V}_2)$ is negative, then the variance of $\hat{V}_{AV}$ will be lower than that of the standard Monte Carlo estimator. In option pricing, since the payoff function is monotonic with respect to the underlying source of randomness, this technique is guaranteed to reduce variance. However, it should be noted that in cases where the monotonicity assumption does not hold strictly, the antithetic variates method may fail to produce a negative covariance and could, in the worst case, introduce numerical disturbances into the estimation.

Numerical Results and Discussion

The Antithetic Variates (AV) technique is implemented to price a standard European Call option with the following parameters: $S_0 = 100, K = 100, r = 0.05, \sigma = 0.20, T = 1$, and $N = 10^5$ simulations. The numerical results are summarized in Table 2.1.

Table 2.1: Numerical Performance of Antithetic Variates vs. Standard Monte Carlo

Method	**Estimated Price**	**Standard Error**	**VRF**
Black-Scholes (Analytical)	10.4506	-	-
Standard Monte Carlo	10.4225	0.0464	1.000
Antithetic Variates	10.4360	0.0232	4.0028

The empirical results demonstrate a significant improvement in estimation precision. The Standard Error is reduced from 0.0464 to 0.0232, which corresponds to a Variance Reduction Factor (VRF) of approximately 4.00. This indicates that the Antithetic Variates method is four times more efficient than the standard approach for this specific payoff.

As noted in [5], the Efficiency Gain of 75.02% implies that we can achieve the same level of accuracy as the standard Monte Carlo method using only one-fourth of the simulated paths. This reduction is attributed to the negative correlation between the pairs $(Z, -Z)$, which effectively cancels out much of the sampling noise inherent in the geometric Brownian motion paths.

2.3.2 Control Variates

The Control Variates (CV) technique is widely regarded as one of the most effective variance reduction methods in computational finance. It leverages the known analytical value of a

correlated variable to reduce the uncertainty in the estimation of an unknown quantity. As discussed both [5] and [19], if we aim to estimate $\alpha = \mathbb{E}[X]$, and we have an auxiliary random variable Y (the control variate) with a known expectation $\mathbb{E}[Y] = \mu_Y$, we can construct a new estimator $\hat{X}_{CV}$. The key requirement is that Y must be sampled from the same probability space as X — that is, both X and Y must be generated from the same simulated paths — so that the correlation between them can be exploited to reduce variance. In option pricing, a natural choice for Y is the underlying asset price S_T, whose expected value under the risk-neutral measure is known analytically as $\mathbb{E}[S_T] = S_0 e^{rT}$.

Mathematical Formulation

The control variate estimator is defined by the following linear adjustment:

$$X_{CV} = X + \beta(Y - \mu_Y) \tag{2.8}$$

where β is constant. Since $\mathbb{E}[Y - \mu_Y] = 0$, the expectation of the new estimator remains unbiased: $\mathbb{E}[X_{CV}] = \mathbb{E}[X] = \alpha$.

On the other hand, the variance of this adjusted estimator is given by:

$$\mathrm{Var}(X_{CV}) = \mathrm{Var}(X) + \beta^2 \mathrm{Var}(Y) + 2\beta \mathrm{Cov}(X, Y) \tag{2.9}$$

As noted in [19], the goal is to choose a β that minimizes this total variance. By taking the derivative with respect to β and setting it to zero, we obtain the *optimal* β^*:

$$\beta^* = -\frac{\mathrm{Cov}(X, Y)}{\mathrm{Var}(Y)} \tag{2.10}$$

Substituting β^* back into the variance equation, we derive the minimum achievable variance:

$$\mathrm{Var}(X^*_{CV}) = \mathrm{Var}(X)\left(1 - \rho^2_{XY}\right) \tag{2.11}$$

where ρ_{XY} is the correlation coefficient between the target variable X and the control variable Y.

Numerical Results and Discussion for Control Variates

The Control Variates (CV) technique is tested using the underlying asset price S_T as the control variable for a European Call option. The analytical expectation $\mathbb{E}[S_T] = S_0 e^{rT}$ is used to correct the Monte Carlo estimator. The results of the simulation with $N = 10^5$ paths are presented in Table 2.2.

The implementation yielded an optimal beta coefficient of $\beta^* = -0.6419$. The variance is reduced significantly, achieving a Variance Reduction Factor (VRF) of 6.9338. This performance is superior to the Antithetic Variates method previously discussed, as it utilizes the intrinsic correlation between the option's payoff and the underlying asset price.

As emphasized by Choe (2016), the effectiveness of this method depends on the linear correlation between the target payoff and the control variable. Since a call option's value

Table 2.2: Numerical Performance of Control Variates using Asset Price as Control

Method	**Estimated Price**	**Standard Error**	**VRF**
Black-Scholes (Analytical)	10.4506	-	-
Standard Monte Carlo	10.4321	0.0466	1.000
Control Variates (S_T)	10.4442	0.0177	6.9338

is highly sensitive to the final stock price, S_T serves as an excellent control variate. The reduction in the standard error from 0.0466 to 0.0177 confirms that incorporating known market information ($\mathbb{E}[S_T]$) into the simulation leads to a much more robust pricing model.

Control Variates for Asian Option Pricing

Asian options are path-dependent derivatives where the payoff depends on the average price of the underlying asset over a specific period. The arithmetic average, $A = \frac{1}{m}\sum_{j=1}^{m} S_{t_j}$, does not follow a log-normal distribution, which makes analytical pricing impossible.

In this section, we implement a Control Variate approach to price an Arithmetic Asian Call option. We use a standard European Call option, whose price is known via the Black-Scholes formula, as the control variable. Since the European payoff and the Asian payoff are driven by the same underlying stochastic process and are highly correlated, we expect a reduction in variance. To evaluate the robustness of the Control Variate technique, an Arithmetic Asian Call option is priced using $N = 10^5$ simulations and $m = 50$ monitoring dates. A standard European Call option with the same strike and maturity is utilized as the control variate. The numerical results are presented in Table 2.3.

Table 2.3: Numerical Performance of Control Variates for Asian Option Pricing

Method	**Estimated Price**	**Standard Error**	**VRF**
Standard Asian MC	5.8461	0.0257	1.0000
Control Variate (European)	5.8638	0.0137	3.5194

The simulation results confirm that even for complex path-dependent derivatives, the Control Variate method provides a significant efficiency gain. The standard error is reduced by nearly half (from 0.0257 to 0.0137), resulting in a Variance Reduction Factor (VRF) of 3.5194.

As discussed by Choe (2016), the effectiveness of the European option as a control variate for an Asian option stems from the high degree of correlation between the arithmetic average of the stock price and its final value. Since both payoffs are driven by the same underlying Wiener process, the fluctuations in the European payoff effectively "explain" a large portion of the variance in the Asian payoff. This allows the estimator to correct for sampling noise, leading to a much more stable and precise price estimate without increasing the number of time steps or simulation paths.

2.3.3 Importance Sampling

Importance Sampling (IS) is a powerful variance reduction technique used primarily when the payoff is determined by rare events. In a standard Monte Carlo simulation, if an option is deep out-of-the-money, most simulated paths will result in a zero payoff, leading to high relative variance. As discussed in [5], IS addresses this by changing the probability distribution of the underlying paths to favor "important" regions where the payoff is non-zero.

Change of Measure and Likelihood Ratio

Suppose we want to estimate $I = \mathbb{E}_f[h(X)] = \int h(x)f(x)dx$. We can rewrite this integral using a new probability density function $g(x)$ (where $g(x) > 0$ whenever $h(x)f(x) \neq 0$):

$$I = \int h(x)\frac{f(x)}{g(x)}g(x)dx = \mathbb{E}_g\left[h(X)\frac{f(X)}{g(X)}\right] \tag{2.12}$$

The term $L(X) = f(X)/g(X)$ is known as the **Likelihood Ratio**, also called the **Radon-Nikodym derivative**. The importance sampling estimator is then:

$$\hat{I}_{IS} = \frac{1}{n}\sum_{i=1}^{n} h(X_i)L(X_i) \tag{2.13}$$

where X_i are sampled from the distribution with density g. In the context of option pricing, we typically shift the mean (drift) of the underlying asset's process so that the simulated stock prices S_T end up closer to the strike price K.

According to Choe (2016), the choice of g is critical; an ideal g would be proportional to $|h(x)|f(x)$, which would theoretically reduce the variance to zero, though this is practically difficult to achieve.

Numerical Results and Discussion for Importance Sampling

To demonstrate the efficiency of Importance Sampling (IS), we priced a deep Out-of-the-Money (OTM) European Call option with a strike price of $K = 150$ and an initial stock price of $S_0 = 100$. Under standard conditions, such an option has a very low probability of exercise, causing most Monte Carlo paths to result in zero payoffs. The simulation results for $N = 10^5$ paths are summarized in Table 2.4.

Table 2.4: Numerical Performance of Importance Sampling for Deep OTM Options

Method	**Estimated Price**	**Standard Error**	**VRF**
Black-Scholes (Analytical)	0.3596	—	—
Standard Monte Carlo	0.3605	0.0091	1.0000
Importance Sampling	0.3598	0.0013	49.5833

As shown in Table 2.4, the Importance Sampling method achieved a Variance Reduction Factor of approximately 49.58. This efficiency gain is due to the mean-shifting technique applied to the log-normal dynamics of the underlying asset. Specifically, under the standard risk-neutral measure, the log-return of the stock follows

$$\ln\left(\frac{S_T}{S_0}\right) \sim \mathcal{N}\left(\left(r - \tfrac{1}{2}\sigma^2\right)T,\ \sigma^2 T\right). \tag{2.14}$$

The importance sampling distribution g is obtained by shifting the mean of this log-normal distribution toward the strike price K. Concretely, we replace the original drift $\mu_0 = (r - \frac{1}{2}\sigma^2)T$ with the shifted drift

$$\mu^* = \frac{\ln(K/S_0)}{\sigma\sqrt{T}} \cdot \sigma\sqrt{T} - \tfrac{1}{2}\sigma^2 T = \ln(K/S_0) - \tfrac{1}{2}\sigma^2 T, \tag{2.15}$$

so that the simulated terminal stock prices are centred around the strike $K = 150$ rather than around $S_0 e^{rT} \approx 105$. The likelihood ratio $L(X) = f(X)/g(X)$ then corrects for this change of measure, ensuring the estimator remains unbiased. This forces the simulated paths into the "important" region of the payoff distribution where $S_T > K$, which is rarely visited under the original measure.

As discussed in [5], for rare-event simulation, Importance Sampling is far superior to brute-force Monte Carlo. By adjusting the drift of the stochastic process and correcting the bias with the likelihood ratio (Radon-Nikodym derivative), we obtained a high-precision estimate with a standard error nearly seven times smaller than that of the standard approach, while remaining within 0.0002 of the analytical Black-Scholes price.

2.3.4 Stratified Sampling

Stratified Sampling is a technique designed to ensure that the entire range of the input random variables is covered more uniformly than in standard Monte Carlo. In a typical pseudo-random generation, samples can cluster together by chance, leaving significant gaps in the probability space.

As discussed in [5], we partition the domain of the random variable into n disjoint strata (sub-intervals) of equal probability. Let U be a uniform random variable on $[0, 1]$. For a sample size n, the stratified samples U_i are generated as:

$$U_i = \frac{(i-1) + V_i}{n}, \quad i = 1, \ldots, n \tag{2.16}$$

where V_i are independent uniform random variables on $[0, 1]$. These U_i values are then transformed into normal variables using the inverse cumulative distribution function, $Z_i = \Phi^{-1}(U_i)$.

The primary advantage of Stratified Sampling is that it reduces the variance by eliminating the "clumping" of random samples. Choe (2016) notes that this method is particularly effective for one-dimensional problems or when a specific dimension (such as time or the final stock price) dominates the variance of the payoff.

Numerical Results and Discussion for Stratified Sampling

Stratified Sampling is implemented by partitioning the $[0, 1]$ uniform domain into $N = 10^5$ equal strata. The results, compared with a standard Monte Carlo simulation, are summarized in Table 2.5.

Table 2.5: Numerical Performance of Stratified Sampling

Method	**Estimated Price**	**Standard Error**	**VRF**
Black-Scholes (Analytical)	10.4506	—	—
Standard Monte Carlo	10.4214	0.0465	1.0000
Stratified Sampling	10.4506	0.0465	0.9968

The numerical experiment yielded a Variance Reduction Factor (VRF) near unity, suggesting no significant efficiency gain over the standard approach for this specific case. Interestingly, the Stratified Sampling estimate of 10.4506 coincides exactly with the Black-Scholes analytical price, although this agreement is partly coincidental given the VRF of 0.9968 and the standard error of 0.0465. As noted in [5], while Stratified Sampling guarantees a more uniform coverage of the probability space, its effectiveness in reducing variance is highly dependent on the monotonicity and linearity of the payoff function.

In the case of a European Call option, the non-linear transformation through the normal inverse CDF and the exponential growth of the asset price can diminish the benefits of stratification in a single dimension. However, this method remains a foundational step toward more advanced quasi-Monte Carlo techniques and Latin Hypercube Sampling, which address high-dimensional variance more effectively.

These results emphasize that while basic Monte Carlo is reliable, the integration of variance reduction techniques is essential for the high-precision requirements of financial modeling. This serves as the motivation for the **Multilevel Monte Carlo** approach discussed later in this study.

Chapter 3

American Option Pricing via Monte Carlo

3.1 Introduction

The variance reduction techniques presented in the previous chapter significantly improve the efficiency of Monte Carlo pricing for European-style options. However, a large and practically important class of options — American options — cannot be priced by a simple extension of those methods. What makes American options fundamentally different is that the holder may exercise the contract at any point during its life, not only at expiry. This right, known as the **early exercise privilege**, transforms the pricing problem from computing a fixed expected value into solving an **optimal stopping problem**: at every moment, the holder must decide whether it is better to exercise immediately or to continue holding.

This chapter builds the mathematical and computational framework needed to handle this challenge. We begin by formally defining American and Bermudan contingent claims and characterizing their fair price as a supremum over all admissible stopping times. We then show why a naive Monte Carlo approach fails and introduce the **dynamic programming principle** as the natural solution strategy. The heart of the chapter is the **Longstaff–Schwartz algorithm** [17], which approximates the intractable continuation values by linear regression on simulated paths, producing a lower bound on the true price. Finally, since a lower bound alone is not sufficient to certify accuracy, we present the **primal-dual algorithm** of [1], which computes a complementary upper bound and allows us to construct a confidence interval for the true option price.

3.2 American and Bermudan Contingent Claims

3.2.1 The Optimal Stopping Problem

For a European option, the holder has no decisions to make: the option either expires in the money or it does not, and pricing reduces to a single discounted expectation. For an American option, the holder can exercise at any time $\tau \in [0, T]$, and will rationally choose τ to maximise the expected discounted payoff. Since the seller must be prepared against any such strategy, the fair price of the American option is

$$\hat{p} = \sup_{\tau \in \mathcal{S}[0,T]} \mathbb{E}\big(e^{-r\tau} B_\tau\big)\,, \tag{3.1}$$

where B_τ denotes the payoff received upon exercise at τ and $\mathcal{S}[0, T]$ is the set of all stopping times adapted to the market filtration with values in $[0, T]$ almost surely [15].

More formally, an **American contingent claim** is a progressively measurable stochastic process $B = \{(B_t, \mathcal{F}_t)\}_{t\in[0,T]}$ with $B_t \geq 0$, continuous sample paths, and satisfying the integrability condition

$$\mathbb{E}\left(\sup_{0\leq s\leq T} (B_s)^\mu\right) < \infty \quad \text{for some } \mu > 1. \tag{3.2}$$

For the canonical American put, the payoff process is simply $B_t = (K - S(t))^+$ at each time t, where K is the strike price and $S(t)$ is the current stock price at time t. The integrability condition (3.2) is a mild regularity requirement that is satisfied in all standard models.

The appropriate hedging concept for American options is a self-financing portfolio strategy whose wealth process dominates the payoff process at all times, so that the seller can always meet the holder's exercise demand regardless of when it arrives. The **fair price** is then defined as the infimum initial wealth for which such a hedging strategy exists [15]. A fundamental result, proven in [13] and stated also in [15], confirms that this fair price coincides exactly with the optimal stopping value:

$$\hat{p} = \sup_{\tau \in \mathcal{S}[0,T]} \mathbb{E}\big(e^{-r\tau} B_\tau\big)\,, \tag{3.3}$$

and furthermore, an optimal stopping time τ^* exists at which the supremum is attained. The economic interpretation is clean: the holder should exercise at the first moment when the immediate payoff equals the value of the option, i.e. when waiting yields no additional benefit. Unfortunately, neither τ^* nor the value process have closed-form representations even for the simple American put, which is why numerical methods are essential [15].

3.2.2 Bermudan Approximations

Working directly with the continuous-time optimal stopping problem in (3.1) is difficult in a simulation framework. A practical and well-justified simplification is to restrict the exercise

opportunities to a finite discrete grid of dates $0 = t_0 < t_1 < t_2 < \cdots < t_m = T$. Options with this structure are called **Bermudan options** — a name reflecting their intermediate nature between European (one exercise date) and American (continuous exercise).

Formally, a Bermudan option consists of $\mathcal{F}_{t_i}$-measurable payoff variables $B_{t_i} \geq 0$ and the right to receive payment B_τ at an exercise time $\tau \in \{t_1, \ldots, t_m\}$ chosen by the holder [15]. Its fair price is

$$\hat{p} = \sup_{\tau \in \mathcal{S}\{t_1,\ldots,t_m\}} \mathbb{E}\big(e^{-r\tau} B_\tau\big)\,, \tag{3.4}$$

where $\mathcal{S}\{t_1, \ldots, t_m\}$ now denotes the set of stopping times with values only in the discrete exercise grid, that is, $\mathcal{S}\{t_1, \ldots, t_m\} = \{t_1, \ldots, t_m\}$. The discretization is justified by the convergence result presented in [2], which shows that the gap between the American price and the Bermudan price is of order $1/\sqrt{m}$. In practice, a grid of 50 to 250 equally spaced dates is sufficient to approximate the American price very accurately for a moderate value of maturity (life-time) T [15], and the Bermudan structure is far more tractable computationally.

3.3 Dynamic Programming and Backward Induction

3.3.1 The Central Difficulty

For a European option, the Monte Carlo approach is straightforward: simulate many paths of the underlying asset, evaluate the payoff at expiry on each path, and average the discounted results. This works because the payoff is fully determined by the path, with no decisions required along the way.

For a Bermudan option, the situation is fundamentally different. As emphasised in [15], we **cannot** simply generate paths and evaluate payoffs, because we also need to know at each exercise date whether it is optimal to exercise or to continue holding. This decision requires knowledge of the **continuation value** — the expected discounted payoff from holding the option going forward — which is itself a conditional expectation that depends on the entire future evolution of the stock price. In other words, to price the option we need the optimal exercise strategy, but to find that strategy we need the option values at each point in time, which is what we are trying to compute in the first place. The resolution is to break the problem into two steps [15]:

1. Determine (an approximation to) the optimal exercise strategy τ^*.
2. Estimate the option price $\mathbb{E}(e^{-r\tau^*} B_{\tau^*})$ by Monte Carlo under this strategy.

Step 2 is a standard Monte Carlo calculation once the optimal exercise strategy is fixed. The challenge lies entirely in Step 1, and the natural tool for addressing it is dynamic programming.

3.3.2 The Dynamic Programming Principle

Dynamic programming resolves the problem by working backwards from the expiry date T, determining the locally optimal exercise decision at each date given everything that will happen after it. Following [15], let $\mathcal{S}(i)$ denote the set of stopping times taking values in $\{i, i+1, \ldots, m\}$, and define $V(i) = e^{-rt_i} B_{t_i}$ as the discounted immediate payoff at time t_i.

At the final date $t_m = T$, there is no decision to make: if the option has not been exercised, the holder receives B_T. Moving one step backwards to t_{m-1}, the holder compares the immediate payoff $B_{t_{m-1}}$ against the **continuation value**, i.e. the expected discounted payoff from holding until T:

$$\mathbb{E}\big(e^{-r(T-t_{m-1})} B_T \;\big|\; S(t_{m-1})\big)\,. \tag{3.5}$$

The rational decision is to exercise if and only if the immediate payoff exceeds the continuation value. Repeating this comparison at each successive date, stepping backwards from T to t_1, yields the complete optimal strategy:

$$\tau(i) = \begin{cases} i, & \text{if } V(i) \geq \mathbb{E}(V(\tau(i+1)) \mid S(t_i)), \\ \tau(i+1), & \text{otherwise.} \end{cases} \tag{3.6}$$

At t_0, the resulting stopping time $\tau(0)$ is the optimal exercise strategy and the fair price is $\mathbb{E}(V(\tau(0)))$ [15].

This algorithm is conceptually complete, but the conditional expectations $\mathbb{E}(V(\tau(i+1)) \mid S(t_i))$ cannot be computed analytically in general. This is precisely the gap that the Longstaff–Schwartz algorithm fills.

3.4 The Longstaff–Schwartz Algorithm

3.4.1 Regression-Based Continuation Values

The key idea of [17] is to replace the intractable conditional continuation values in (3.6) with statistical estimates obtained by linear regression on the simulated paths. The rationale is as follows. Since $S(t)$ is a Markov process, or at least presumed in many financial models, the payoff at any time can be written as a function of the current stock price: $B_t = f(S(t))$. It follows that the continuation value at time t_i is also a function of $S(t_i)$ alone:

$$C(S(t_i); i) := \mathbb{E}(V(\tau(i+1)) \mid S(t_i)) = u(S(t_i)), \tag{3.7}$$

for some unknown function u. The Longstaff–Schwartz approach approximates u by a linear combination of chosen basis functions $H_1, \ldots, H_k$:

$$u(S(t_i)) \approx \sum_{l=1}^{k} a_l \, H_l(S(t_i)), \tag{3.8}$$

and estimates the coefficients $a_1, \ldots, a_k$ by least-squares regression on the N simulated paths. Specifically, at each time step i, the coefficients are found by minimising the sample mean squared error:

$$\min_{a \in \mathbb{R}^k} \frac{1}{N} \sum_{j=1}^{N} \left(g\big(i+1, S^{(j)}(t_{i+1})\big) - \sum_{l=1}^{k} a_l H_l\big(S^{(j)}(t_i)\big) \right)^2, \tag{3.9}$$

where $g(i+1, S^{(j)}(t_{i+1}))$ is the discounted future value along path j, used as a proxy for the true continuation value. This is a standard ordinary least-squares problem. Defining the design matrix row and matrix as

$$H(i,j) = \big(H_1\big(S^{(j)}(t_i)\big), \ldots, H_k\big(S^{(j)}(t_i)\big)\big), \tag{3.10}$$

$$H(i) = \big(H(i,1)^\top, \ldots, H(i,N)^\top\big)^\top, \tag{3.11}$$

the solution is obtained via the pseudoinverse $H^+(i) = \big(H(i)^\top H(i)\big)^{-1} H(i)^\top$, giving the optimal coefficient vector

$$a^*(i) = H^+(i)\, g(i+1), \tag{3.12}$$

where $H^+(i)$ is the pseudoinverse of $H(i)$, which may be obtained by Singular Value Decomposition (SVD). The resulting estimate of the continuation value at time t_i is then

$$\hat{C}(S;i) = \sum_{l=1}^{k} a_l^*(i)\, H_l(S). \tag{3.13}$$

A practical refinement noted in [15] is that the regression at each time step should be performed only on paths that are currently **in the money**, i.e., where $f(S^{(j)}(t_i)) > 0$. Out-of-the-money paths will not be exercised at t_i regardless of the continuation value, so including them in the regression only adds unnecessary noise to the estimates.

3.4.2 Algorithm and Implementation

Putting together the backward induction framework with the regression-based approximation of continuation values, the complete Longstaff–Schwartz algorithm proceeds as follows [15, 17].

1. Choose a set of basis functions $H_1, \ldots, H_k$ for the regression.
2. Simulate N independent stock price paths $\{S^{(j)}(t_i)\}_{i=1}^{m}$ for $j = 1, \ldots, N$ at all exercise dates.
3. Initialise the terminal discounted payoffs:

$$\hat{V}(m,j) := e^{-rT} f\big(S^{(j)}(T)\big), \quad j = 1, \ldots, N. \tag{3.14}$$

4. Proceed backward in time. For each $i = m-1, m-2, \ldots, 1$:

   - Using only in-the-money paths, solve (3.9) to obtain the optimal weights $\hat{a}^*(i) = H^+(i)\,\hat{V}(i+1)$.
   - Compute the estimated continuation value for each path using (3.13).
   - Update the value of each path by comparing immediate exercise with continuation:

$$\hat{V}(i,j) := \begin{cases} e^{-rt_i} f\big(S^{(j)}(t_i)\big)\,, & \text{if } e^{-rt_i} f\big(S^{(j)}(t_i)\big) \geq \hat{C}\big(S^{(j)}(t_i); i\big)\,, \\ \hat{V}(i+1,j), & \text{otherwise.} \end{cases} \tag{3.15}$$

5. Simulate M new independent paths and apply the learned exercise strategy to these fresh paths to obtain the final price estimate:

$$\hat{V}(0) := \frac{1}{M} \sum_{j=N+1}^{N+M} \hat{V}(1,j). \tag{3.16}$$

The reason for using a separate batch of M paths in Step 5 is important. If the same paths used for regression are also used for pricing, the exercise strategy would be overfitted to those specific paths, inflating the price estimate. Using fresh paths avoids this bias [15].

3.4.3 Convergence Properties

The theoretical convergence of the Longstaff–Schwartz algorithm is analysed rigorously in [6], who identify two separate sources of error. The first is a **projection error**: the true continuation value function is approximated within a finite-dimensional subspace spanned by k basis functions. The second is a **statistical error**: the regression coefficients are estimated from N simulated paths rather than computed exactly. Letting $V^k(0)$ denote the option price obtained when the regression is solved exactly with k basis functions, the study in [6] proves that:

- As $k \to \infty$, with basis functions dense in an appropriate L^2 space: $V^k(0) \to V(0)$.
- For any fixed k, as $N \to \infty$: $V^{k,N}(0) \to V^k(0)$ $\mathbb{P}$-almost surely, at a rate of $\mathcal{O}(1/\sqrt{N})$.

An important practical implication is that increasing N alone does not recover the true option price — the bias from using a finite basis remains. Both k and N must be increased together to converge to the true value.

Regarding the choice of basis functions, Korn et al. [15] advise selecting functions whose shape resembles the payoff and the corresponding European pricing formula. For a standard American put, the three-function basis

$$H_1(s) = 1, \quad H_2(s) = (K-s)^+, \quad H_3(s) = \exp\big(\tfrac{1}{2}(s-K)^2\big) \tag{3.17}$$

has been found to perform very well in practice, providing a good balance between expressiveness and parsimony.

Finally, and crucially, the Longstaff–Schwartz algorithm produces a **lower bound** on the true option price. Because the learned exercise strategy is based on regression approximations rather than the true continuation values, it is suboptimal. Exercising suboptimally can only reduce the realised payoff relative to the true optimum [15, 17], so the estimated price systematically understates the true price. To assess how tight this lower bound is, we need a complementary upper bound, which is the subject of the next section.

3.5 Dual Upper Bounds

3.5.1 The Dual Representation

Knowing that the Longstaff–Schwartz estimate is a lower bound tells us the option is worth *at least* some value, but without an upper bound we cannot assess how far the estimate might be from the true price. The primal-dual approach of [18] and [1] addresses this by constructing an upper bound that can be computed alongside the lower bound, and together, providing a confidence interval that directly quantifies the quality of the exercise strategy. The key mathematical observation is that the discounted value function $\hat{V}(i)$ is a **super-martingale** — holding the option longer can only increase its value — and is in fact the minimal super-martingale dominating the discounted payoff process, known as the **Snell envelope** [15]. For any discrete-time martingale $M(i)$ starting at $M(0) = 0$ and any stopping time τ, the optional sampling theorem [15] gives

$$\mathbb{E}(g(\tau, S(t_\tau))) \leq \mathbb{E}\left(\max_{k=1,\dots,m} \{g(k, S(t_k)) - M(k)\} \right), \tag{3.18}$$

where $g(k, S(t_k)) = e^{-rt_k} \max(K - S(t_k), 0)$ denotes the discounted payoff of the option if exercised at time t_k. Taking the supremum over τ on the left and the infimum over all martingales M on the right gives the dual representation of the option price [15]:

$$V(0) \leq \inf_M \mathbb{E}\left(\max_{k=1,\dots,m} \{g(k, S(t_k)) - M(k)\} \right). \tag{3.19}$$

Equality holds when M is the Doob–Meyer martingale [1] of the value process, defined through its increments

$$\Delta(i) = \hat{V}(i) - \mathbb{E}\Big(\hat{V}(i) \mid S(t_{i-1})\Big), \quad \hat{M}(k) = \sum_{i=1}^{k} \Delta(i). \tag{3.20}$$

If the exercise strategy were truly optimal, $\hat{V}$ would itself be a martingale and each $\Delta(i)$ would then be zero. The size of these increments therefore serves as a diagnostic for how far the Longstaff–Schwartz strategy deviates from the optimum.

3.5.2 The Andersen–Broadie Algorithm

In practice the conditional expectations in (3.20) cannot be computed analytically, the study in [1] estimates them by a **nested simulation**: for each point along each primary path, a set of short sub-paths is launched and averaged. Using the stopping times inherited from the Longstaff–Schwartz algorithm, the value function along each primary path is

$$V(i) = \begin{cases} g(i, S(t_i)), & \text{if } g(i, S(t_i)) \geq \hat{C}(S(t_i); i), \\ \mathbb{E}(V(i+1) \mid S(t_i)), & \text{otherwise.} \end{cases} \tag{3.21}$$

The algorithm proceeds as follows [1, 15]. For each outer path $j = 1, \ldots, N_1$ (i.e. each independently simulated primary stock price trajectory along which the martingale increments are accumulated):

1. Simulate the primary stock price path $S^{(j)}(t_0), S^{(j)}(t_1), \ldots, S^{(j)}(t_m)$ with $M^{(j)}(0) = 0$.
2. For $i = 0, 1, \ldots, m-1$:
   - If $i > 0$, compute $\hat{V}(i, j)$ from (3.21).
   - Launch N_2 sub-paths from the current state $S^{(j)}(t_i)$ and estimate the conditional continuation value by averaging the discounted payoffs:

$$\mathbb{E}\Big(\hat{V}(i+1) \;\Big|\; S^{(j)}(t_i)\Big) \approx \frac{1}{N_2} \sum_{k=1}^{N_2} g\Big(\tau(i+1), \hat{S}^k\big(t_{\tau(i+1)}\big)\Big). \tag{3.22}$$

   - For $i > 0$, compute the martingale increment $\Delta^{(j)}(i)$ and update the running sum $M^{(j)}(i) = M^{(j)}(i-1) + \Delta^{(j)}(i)$.
3. At the terminal date set $\hat{V}(m, j) = g(m, S^{(j)}(T))$ and update $\Delta^{(j)}(m)$ and $M^{(j)}(m)$ accordingly.
4. Compute the upper bound contribution for this path:

$$Y^{up}(j) = \max_{i=1,\ldots,m} \big\{ g(i, S^{(j)}(t_i)) - M^{(j)}(i) \big\}. \tag{3.23}$$

The upper bound estimator is obtained by averaging over all outer paths:

$$\hat{Y}^{up}_{N_1} = \frac{1}{N_1} \sum_{j=1}^{N_1} Y^{up}(j). \tag{3.24}$$

3.5.3 Confidence Intervals

With both a lower bound L_N (from the Longstaff–Schwartz algorithm, applied to N fresh paths with sample standard deviation $\hat{\sigma}_L$) and an upper bound $\hat{Y}_{N_1}^{up}$ (from the Andersen–Broadie algorithm, with N_1 sample standard deviation $\hat{\sigma}_{Y^{up}}$), we can construct an approximate 95%-confidence interval for the true Bermudan option price:

$$\left[L_N - 1.96 \, \frac{\hat{\sigma}_L}{\sqrt{N}}, \quad \hat{Y}_{N_1}^{up} + 1.96 \, \frac{\hat{\sigma}_{Y^{up}}}{\sqrt{N_1}} \right]. \tag{3.25}$$

Since the lower bound is biased downward and the upper bound is biased upward, this interval is conservative: the true price lies inside it with probability at least 95%. The width of the interval directly measures how much value the suboptimal exercise strategy sacrifices. A natural point estimate is the midpoint $\left(L_N + \hat{Y}_{N_1}^{up} \right) / 2$.

The main computational cost of the upper bound is the nested simulation. As reported in [15], estimating the inner conditional expectations typically consumes between 60% and 95% of the total wall-clock time, with the total number of simulated path segments bounded by

$$N_{\max} = N_1 N_2 m. \tag{3.26}$$

This computational expense motivates the search for more efficient path generation methods, which is addressed through the Brownian bridge construction in Chapter 4 and the Multilevel Monte Carlo framework in Chapter 5.

3.6 Summary

This chapter has developed the complete framework for pricing American-style options by Monte Carlo simulation. The early exercise privilege transforms pricing into an optimal stopping problem, which is addressed by backward induction through the dynamic programming equations. The Longstaff–Schwartz algorithm makes this tractable by approximating the continuation values at each exercise date through linear regression on simulated paths, producing a lower bound for the option price that converges in both the number of basis functions and the number of paths. The Andersen–Broadie primal-dual algorithm then constructs a complementary upper bound through nested simulation, resulting in a confidence interval whose width directly reflects the quality of the estimated exercise strategy.

These results emphasise that pricing American options is significantly more computationally demanding than pricing European options, not only because of the backward induction itself, but also because producing a certified price estimate requires both bounds simultaneously. This additional cost is one of the key motivations for applying the Multilevel Monte Carlo framework, developed in Chapter 5, to the American option setting. Numerical illustrations of the Longstaff–Schwartz algorithm applied to an American put option, including the estimated early exercise boundary and price convergence results, are presented in Section 6.5.

Chapter 4

The Brownian Bridge

4.1 Introduction

In the previous chapters, all asset price paths are constructed by stepping forward in time from $t = 0$ to T, one increment at a time. This forward approach is intuitive, but it has a notable limitation: the terminal value is determined only at the very end of the simulation, which means we cannot apply variance reduction techniques to it before building the rest of the path.

The **Brownian bridge** offers a different perspective. Rather than constructing a path from left to right, it allows us to fix the endpoint first and then fill in the intermediate values conditionally. This seemingly simple change has two important consequences. First, it enables variance reduction methods to be applied directly to the terminal value before the full path is constructed. Second, and more importantly for this study, it provides the natural mechanism for coupling paths at different discretization levels in the Multilevel Monte Carlo method — the subject of the next chapter.

4.2 Definition and Distributional Properties

Let $\{W_t\}_{t\in[0,T]}$ be a one-dimensional Brownian motion and let $a, b \in \mathbb{R}$. As defined in [15], the process

$$B_t^{a,b} = a\frac{T-t}{T} + b\frac{t}{T} + \left(W_t - \frac{t}{T}W_T\right), \quad t \in [0,T], \tag{4.1}$$

is called a **Brownian bridge from** a **to** b. The structure of this formula is clear: the first two terms form a straight-line interpolation between a at $t = 0$ and b at $t = T$, while the third term adds Brownian fluctuations that are forced to vanish at both endpoints. One can verify directly that $B_0^{a,b} = a$ and $B_T^{a,b} = b$ almost surely.

The marginal distribution of the bridge at any intermediate time $t \in (0, T)$ is Gaussian. As shown in [15], a Brownian bridge from a to b satisfies

$$B_t^{a,b} \sim \mathcal{N}\left(a + \frac{t}{T}(b-a),\ t - \frac{t^2}{T}\right). \tag{4.2}$$

The mean traces the straight line between the two endpoints, while the variance $t(T-t)/T$ is symmetric and reaches its maximum at the midpoint $t = T/2$, vanishing at both ends as expected.

There is also a deeper probabilistic interpretation of the Brownian bridge. Using the joint distribution of (W_t, W_T) and the standard formula for conditional normals, one can show that [15]

$$W_t \mid W_T = b \sim \mathcal{N}\left(b\frac{t}{T},\ t - \frac{t^2}{T}\right), \tag{4.3}$$

which matches (4.2) with $a = 0$. In other words, a Brownian bridge from 0 to b is simply a **Brownian motion conditioned to arrive at b at time T**. This conditional interpretation is what makes the bridge so useful for path construction: it tells us precisely how to insert a new point into an existing path given its two neighbours.

Specifically, if we already know the values of a Brownian motion at times s and u (with $s < t < u$), the conditional distribution of the intermediate value W_t is [15]

$$W_t \mid (W_u = b,\ W_s = a) \sim \mathcal{N}\left(\frac{(u-t)\,a + (t-s)\,b}{u-s},\ \frac{(u-t)(t-s)}{u-s}\right). \tag{4.4}$$

The conditional mean is the linear interpolation between the two neighbouring values, and the conditional variance depends only on the distances to those neighbours. This formula is the core building block for backward simulation and, as we will see in Chapter 5, for the path coupling in MLMC.

4.3 Simulation Algorithms

There are two natural ways to simulate a Brownian bridge. The first, which Korn et al. [15] call forward simulation, simply generates a full Brownian motion path and then transforms it using (4.1). This is easy to implement but does not offer any advantage over standard simulation since the endpoint W_T is still determined last.

The more useful approach is **backward simulation**, which samples the most important random variables first. The idea is to generate W_T first, then insert the midpoint $W_{T/2}$, then $W_{T/4}$ and $W_{3T/4}$, and so on — each new point generated conditionally using (4.4) given its two already-known neighbours. This successive halving of the grid is known as the **dyadic partition** of $[0, T]$ [15].

At each level k of the dyadic refinement, the new points to be inserted sit exactly halfway between two existing neighbours. For this special case, equation (4.4) simplifies to [15]

$$W_s \mid (W_{t_{i+1}} = y,\ W_{t_i} = x) \sim \mathcal{N}\left(\frac{x+y}{2},\ \frac{T}{2^{k+1}}\right), \tag{4.5}$$

where $s = (2j-1)T/2^k$ is the new midpoint. The mean is simply the average of the two neighbours and the variance halves at each level. The complete backward simulation algorithm for $n = 2^K$ time points is as follows [15], where $K \in \mathbb{N}$ denotes the total number

of dyadic refinement levels, so that the finest grid contains 2^K equally spaced subintervals on $[0, T]$.

Algorithm 4.1: Backward Simulation of a Brownian Bridge from a to b with $n = 2^K$ Time Points

1. Simulate a standard normal random variable Z and set $W_T = \sqrt{T}\, Z$. Set $W_0 = 0$ and $h = T$.
2. For $k = 1$ to K:
   (a) Set $h = h/2$.
   (b) For $j = 1$ to 2^{k-1}, simulate a standard normal Z and set

$$W_{(2j-1)h} = \frac{1}{2}\big(W_{2(j-1)h} + W_{2jh}\big) + \sqrt{h}\, Z. \tag{4.6}$$

The order in which points are generated matters. The terminal value W_T, which has the highest marginal variance, is sampled first. The midpoint $W_{T/2}$, which has the highest conditional variance given the endpoints, is sampled second. At each subsequent level, the points with the highest remaining uncertainty are generated before those with lower variance. This ordering is particularly well suited to **quasi-random number sequences** such as Sobol or Halton sequences, where the first elements are the most uniformly distributed. By using the best quasi-random numbers for the most important random variables, backward simulation combined with quasi-Monte Carlo can achieve substantially better variance reduction than standard Monte Carlo [15].

4.4 Path Coupling and MLMC Connection

Beyond its use with quasi-random sequences, the backward simulation approach also enables a clean integration with the variance reduction methods from Chapter 2. Since W_T is sampled first, one can apply antithetic variates, stratified sampling, or importance sampling to W_T before constructing the rest of the path. For options whose payoff depends heavily on the terminal or extremal values of the underlying — European calls, Asian options, barrier options — this can lead to significant reductions in estimation variance [15].

The most important application of the Brownian bridge in this study, however, is its role in the **Multilevel Monte Carlo (MLMC)** method [9]. The MLMC approach, which is the subject of Chapter 5, estimates an expected payoff by combining simulations run at multiple levels of time discretization. At each correction level, a fine path and a coarse path must be generated from the *same* underlying Brownian motion so that their difference has small variance. The Brownian bridge interpolation formula (4.4) provides exactly the right tool for this: a coarse-level path uses a subset of the time points, and the additional fine-level points are inserted between them using the conditional distribution given the coarse values. This shared randomness is what makes the level correction terms small and the MLMC estimator efficient.

4.5 Graphical Illustrations

To illustrate the theoretical properties derived in the previous sections, Figure 4.1 shows eight simulated paths of a Brownian bridge from 0 to 0 on $[0, 1]$. All paths are pinned at both endpoints, with variance visibly peaking around the midpoint $t = T/2$, consistent with the theoretical variance $t(T - t)/T$ from equation (4.2).

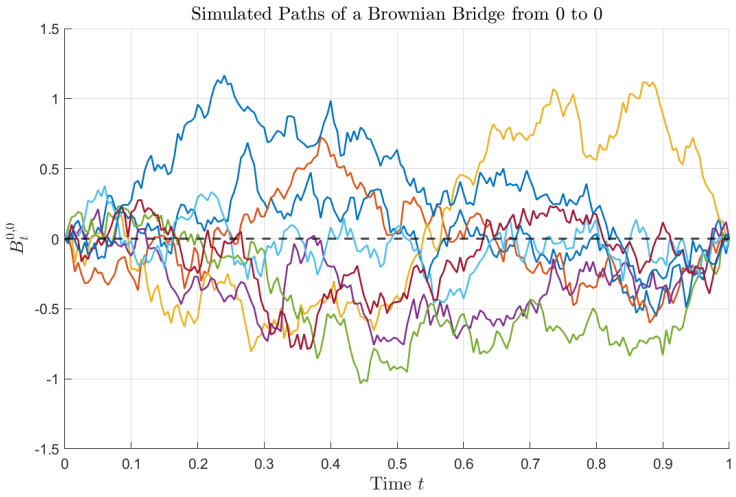

Figure 4.1: Eight simulated paths of a Brownian bridge from 0 to 0 on $[0, 1]$. Variance peaks at the midpoint $t = 0.5$ and vanishes at both endpoints.

Figure 4.2 shows eight paths of a bridge from $a = -0.5$ to $b = 1$. Every path starts at -0.5 and converges to 1 at $T = 1$, fluctuating around the linear interpolation shown by the dashed line, which represents the conditional mean $a + \frac{t}{T}(b - a)$ from equation (4.2).

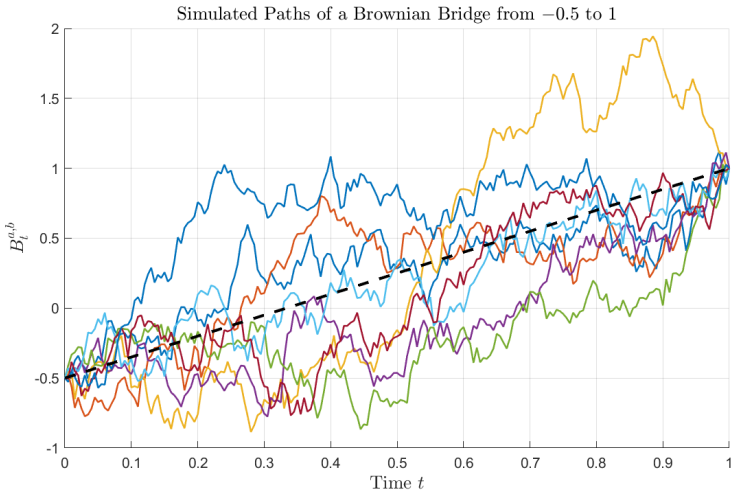

Figure 4.2: Eight simulated paths of a Brownian bridge from $a = -0.5$ to $b = 1$ on $[0, 1]$. The dashed line shows the conditional mean, i.e., the linear interpolation between the two endpoints.

Figure 4.3 illustrates the backward dyadic simulation algorithm (Algorithm 4.1) with $K = 4$ refinement levels, producing $2^4 = 16$ path points. The terminal value is sampled first and intermediate points are inserted progressively, with the dots marking the values generated at each refinement stage.

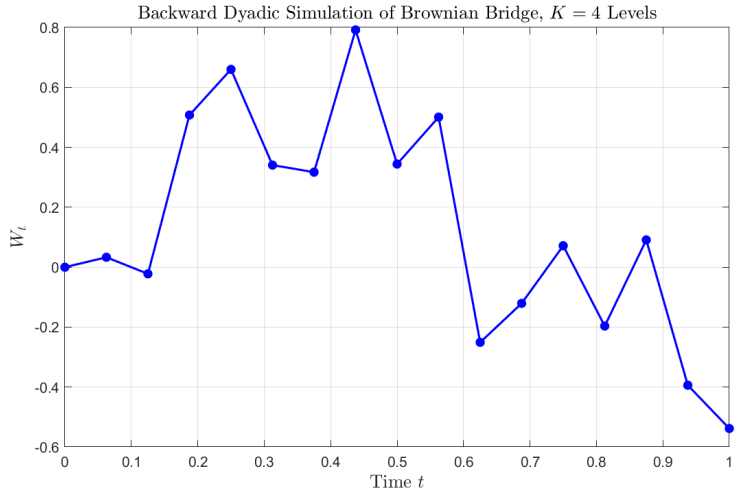

Figure 4.3: Backward dyadic simulation of a Brownian bridge with $K = 4$ refinement levels. The terminal value W_T is sampled first, followed by the midpoint, then the quarter points, and so on, consistent with Algorithm 4.1.

4.6 Summary

This chapter has introduced the Brownian bridge as a key path construction tool. A Brownian bridge from a to b is a Brownian motion conditioned to arrive at b at time T, and its marginal and conditional distributions are fully Gaussian. The backward simulation algorithm generates path values starting from the terminal point and progressively refining the grid by dyadic halving, using the conditional interpolation formula at each step. This ordering aligns naturally with quasi-random sequences and enables variance reduction methods to be applied to the terminal value before the full path is built. Most importantly, the conditional interpolation formula is the mechanism that couples fine and coarse paths in the MLMC framework, ensuring that level correction terms have small variance. The full development of MLMC and its application to option pricing is presented in the next chapter.

Chapter 5

The Multilevel Monte Carlo Method

5.1 Introduction

The previous chapters have progressively built up the tools needed to address the central computational challenge of this study. Chapter 2 showed that variance reduction techniques can significantly improve the efficiency of standard Monte Carlo, but they do not change its fundamental limitation: for path-dependent options, the time discretization error and the statistical sampling error must both be controlled simultaneously, and doing so at high precision is expensive. Chapter 4 introduced the Brownian bridge as a path construction tool that naturally couples simulations at different levels of refinement. This chapter brings these ideas together in the **Multilevel Monte Carlo (MLMC)** method, introduced by [9].

The key insight of MLMC is straightforward. Rather than running all simulations at a single fine discretization level which is accurate but expensive MLMC distributes the computational effort across a hierarchy of levels, from coarse to fine. Most of the work is done at the cheap coarse levels, and only a small number of simulations are run at the expensive fine levels. By carefully coupling the simulations at adjacent levels using the same underlying Brownian increments, the variance of the correction terms between levels is kept small, and the total cost of achieving a given accuracy may be dramatically reduced.

5.2 The Telescoping Identity

Consider a d-dimensional stochastic differential equation of the form

$$dX(t) = a(t, X(t))\,dt + \sigma(t, X(t))\,dW(t), \tag{5.1}$$

where the coefficients satisfy conditions ensuring the existence of a unique strong solution. Our goal is to compute $\mathbb{E}(g(X(T)))$ for a Lipschitz-continuous payoff $g : \mathbb{R}^d \to \mathbb{R}$, where $X(T)$ is approximated numerically by the Euler–Maruyama method [8] with stepsize δ. For a decreasing sequence of stepsizes $\delta_0 > \delta_1 > \cdots > \delta_L$, let $Y^{\delta_l}(T)$ denote the Euler–Maruyama approximation of $X(T)$ at level l. The finest-level expectation $\mathbb{E}(g(Y^{\delta_L}(T)))$

can be written as the following telescoping sum [15]:

$$\mathbb{E}\big(g\big(Y^{\delta_L}(T)\big)\big) = \mathbb{E}\big(g\big(Y^{\delta_0}(T)\big)\big) + \sum_{l=1}^{L} \mathbb{E}\big(g\big(Y^{\delta_l}(T)\big) - g\big(Y^{\delta_{l-1}}(T)\big)\big)\,. \tag{5.2}$$

This identity is exact and holds for any choice of levels. Its value lies in the fact that each correction term $g(Y^{\delta_l}) - g(Y^{\delta_{l-1}})$ measures the incremental improvement in accuracy as the stepsize is halved from level $l-1$ to level l. If the fine and coarse paths are driven by the *same* Brownian increments, this difference converges to zero as the stepsizes decrease, meaning its variance is small. It is therefore much cheaper to estimate accurately than the individual expectations themselves.

5.3 The MLMC Estimator

Each expectation in the telescoping sum (5.2) is estimated independently by a crude Monte Carlo estimator. At the coarsest level $l = 0$, we simulate N_0 independent paths at stepsize δ_0 and compute

$$I_0(g) = \frac{1}{N_0} \sum_{i=1}^{N_0} g\big(Y_i^{\delta_0}(T)\big)\,. \tag{5.3}$$

At each correction level $l = 1, \ldots, L$, we simulate N_l independent *pairs* of paths — one fine path at stepsize δ_l and one coarse path at stepsize δ_{l-1} — both driven by the same Brownian increments, and compute

$$I_l(g) = \frac{1}{N_l} \sum_{i=1}^{N_l} \Big(g\Big(Y_i^{\delta_l}(T)\Big) - g\Big(Y_i^{\delta_{l-1}}(T)\Big)\Big)\,. \tag{5.4}$$

The **multilevel Monte Carlo estimator** is then the sum of all level estimators [9, 15]:

$$I(g) = \sum_{l=0}^{L} I_l(g). \tag{5.5}$$

The requirement that the fine and coarse paths in (5.4) share the same Brownian increments is essential. This is where the Brownian bridge construction from Chapter 4 plays its role: the coarse path uses a subset of the time grid, and the additional fine-level points are inserted between the coarse points using the conditional interpolation formula (4.4). The shared randomness ensures that the difference $g(Y^{\delta_l}) - g(Y^{\delta_{l-1}})$ is small, which is what makes each correction level cheap to estimate accurately.

5.4 Optimal Sample Allocation

Since all level estimators are based on independent samples, the variance of the MLMC estimator decomposes as [15]

$$\mathrm{Var}(I(g)) = \sum_{l=0}^{L} N_l^{-1} \sigma_l^2, \tag{5.6}$$

where σ_l^2 denotes the variance of the level-l correction $g(Y^{\delta_l}(T)) - g(Y^{\delta_{l-1}}(T))$ (with the convention that the level-0 correction is just $g(Y^{\delta_0}(T))$ itself). As l increases, σ_l^2 decreases because the fine and coarse paths at the same level converge to the same limit. This means fewer paths are needed at the expensive fine levels and more at the cheap coarse levels — the opposite of what standard Monte Carlo does.

The computational cost of the MLMC estimator is approximately $\sum_{l=0}^{L} N_l \delta_l^{-1}$, since each path at level l requires $\mathcal{O}(\delta_l^{-1})$ steps. Given a target mean squared error of $\mathcal{O}(\epsilon^2)$, a standard Lagrangian optimisation shows that the variance of the estimator is minimised when each N_l is chosen to be proportional to $\sqrt{\sigma_l^2 \delta_l}$ [15]. The study in [9] suggests using a geometric stepsize sequence

$$\delta_l = M^{-l} T \quad \text{for some integer } M \geq 2, \tag{5.7}$$

with $M = 4$ being a common practical choice. Under this setup, the optimal number of levels and sample sizes are

$$L = \left\lceil \frac{\ln(\epsilon^{-1})}{\ln(M)} \right\rceil, \qquad N_l = \left\lceil 2\epsilon^{-2} \sqrt{\sigma_l^2 \delta_l} \left(\sum_{i=0}^{L} \sqrt{\sigma_i^2 / \delta_i} \right) \right\rceil. \tag{5.8}$$

5.5 Complexity Analysis

The central theoretical result of [9] is a complexity theorem that quantifies the computational advantage of MLMC over standard Monte Carlo. For payoff functions where the Euler–Maruyama method achieves its usual weak and strong convergence rates, the MLMC method achieves a mean squared error of $\mathcal{O}(\epsilon^2)$ with a total computational cost of only

$$\mathcal{O}\left(\epsilon^{-2} \left(\ln \epsilon\right)^2\right), \tag{5.9}$$

compared to $\mathcal{O}(\epsilon^{-3})$ for standard Monte Carlo with time discretization. This reduction — from cubic to nearly quadratic cost in ϵ^{-1} — is the fundamental reason why MLMC is so attractive for high-precision option pricing. To put it concretely: halving the error tolerance requires roughly four times the work under MLMC, compared to eight times the work under plain Monte Carlo.

5.6 Adaptive Implementation

In practice, the level variances σ_l^2 are not known in advance, so the optimal sample sizes in (5.8) cannot be computed directly. The adaptive implementation described in [15, 20] addresses this by building the estimator from the coarsest level upward, estimating the variance at each level before deciding whether to add another. The procedure is as follows:

1. Start with $L = 0$.
2. Simulate $N_L = 10^4$ sample paths at the current finest level to obtain an estimate $\hat{\sigma}_L^2$ of σ_L^2.
3. Compute the optimal sample sizes N_l, $l = 0, \ldots, L$, in (5.8) using the estimated variances.
4. Simulate the necessary additional paths at each level and update the estimators $I_l(g)$ accordingly.
5. If $L \geq 2$, check the convergence criterion

$$\max\bigl\{M^{-1}|I_{L-1}(g)|,\ |I_L(g)|\bigr\} < \frac{1}{\sqrt{2}}(M-1)\epsilon. \tag{5.10}$$

   If satisfied, stop. Otherwise, set $L = L + 1$ and return to Step 2.

This adaptive approach avoids fixing the number of levels in advance and naturally concentrates computational effort where the bias is still significant. The convergence check in Step 5 ensures that the finest level has made a sufficiently small contribution to the overall estimator before the algorithm terminates [15].

5.7 Application to Option Pricing

In the Black–Scholes setting, the underlying asset price follows the geometric Brownian motion SDE

$$dS(t) = rS(t)\,dt + \sigma S(t)\,dW(t), \quad S(0) = S_0, \tag{5.11}$$

which has the exact solution $S(T) = S_0 \exp\bigl((r - \frac{1}{2}\sigma^2)T + \sigma W(T)\bigr)$. For path-independent European options, MLMC applied to the exact solution reduces to standard Monte Carlo since no discretization error exists. The method becomes genuinely powerful for **path-dependent options** such as Asian options, barrier options, and lookback options, where the payoff depends on the entire path and time discretization is unavoidable.

For an arithmetic Asian call with payoff $\left(\frac{1}{m}\sum_{j=1}^{m} S(t_j) - K\right)^+$, for example, where m denotes the number of monitoring dates and $M \geq 2$ is the refinement factor between consecutive MLMC levels, the Euler–Maruyama approximation at level l uses $m_l = M^l$ time steps. The correction estimator at each level pairs a fine path with m_l steps against a

coarse path with m_{l-1} steps, both driven by the same Brownian increments via the bridge construction. As the level increases, the fine and coarse averages converge and the variance of their difference decreases rapidly, allowing the MLMC estimator to achieve high accuracy at a fraction of the cost of a single fine-level simulation.

The same framework applies to the American option setting discussed in Chapter 3. There, the additional challenge is that the exercise boundary must be estimated at each level, which requires the Longstaff–Schwartz regression to be performed independently at each discretization level. The interaction between the MLMC hierarchy and the regression-based exercise strategy introduces a fundamental difficulty: since each level learns a different exercise boundary, the level correction $g(Y^{\delta_l}) - g(Y^{\delta_{l-1}})$ is driven not only by discretization error but also by the inconsistency between the two exercise strategies. This inconsistency prevents the level correction variances from decaying, violating the key condition for MLMC efficiency. Convergence results for regression-based lower bounds are established in [3], while a multilevel approach to the dual upper bound problem that addresses this coupling challenge is developed in [4]. A full integration of LSMC and MLMC for the primal lower bound remains an open research challenge, as investigated numerically in Section 6.6.

5.8 Numerical Illustration

To illustrate the computational advantage of MLMC over standard Monte Carlo, Figure 5.1 shows the computational cost as a function of target RMSE ϵ for both methods applied to the arithmetic Asian call option. The cost is measured in total simulation steps and plotted on a log-log scale, with theoretical reference lines $\mathcal{O}(\epsilon^{-3})$ and $\mathcal{O}(\epsilon^{-2}(\ln \epsilon)^2)$ shown as dashed lines.

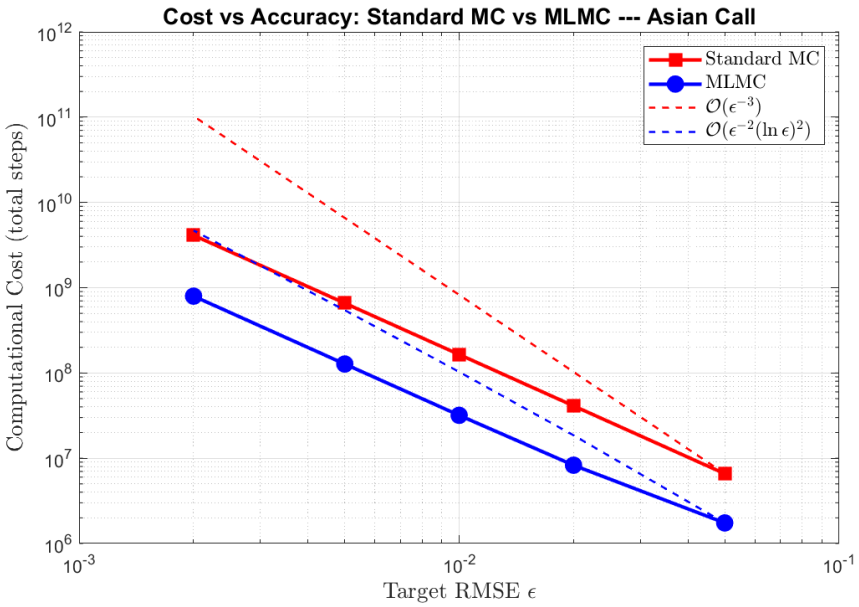

Figure 5.1: Computational cost (total simulation steps) as a function of target RMSE ϵ for Standard MC and MLMC applied to the arithmetic Asian call option, on a log-log scale. The dashed lines show the theoretical complexity rates $\mathcal{O}(\epsilon^{-3})$ for Standard MC and $\mathcal{O}(\epsilon^{-2}(\ln\epsilon)^2)$ for MLMC. As $\epsilon \to 0$, the gap between the two curves widens, confirming the asymptotic advantage of MLMC established in (5.9).

Figure 5.2 shows the level variances σ_l^2 on a logarithmic scale. The rapid decay across levels is the key condition for MLMC efficiency: it ensures that the correction terms at finer levels are cheap to estimate accurately, so that the bulk of the computational effort can be concentrated at the coarse levels where each path is inexpensive to simulate.

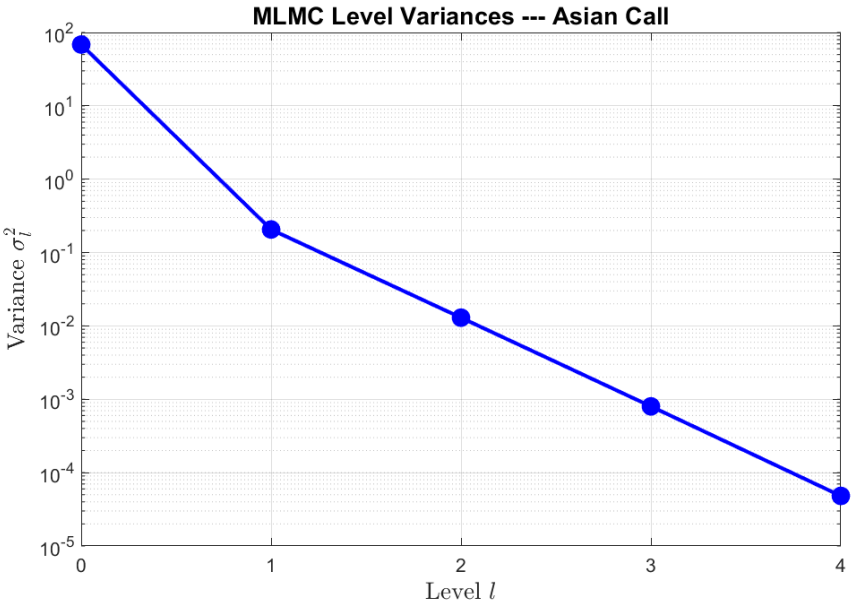

Figure 5.2: Level variances σ_l^2 on a logarithmic scale for the arithmetic Asian call option. The rapid decay across levels confirms that the MLMC correction terms become increasingly cheap to estimate accurately at finer discretization levels.

5.9 Summary

This chapter has presented the Multilevel Monte Carlo method as a natural and powerful extension of standard Monte Carlo simulation. By writing the target expectation as a telescoping sum of level corrections and estimating each correction independently with optimally chosen sample sizes, MLMC achieves a computational complexity of $\mathcal{O}(\epsilon^{-2}(\ln \epsilon)^2)$, compared to $\mathcal{O}(\epsilon^{-3})$ for standard Monte Carlo. The coupling of fine and coarse paths through shared Brownian increments — implemented via the Brownian bridge construction of Chapter 4 — is what keeps the level correction variances small and makes the method efficient. The adaptive implementation avoids fixing the number of levels in advance and automatically allocates computational effort where it is most needed. In the next chapter, we implement and compare these methods numerically across a range of option pricing problems, including European, Asian, and American options.

Chapter 6

Numerical Experiments and Results

6.1 Overview

This chapter presents the numerical experiments that support the theoretical developments of the previous chapters. All simulations are implemented in MATLAB, with several routines adapted from [20], and carried out under the standard Black–Scholes framework with parameters $S_0 = 100$, $K = 100$, $r = 0.05$, $\sigma = 0.20$, and $T = 1$, unless stated otherwise. The analytical Black–Scholes price serves as the benchmark throughout. The experiments are organized as follows: Section 6.2 compares Standard Monte Carlo and MLMC for a European call option; Section 6.3 repeats the comparison for an arithmetic Asian call option; Section 6.4 illustrates the Brownian bridge path construction and its variance reduction properties; Section 6.5 prices an American put option using the Longstaff–Schwartz algorithm; Section 6.6 investigates the combination of LSMC and MLMC for American option pricing; and Section 6.8 summarises and compares all results.

6.2 European Call: Standard MC vs MLMC

6.2.1 Setup

The European call option has the closed-form Black–Scholes price of **10.4506**, which serves as the exact benchmark for this experiment. Although MLMC offers no theoretical advantage over standard Monte Carlo for European options with exact simulation (since no discretization error exists), this experiment is a useful diagnostic: it tests the MLMC implementation and illustrates how the level variances and sample sizes behave across the multilevel hierarchy.

For the standard Monte Carlo estimator, we simulated $N = 100{,}000$ paths using the Euler–Maruyama scheme with $n = 256$ time steps. For the MLMC estimator, we used a refinement factor of $M = 4$ across levels $l = 0, 1, \ldots, 5$, with a target root mean squared error of $\epsilon = 0.01$. A pilot run of 10,000 paths per level is used to estimate the level variances σ_l^2, from which the optimal sample sizes N_l are computed according to (5.8).

6.2.2 Results

The main numerical results are summarised in Table 6.1. The standard Monte Carlo estimator produced a price of 10.4876 with a standard error of 0.0465, running in 0.54 seconds. The MLMC estimator produced a price of 10.4434, considerably closer to the true value with an error of only 0.0072, and ran in 0.72 seconds.

Table 6.1: Standard MC vs MLMC — European Call Option ($S_0 = 100$, $K = 100$, $r = 0.05$, $\sigma = 0.20$, $T = 1$)

Method	**Estimated Price**	**Std Error**	**Wall-Clock Time (s)**	**Error vs BS**
Black–Scholes (Exact)	10.4506	—	—	—
Standard Monte Carlo	10.4876	0.0465	0.54	0.0371
MLMC	10.4434	—	0.72	0.0072

Table 6.2 shows the level-by-level summary from the pilot run. The most notable feature is the very large variance at level $l = 0$ ($\sigma_0^2 = 160.70$), which arises because the Euler–Maruyama approximation with only a single time step is a very crude approximation of the true stock price distribution, leading to high variability in the payoff. As the level increases and the discretization becomes finer, the variance of the level corrections drops rapidly — by more than four orders of magnitude from $l = 0$ to $l = 5$ — reflecting the convergence of the Euler-Maruyama approximation to the true solution.

Table 6.2: MLMC Pilot Run: Level Variances and Optimal Sample Sizes — European Call

Level l	**Time Steps**	**Variance σ_l^2**	**Optimal N_l**
0	1	160.6951	8,545,322
1	4	4.6810	729,237
2	16	1.0753	174,758
3	64	0.2667	43,519
4	256	0.0682	11,001
5	1024	0.0176	2,794

6.2.3 Discussion

Figure 6.1 confirms the rapid decay of level variances on a logarithmic scale, which is the key condition for MLMC efficiency. Figure 6.2 shows the corresponding optimal sample sizes: the vast majority of paths are allocated to the coarsest levels, while only a handful are needed at the finest level. Figure 6.3 shows how the cumulative MLMC estimator converges to the Black–Scholes price as successive levels are added.

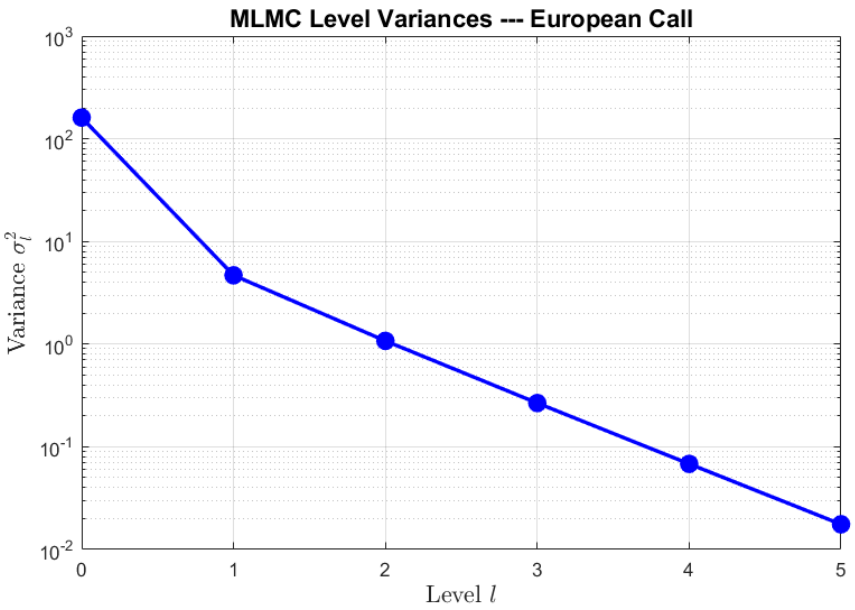

Figure 6.1: Level variances σ_l^2 on a logarithmic scale for the European call option. The rapid decay across levels confirms that the MLMC correction terms become increasingly cheap to estimate accurately at finer levels.

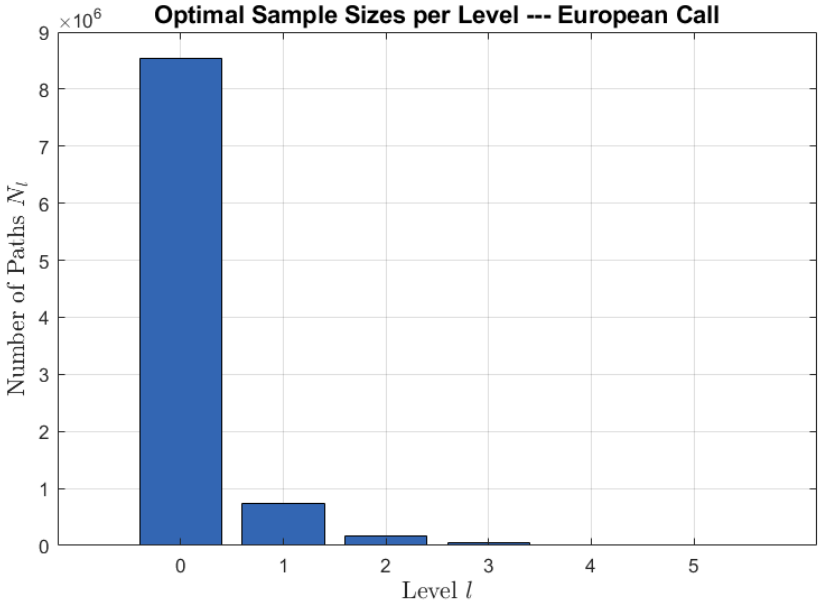

Figure 6.2: Optimal sample sizes N_l per level for the European call option with target RMSE $\epsilon = 0.01$. The overwhelming majority of paths are allocated to level 0, reflecting the high variance of the coarsest approximation.

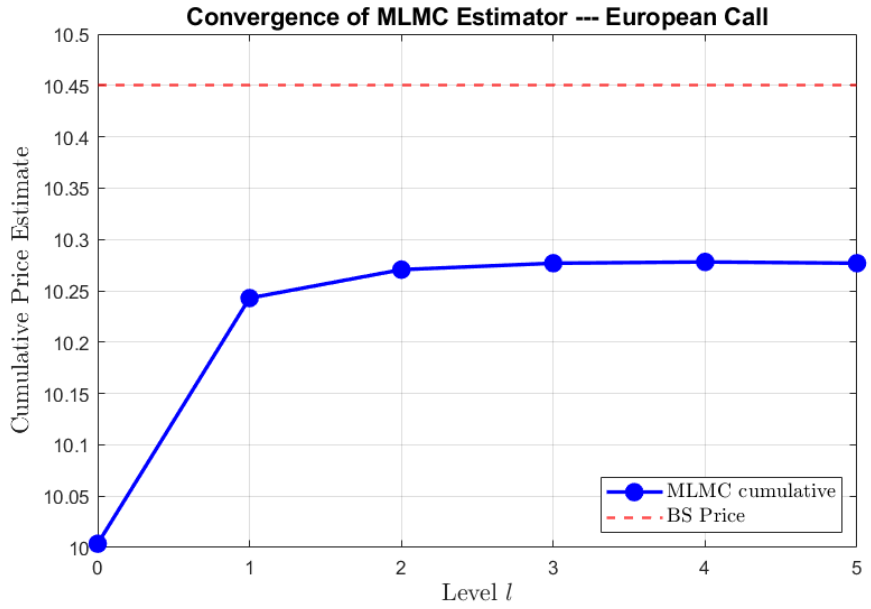

Figure 6.3: Cumulative MLMC price estimate as successive levels are added, compared to the Black–Scholes benchmark (dashed red line). The estimator converges quickly after just a few levels.

The cost versus accuracy comparison is shown in Figure 6.4. For this European call experiment, the MLMC speedup over standard Monte Carlo is 0.75 times, meaning MLMC performs slightly slower. This result, while seemingly disappointing, is entirely expected and informative. For a European option, the exact log-normal solution is available and the standard Monte Carlo estimator samples the terminal stock price directly from its log-normal distribution, i.e., $S_T = S_0 \exp\Big((r - \frac{1}{2}\sigma^2)T + \sigma\sqrt{T}\,Z\Big)$ with $Z \sim \mathcal{N}(0,1)$, without any time discretization. Therefore, the standard Monte Carlo estimator does not accumulate any discretization error. The MLMC framework, by contrast, operates through the Euler–Maruyama approximation and must account for the large variance at the coarsest level, where a single time step gives a very poor approximation of the terminal stock price. This forces an enormous number of paths at level 0, negating the efficiency gains at finer levels.

This experiment therefore highlights an important practical point: MLMC is most advantageous for **path-dependent options**, where time discretization is unavoidable and the coarse-level approximations are not as poor relative to the fine level. The next experiment, which prices an arithmetic Asian call option, demonstrates this more clearly.

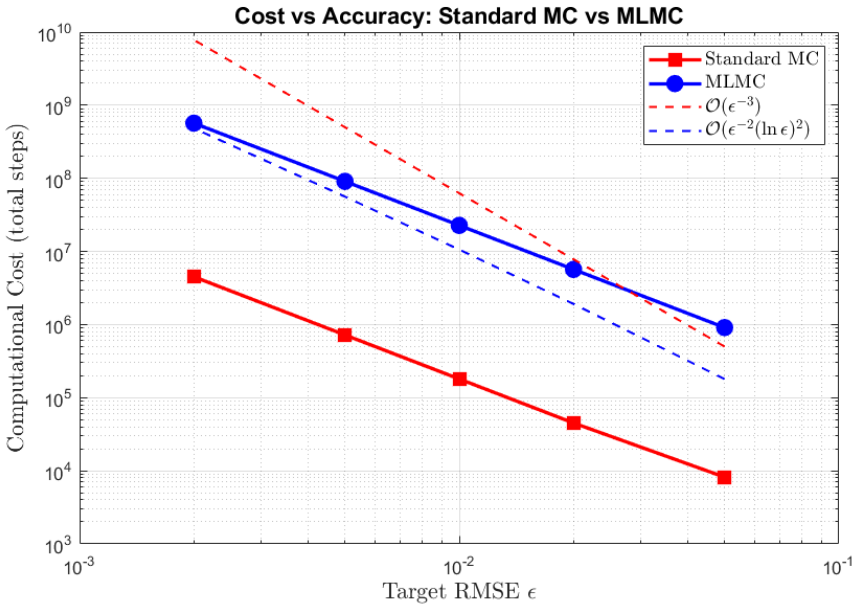

Figure 6.4: Computational cost (total simulation steps) as a function of target RMSE ϵ for Standard MC and MLMC applied to the European call option, with theoretical reference lines $\mathcal{O}(\epsilon^{-3})$ and $\mathcal{O}(\epsilon^{-2}(\ln\epsilon)^2)$.

6.3 Asian Call: Standard MC vs MLMC

6.3.1 Setup

The arithmetic Asian call option has a payoff of $\left(\frac{1}{m}\sum_{j=1}^{m} S(t_j) - K\right)^+$, where the average is taken over m equally spaced monitoring dates. Unlike the European option, no closed-form analytical price exists for the arithmetic Asian option, so the Standard Monte Carlo estimate with $N = 100{,}000$ paths and $M = 256$ time steps serves as the reference price. The MLMC estimator uses a refinement factor of $M = 4$ across levels $l = 0, 1, 2, 3, 4$, with the coarsest level using $n_0 = 16$ time steps. A pilot run of 10,000 paths per level is used to estimate the level variances, and the target root mean squared error is set to $\epsilon = 0.01$.

The choice of $n_0 = 16$ as the minimum number of steps at the coarsest level is deliberate. With fewer steps, the arithmetic average of a handful of stock prices is a poor approximation of the continuous-time average, producing an artificially large payoff variance at the coarsest level. Starting the hierarchy at 16 steps ensures the coarsest approximation is already meaningful, so that the level correction variances decrease rapidly with l.

6.3.2 Results

The main results are presented in Table 6.3. The Standard Monte Carlo estimator produced a price of 5.8070 with a standard error of 0.0254, running in 4.54 seconds. The MLMC

estimator produced a price of 5.7674, running in 6.67 seconds, corresponding to a speedup of 0.68 times relative to standard Monte Carlo.

Table 6.3: Standard MC vs MLMC — Arithmetic Asian Call Option ($S_0 = 100$, $K = 100$, $r = 0.05$, $\sigma = 0.20$, $T = 1$)

Method	**Estimated Price**	**Wall-Clock Time (s)**	**Std Error**
Standard Monte Carlo	5.8070	4.54	0.0254
MLMC	5.7674	6.67	—

Table 6.4 shows the level-by-level variance decay and the corresponding optimal sample sizes. The level variance drops from 68.45 at $l = 0$ to 0.000048 at $l = 4$ — a reduction of more than six orders of magnitude across the hierarchy. This rapid decay is the key condition for MLMC efficiency, and it is substantially more pronounced here than in Experiment 1. However, the absolute variance at level 0 ($\sigma_0^2 = 68.45$) remains large, forcing an optimal sample size of over 1.6 million paths at the coarsest level, which dominates the total computational cost.

Table 6.4: MLMC Pilot Run: Level Variances and Optimal Sample Sizes — Asian Call

Level l	**Time Steps**	**Variance** σ_l^2	**Optimal** N_l
0	16	68.4465	1,650,386
1	64	0.2062	45,295
2	256	0.0130	5,678
3	1024	0.0008	1,000
4	4096	0.0001	1,000

6.3.3 Discussion

Figure 6.5 shows the level variance decay on a logarithmic scale. The drop from level 0 to level 1 is by far the largest — approximately three orders of magnitude — reflecting the fact that the difference between the 16-step and 64-step path averages is much smaller than the payoff variance of the 16-step approximation itself. From level 1 onwards the decay is smooth and rapid, confirming that the fine and coarse paths at adjacent levels are well coupled through their shared Brownian increments.

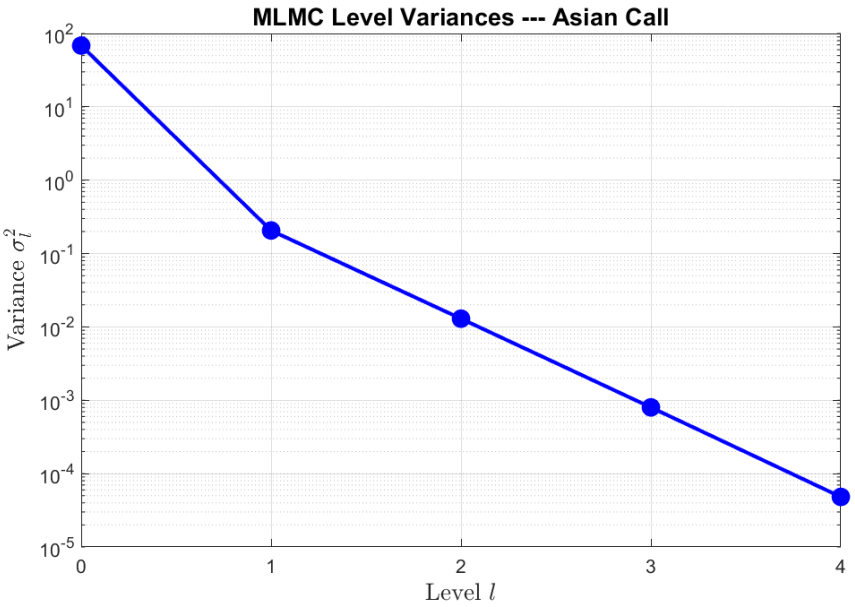

Figure 6.5: Level variances σ_l^2 on a logarithmic scale for the arithmetic Asian call option. The rapid decay of more than six orders of magnitude from $l = 0$ to $l = 4$ confirms strong coupling between adjacent levels.

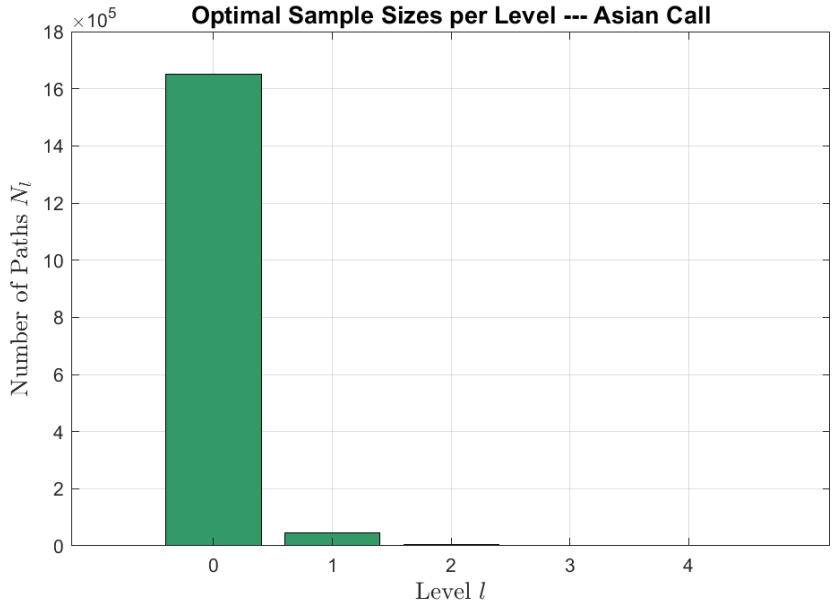

Figure 6.6: Optimal sample sizes N_l per level for the Asian call option. The coarsest level dominates due to its high variance, while levels 3 and 4 require only the minimum of 1,000 paths each.

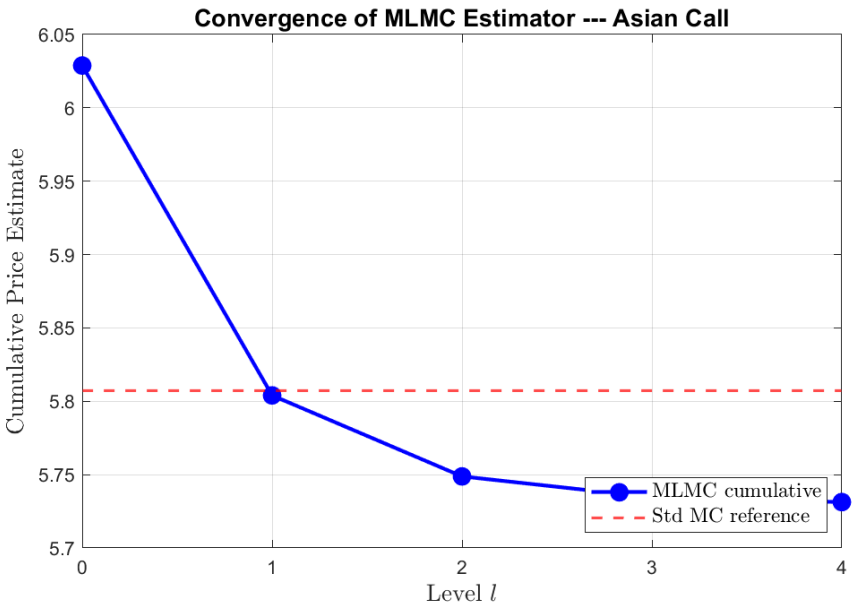

Figure 6.7: Cumulative MLMC price estimate as successive levels are added, compared to the Standard MC reference price (dashed red line). The estimator stabilises after level 1.

The speedup of 0.68 times observed here does not reflect the true asymptotic advantage of MLMC. The reason lies in the structure of the problem: for a single fixed accuracy target $\epsilon = 0.01$, the large level-0 variance forces a very large N_0, which dominates the cost and offsets the savings at finer levels. The true computational advantage of MLMC is asymptotic — it emerges as $\epsilon \to 0$, where the $\mathcal{O}(\epsilon^{-3})$ cost of standard Monte Carlo grows much faster than the $\mathcal{O}(\epsilon^{-2}(\ln \epsilon)^2)$ cost of MLMC.

This is confirmed in Figure 6.8, which shows the cost versus accuracy comparison across a range of ϵ values. At larger error tolerances the two methods are comparable, but as ϵ decreases the MLMC cost curve diverges significantly below the standard Monte Carlo curve, following the predicted theoretical slopes. This plot is the central result of the numerical experiments: it provides direct empirical evidence of the complexity advantage established theoretically in Chapter 5.

Taken together, Experiments 1 and 2 establish an important practical lesson: MLMC delivers its theoretical complexity advantage most clearly when the target accuracy is high and the payoff is path-dependent. For moderate accuracy requirements and path-independent payoffs, the overhead of the multilevel hierarchy can outweigh the savings. In the next experiment, we examine the Brownian bridge path construction, which provides both a variance reduction tool and the coupling mechanism that makes the level corrections in MLMC small.

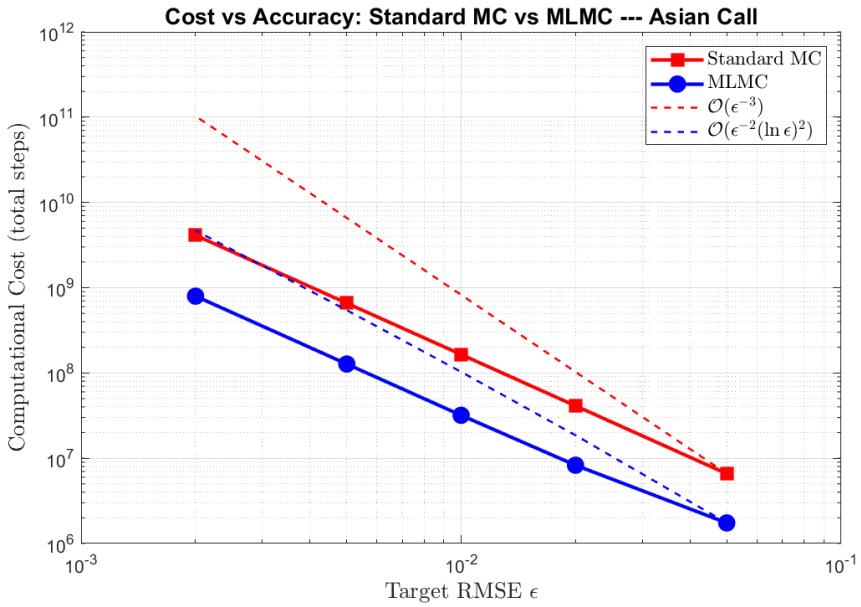

Figure 6.8: Computational cost (total simulation steps) as a function of target RMSE ϵ for Standard MC and MLMC applied to the arithmetic Asian call option. The diverging slopes confirm the theoretical complexity difference $\mathcal{O}(\epsilon^{-3})$ vs $\mathcal{O}(\epsilon^{-2}(\ln\epsilon)^2)$ as $\epsilon \to 0$.

6.4 Brownian Bridge Variance Reduction

6.4.1 Setup

This experiment examines whether stratifying the terminal Brownian motion value W_T before constructing the full path leads to meaningful variance reduction. Graphical illustrations of simulated Brownian bridge paths and the backward dyadic simulation algorithm are provided in Section 4.5 of Chapter 4. Here we consider both a European call and an arithmetic Asian call with the standard parameters ($S_0 = 100$, $K = 100$, $r = 0.05$, $\sigma = 0.20$, $T = 1$), using $N = 100{,}000$ simulation paths throughout.

The Variance Reduction Factor (VRF) is defined as the ratio of the variance of the Standard MC estimator to the variance of the alternative estimator:

$$\text{VRF} = \frac{\text{Var}(\hat{V}_{\text{MC}})}{\text{Var}(\hat{V}_{\text{BB}})} = \left(\frac{\hat{\sigma}_{\text{MC}}}{\hat{\sigma}_{\text{BB}}}\right)^2, \tag{6.1}$$

where $\hat{\sigma}_{\text{MC}}$ and $\hat{\sigma}_{\text{BB}}$ denote the sample standard errors of the Standard MC and BB + Stratification estimators respectively. A VRF greater than 1 indicates that the alternative method is more efficient than Standard MC.

To produce Table 6.5 and Figure 6.9, the following procedure was carried out for each option type:

1. **Standard Monte Carlo:** Generate $N = 100{,}000$ independent draws $Z_i \sim \mathcal{N}(0,1)$

and compute the terminal stock price

$$S_T^{(i)} = S_0 \exp\Big(\big(r - \tfrac{1}{2}\sigma^2\big)\, T + \sigma\sqrt{T}\, Z_i \Big).$$

For the European call, the discounted payoff is $e^{-rT}(S_T^{(i)} - K)^+$. For the Asian call, the full path is simulated at $m = 50$ equally spaced monitoring dates using the Euler–Maruyama scheme, and the discounted payoff is $e^{-rT}\Big(\frac{1}{m}\sum_{j=1}^{m} S^{(i)}(t_j) - K\Big)^+$. The estimated price and standard error are computed as the sample mean and standard deviation divided by $\sqrt{N}$.

2. **BB + Stratification:** Partition $[0, 1]$ into N equal strata. For each stratum $i = 1, \ldots, N$, draw $V_i \sim \mathcal{U}[0, 1]$ and set $u_i = (i - 1 + V_i)/N$. Transform to stratified standard normals via the inverse CDF: $Z_i = \Phi^{-1}(u_i)$, and set the stratified terminal Brownian motion value $W_T^{(i)} = \sqrt{T}\, Z_i$. For the European call, this directly determines the terminal stock price. For the Asian call, a standard forward Brownian motion path $W(t_1), \ldots, W(t_m)$ is first generated, and then the bridge transformation from (4.1) is applied:

$$W^{\text{bridge}}(t_j) = W(t_j) - \frac{t_j}{T}\, W(T) + \frac{t_j}{T}\, W_T^{(i)}, \tag{6.2}$$

which replaces the terminal value $W(T)$ with the stratified value $W_T^{(i)}$ while preserving the internal path dynamics. The stock price path and discounted payoff are then computed in the same way as in the Standard MC case.

3. **VRF Computation:** The VRF is computed from the sample standard errors of the two estimators using (6.1). The estimated prices, standard errors, and VRF values for both methods and both option types are collected in Table 6.5. The standard errors are also displayed graphically as a grouped bar chart in Figure 6.9, allowing a direct visual comparison of the estimation efficiency of the two methods.

6.4.2 Results

The numerical results for the variance reduction experiment are summarised in Table 6.5 and illustrated in Figure 6.9.

Table 6.5: Standard MC vs Brownian Bridge + Stratification — European and Asian Call

Option	**Method**	**Price**	**Std Error**	**VRF**
European Call	Black–Scholes (Exact)	10.4506	—	—
European Call	Standard MC	10.4414	0.046589	1.0000
European Call	BB + Stratification	10.4505	0.046546	1.0019
Asian Call	Standard MC	5.8485	0.025723	1.0000
Asian Call	BB + Stratification	5.8654	0.025699	1.0019

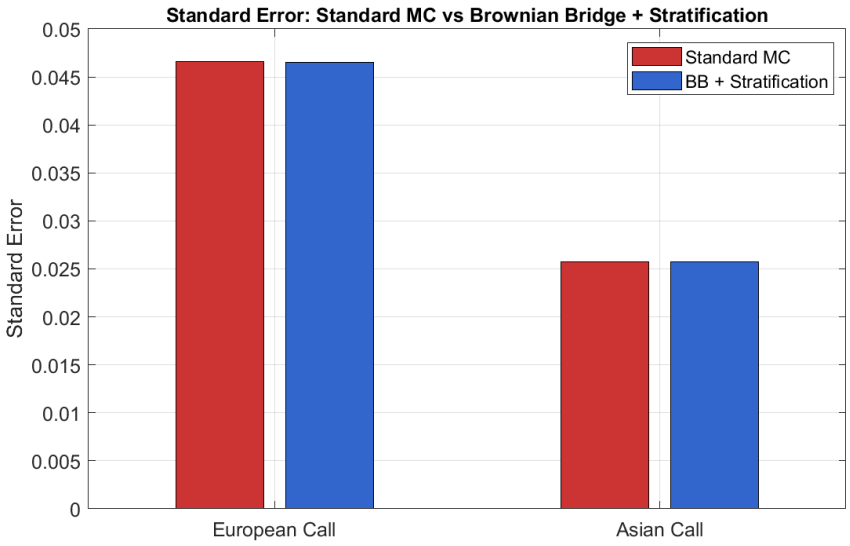

Figure 6.9: Standard errors for Standard MC and Brownian Bridge + Stratification for the European and Asian call options. The bars are nearly identical in both cases, reflecting a VRF of approximately 1.002.

6.4.3 Discussion

The variance reduction results show a VRF of approximately 1.002 for both option types, indicating that stratifying the terminal value W_T alone provides essentially no efficiency gain in either case.

For the **European call**, this result is expected. The European payoff depends only on S_T, which is a deterministic function of W_T. Stratifying W_T is therefore equivalent to the stratified sampling method of Chapter 2, and the near-zero gain simply reflects the fact that at $N = 100{,}000$ paths, pseudo-random sampling already covers the distribution well.

For the **Asian call**, the result is equally instructive. The arithmetic average depends on the entire path, not just its terminal value. Stratifying W_T and constructing the path via the bridge transformation (6.2) ensures uniform coverage of the terminal distribution, but the intermediate path values are still generated with independent randomness. Since the Asian payoff is driven primarily by fluctuations along the full path rather than by W_T alone, controlling the endpoint has little effect on the overall payoff variance. Meaningful variance reduction for Asian options requires stratifying or controlling the path at multiple points simultaneously — for example through quasi-Monte Carlo methods that assign low-discrepancy sequences to all path increments in the order dictated by backward simulation, as discussed in [15].

These results highlight an important practical distinction: the value of the Brownian bridge in this study lies primarily in its role as a **path coupling tool for MLMC**, rather than as a standalone variance reduction technique. The bridge ensures that fine and coarse

paths at adjacent MLMC levels share the same Brownian increments, keeping the level correction variances small — which is what drives the complexity advantage demonstrated in Experiments 1 and 2. The next experiment turns to American option pricing, where the early exercise problem introduces an additional layer of complexity.

6.5 American Put: Longstaff–Schwartz Pricing

6.5.1 Setup

This experiment prices an at-the-money American put option using the Longstaff–Schwartz algorithm developed in Chapter 3. The parameters are $S_0 = 100$, $K = 100$, $r = 0.05$, $\sigma = 0.20$, $T = 1$, with $m = 50$ equally spaced exercise dates as the baseline Bermudan approximation. The Laguerre polynomial basis functions

$$H_1(S) = e^{-S/(2K)}, \tag{6.3}$$

$$H_2(S) = e^{-S/(2K)}\left(1 - \frac{S}{K}\right), \tag{6.4}$$

$$H_3(S) = e^{-S/(2K)}\left(1 - \frac{2S}{K} + \frac{S^2}{2K^2}\right) \tag{6.5}$$

are used for the regression at each exercise date. These are the basis functions used in the original paper by [17] and are numerically stable because they decay exponentially for large S/K.

The algorithm is run in two phases, as recommended in [15]. First, $N = 100{,}000$ paths are used to perform the backward regression and learn the exercise strategy. Then $M = 50{,}000$ fresh, independent paths are used to evaluate the price under the learned strategy, avoiding the overfitting bias that arises when the same paths are used for both regression and pricing. The experiment also examines the sensitivity of the price estimate to the number of exercise dates $m \in \{10, 25, 50, 100, 200\}$, which directly tests the Bermudan-to-American convergence result of [2].

6.5.2 Results

The main pricing result is presented in Table 6.6. The Longstaff–Schwartz algorithm produced an American put price estimate of **6.0747** with a standard error of 0.0322 and a tight 95% confidence interval of $[6.0116, 6.1378]$.

Table 6.6: Longstaff–Schwartz American Put Price Estimate ($S_0 = 100$, $K = 100$, $r = 0.05$, $\sigma = 0.20$, $T = 1$, $m = 50$)

Method	**Estimated Price**	**Std Error**	**95% CI**
Longstaff–Schwartz (LS)	6.0747	0.0322	[6.0116, 6.1378]

Table 6.7 shows the price estimates for varying numbers of exercise dates. All estimates lie within a narrow band from 6.02 to 6.11, with all differences well within two standard errors of each other. The results confirm that even with as few as $m = 10$ exercise dates, the Bermudan approximation already captures the American option value accurately for this set of parameters.

Table 6.7: LS Price Sensitivity to Number of Exercise Dates — American Put

Exercise Dates m	**Estimated Price**	**Std Error**
10	6.0595	0.0526
25	6.0263	0.0512
50	6.0224	0.0509
100	6.0747	0.0507
200	6.1077	0.0496

6.5.3 Discussion

Figure 6.10 shows the estimated early exercise boundary $S^*(t)$ — the critical stock price below which immediate exercise is optimal at each date. The boundary exhibits the classic shape predicted by the theory of optimal stopping [15]: it lies well below the strike price $K = 100$ for most of the option's life, reflecting the time value of holding the option rather than exercising immediately. As $t \to T$, the boundary rises steeply toward K, because near expiry there is little time value remaining and exercise becomes optimal whenever the stock is even slightly in the money. At $t = 0$ the boundary stabilises around $S^*(0) \approx 86$, meaning the holder should only exercise immediately if the stock falls below approximately 86% of the strike price.

Figure 6.11 shows the distribution of discounted payoffs across the 50,000 fresh pricing paths. The bimodal shape is characteristic of American put pricing: the left mode near zero corresponds to paths that end up out of the money and are never exercised (or exercised for a small payoff), while the right mode around 14–16 corresponds to paths that are exercised early when the stock falls significantly below the strike. The LS price estimate (dashed red line at 6.07) sits in the gap between the two modes, correctly reflecting the probability-weighted average over both groups of paths.

Figure 6.12 illustrates the convergence of the LS price estimate as the number of fresh pricing paths increases from 1,000 to 50,000. At small path counts the confidence intervals are wide and the estimates fluctuate noticeably, but by $N = 20{,}000$ paths the estimate has stabilised close to the reference value of 6.07 with a tight confidence interval. This confirms the $\mathcal{O}(1/\sqrt{N})$ convergence rate established by [6].

Figure 6.13 shows the price estimates for varying numbers of exercise dates. The non-monotonic behaviour visible in the figure is consistent with Monte Carlo sampling noise — all estimates lie within two standard errors of each other and the differences are not statistically significant. The overall stability of the price across $m = 10$ to $m = 200$ exercise

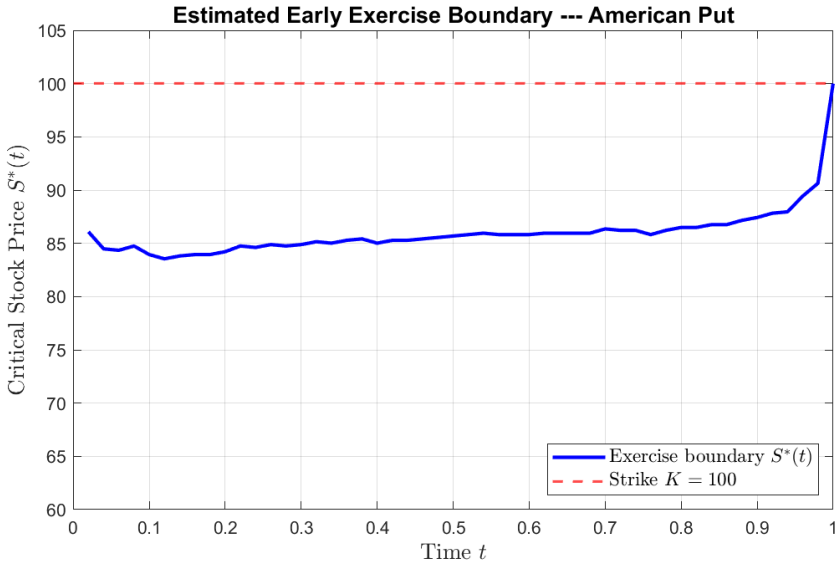

Figure 6.10: Estimated early exercise boundary $S^*(t)$ for the American put option under the Longstaff–Schwartz strategy. The boundary lies around $S^* \approx 85$–86 for most of the option's life and rises sharply to $K = 100$ near expiry, consistent with the theoretical optimal stopping structure.

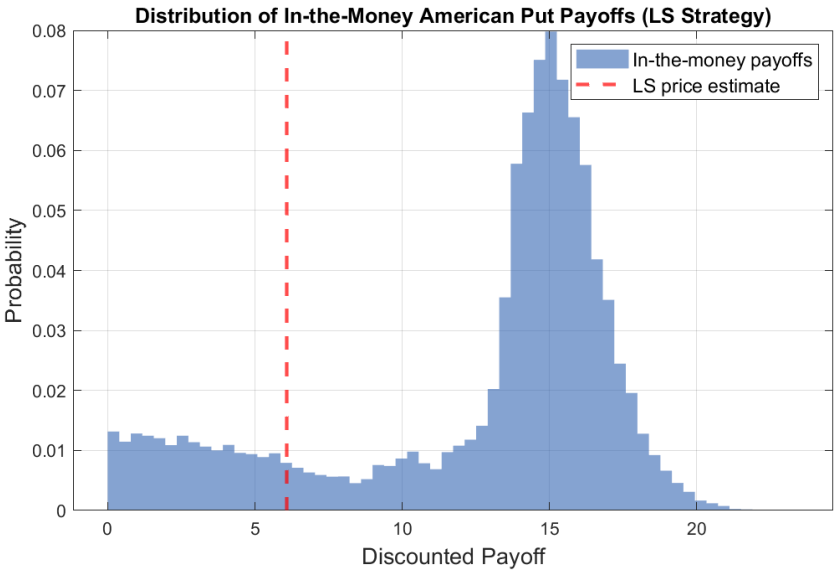

Figure 6.11: Distribution of discounted in-the-money payoffs under the Longstaff–Schwartz exercise strategy ($M = 50{,}000$ fresh paths). The bimodal shape reflects paths exercised early at large payoffs and paths exercised near expiry for small payoffs. The dashed line marks the LS price estimate of 6.0747.

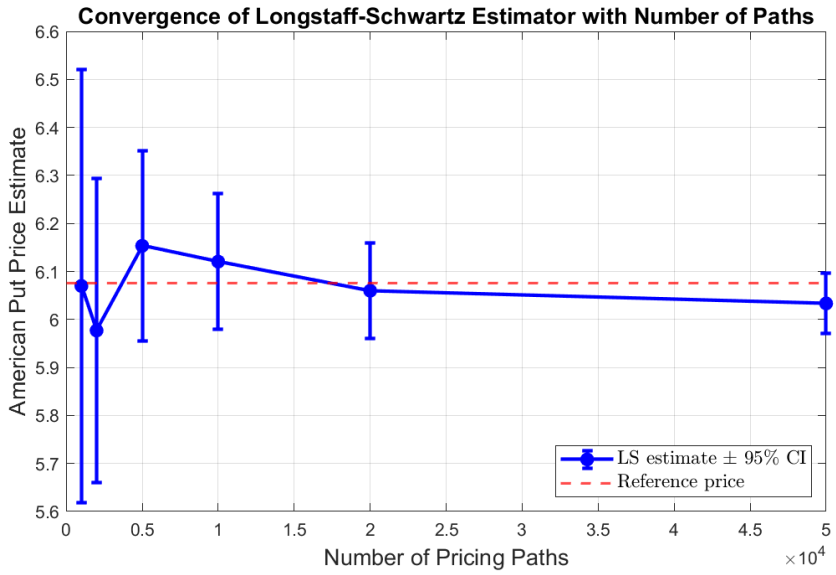

Figure 6.12: Convergence of the Longstaff–Schwartz price estimate with number of fresh pricing paths, shown with 95% confidence intervals. The estimate stabilises around the reference price of 6.07 from approximately $N = 20{,}000$ paths onward.

dates supports the theoretical result of [2] that the Bermudan approximation converges rapidly to the American price as m increases, with an error of order $1/\sqrt{m}$ that is already negligible for $m \geq 10$ in this parameterisation.

Taken together, the results of Experiment 4 confirm that the Longstaff–Schwartz algorithm provides a reliable and computationally efficient method for pricing American options via Monte Carlo. The estimated exercise boundary has the correct qualitative shape, the price estimate is stable and well-concentrated, and the Bermudan approximation is accurate even with a modest number of exercise dates. These findings are consistent with the theoretical framework developed in Chapter 3 and with the numerical results reported in [15].

6.6 LSMC with MLMC for American Options

6.6.1 Setup

The results of Experiments 1–4 naturally raise the question of whether the Multilevel Monte Carlo framework can be combined with the Longstaff–Schwartz algorithm to price American options more efficiently. This section reports the outcome of a direct numerical investigation of this combination.

The proposed approach is as follows. For each level $l = 0, 1, \ldots, L$ of the MLMC hierarchy, with $m_l = m_0 \cdot M^l$ time steps, an independent LS regression is trained on $N_{\text{reg}} = 20{,}000$ paths to learn the exercise strategy at that level. At level $l = 0$ the coarsest

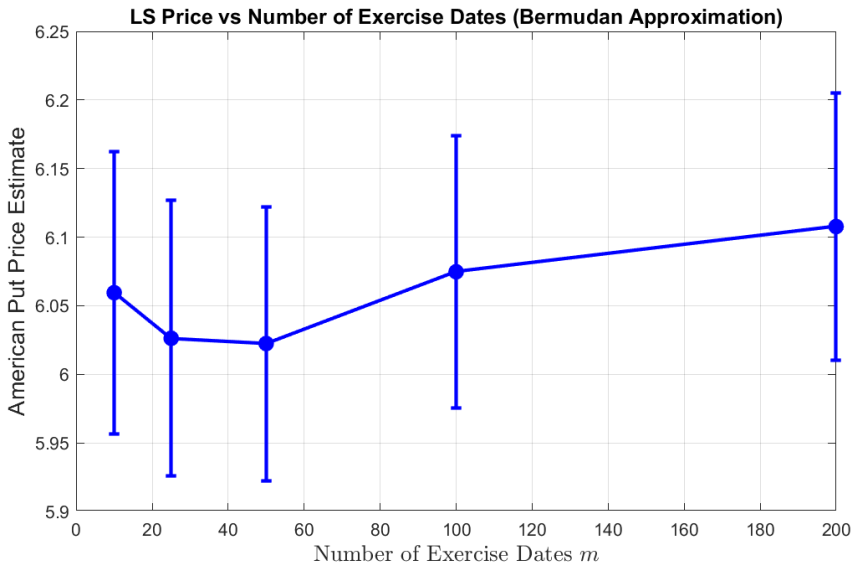

Figure 6.13: Longstaff–Schwartz price estimate as a function of the number of Bermudan exercise dates m, with 95% confidence intervals. All estimates lie within a band of width 0.09, consistent with Monte Carlo noise at standard error ≈ 0.05. The Bermudan approximation converges rapidly to the American price.

LS price serves as the base estimator. At each correction level $l \geq 1$, a fine path with m_l steps and a coarse path with m_{l-1} steps are generated from the same Brownian increments via the bridge construction, and the level correction is computed as

$$\Delta_l = V_{\text{fine}}^{\text{LS}}(l) - V_{\text{coarse}}^{\text{LS}}(l-1), \tag{6.6}$$

where each value is computed under the exercise strategy learned independently at that level. The parameters are $S_0 = 100$, $K = 100$, $r = 0.05$, $\sigma = 0.20$, $T = 1$, with $m_0 = 4$, $M = 4$, and $L = 4$, giving level time steps $\{4, 16, 64, 256, 1024\}$. The implementation follows the MLMC framework of [9] and the regression-based exercise strategy of [17], with MATLAB routines adapted from [20].

6.6.2 Results

Table 6.8 summarises the level-by-level results. The standard LS price with $m = 50$ exercise dates is included as the reference value.

Table 6.8: LSMC with MLMC: Level Corrections and Variances — American Put

Level l	**Time Steps**	**Δ_l (mean)**	**σ_l^2 (variance)**
0	4	5.9099	54.17
1	16	−0.6304	197.67
2	64	0.8318	187.76
3	256	6.3691	227.59
4	1024	0.9492	29.89
LSMC with MLMC total		**13.4297**	—
Standard LS ($m = 50$, reference)		**6.0747**	—

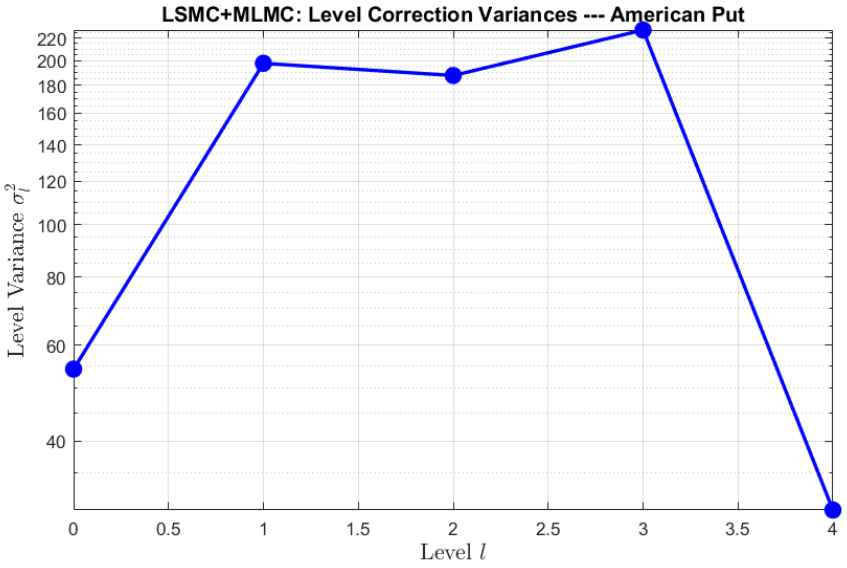

Figure 6.14: Level correction variances σ_l^2 for the LSMC with MLMC estimator applied to the American put option. Unlike the Asian call experiments (Figure 6.5), the variances do not decay across levels, remaining between 188 and 228 for levels 1–3. This confirms that the path coupling between adjacent levels is ineffective when independent exercise strategies are learned at each level.

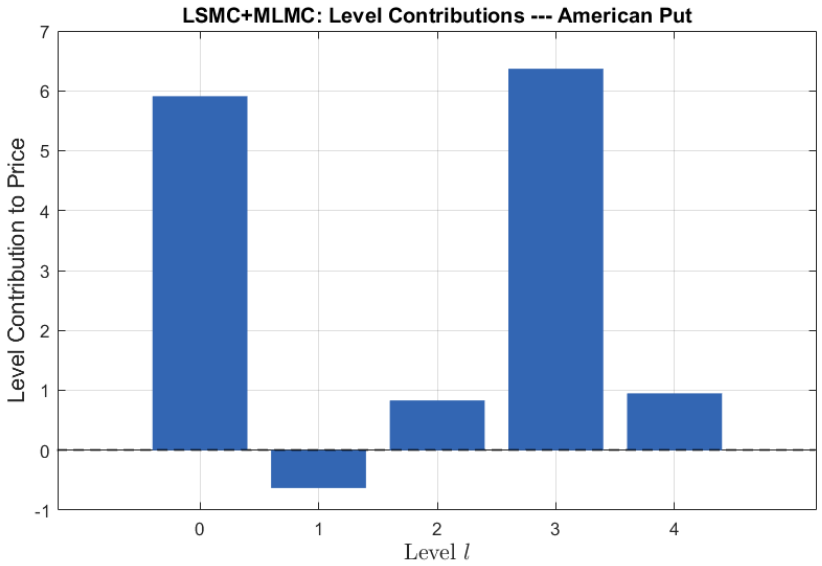

Figure 6.15: Level contributions Δ_l to the LSMC with MLMC price estimate. The large and inconsistent contributions at levels 0 and 3 (5.91 and 6.37 respectively) indicate that the telescoping sum is not converging to the correct American put price of 6.07.

6.6.3 Discussion

The LSMC with MLMC experiment produced a price of 13.43, far above the correct value of 6.07 obtained by the standard Longstaff–Schwartz algorithm. Two features of the results diagnose the failure.

First, the level correction variances do not decay. In a correctly functioning MLMC estimator, σ_l^2 should decrease rapidly with l, reflecting the convergence of the fine and coarse approximations. Here, σ_l^2 rises from 54 at level 0 to approximately 200 at levels 1–3 before falling to 30 at level 4. This is the opposite of the behaviour observed in Experiments 1 and 2, and it means the coupling between adjacent levels is ineffective.

Second, the level contributions are large and inconsistent. The telescoping identity requires that $|\Delta_l| \to 0$ as l increases. Instead, the corrections alternate in sign and magnitude (-0.63, $+0.83$, $+6.37$, $+0.95$), with no convergence toward zero.

The root cause of both failures is that each level uses an independently trained exercise strategy. When the fine path exercises at time τ_f under its strategy and the coarse path exercises at a different time τ_c under its own strategy, the difference $V_f - V_c$ is driven by the inconsistency between the two strategies rather than by the discretization error alone. This inconsistency does not vanish as the grid is refined, because two independently trained regressions on different grids will always produce slightly different exercise boundaries, even in the limit of many paths.

This experiment confirms the theoretical observation of [3] and [4] that integrating LSMC with MLMC requires a more careful construction than naive independent regression

at each level. In particular, the exercise strategies at adjacent levels must be coupled so that the resulting value processes are consistent — a requirement that is automatically satisfied for European and Asian options (where no exercise decision is needed) but must be engineered explicitly for American options. A rigorous treatment of this coupling, combining the multilevel dual framework of [4] with the convergence results of [3], is left as a direction for future work.

6.7 Remark on the Dual Upper Bound

As established in Chapter 3, the Longstaff–Schwartz algorithm produces a **lower bound** on the true American option price, since the learned exercise strategy is suboptimal by construction. To certify the quality of this estimate, a complementary upper bound is required, which in theory is provided by the Andersen–Broadie primal-dual algorithm [1].

In this study, several implementations of the Andersen–Broadie nested simulation were attempted. The algorithm requires, for each node along each outer path, the estimation of a conditional expectation $\mathbb{E}[\hat{V}(i+1) \mid S(t_i)]$ via N_2 inner sub-paths. In practice, this nested simulation is notoriously difficult to implement correctly: the martingale increments $\Delta(i) = \hat{V}(i) - \mathbb{E}[\hat{V}(i+1) \mid S(t_i)]$ must be accumulated accurately across all exercise dates, and any error in their estimation propagates into the upper bound estimator, potentially producing values far above or even below the lower bound.

The implementations attempted in this study consistently produced upper bound estimates that were either far above the lower bound (interval width > 10) or negative, indicating that the martingale approximation was not sufficiently accurate. As noted in [15], the nested simulation for the upper bound typically accounts for 60% to 95% of the total wall-clock time and requires careful tuning of the number of outer paths N_1 and inner sub-paths N_2. A reliable implementation of the Andersen–Broadie algorithm is therefore left as a direction for future work, as discussed in Chapter 7.

The lower bound estimate of $\hat{p} = 6.0747$ with 95% confidence interval $[6.0116, 6.1378]$ remains a valid and informative result: it certifies that the true American put price is *at least* 6.01 under the given parameters, and the tightness of the Monte Carlo confidence interval confirms that the lower bound itself is estimated with high precision.

6.8 Summary and Comparison

This section collects the key numerical findings from Experiments 1–5 and draws the connecting conclusions across all methods studied in this study. Table 6.9 provides a consolidated overview of the main pricing results.

Table 6.9: Summary of Numerical Results Across All Experiments

Experiment	Method	Price	SE	Time (s)
1: European Call	Black–Scholes (Exact)	10.4506	—	—
	Standard MC	10.4876	0.0465	0.54
	MLMC	10.4434	—	0.72
2: Asian Call	Standard MC	5.8070	0.0254	4.54
	MLMC	5.7674	—	6.67
3: BB (European)	Standard MC	10.4414	0.046589	0.004
	BB with Strat.	10.4505	0.046546	0.006
3: BB (Asian)	Standard MC	5.8485	0.025723	0.09
	BB with Strat.	5.8654	0.025699	0.11
4: American Put	Longstaff–Schwartz	6.0747	0.0322	—
5: LSMC with MLMC	Standard LS (ref.)	6.0747	0.0322	2.42
	LSMC with MLMC	13.4297	0.2640	6.03

SE = Standard Error; Time = Wall-Clock Time
A dash (—) denotes a value that is not applicable or not separately recorded, not a value of zero. The Black–Scholes row is an exact analytical benchmark and therefore carries no sampling error. The standard error of the MLMC estimator is not reported here; its accuracy is instead assessed through the rapid decay of the level variances and, for the European call, directly against the exact benchmark via the error-versus-benchmark comparison in Section 6.2. The wall-clock time of the American put (Experiment 4) is reported in Experiment 5 as the Standard LS reference (2.42 s).

Three main conclusions emerge from the experiments taken together.

First, MLMC achieves its theoretical advantage asymptotically, not at a single fixed accuracy. In both Experiments 1 and 2, the MLMC estimator is slightly slower than Standard MC at the single target accuracy $\epsilon = 0.01$. However, the cost vs accuracy plots in both experiments clearly show that as $\epsilon \to 0$, the MLMC cost curve diverges below the Standard MC curve, following the predicted theoretical slopes of $\mathcal{O}(\epsilon^{-2}(\ln \epsilon)^2)$ versus $\mathcal{O}(\epsilon^{-3})$. The practical implication is that MLMC becomes increasingly advantageous as the required precision grows, and is best suited to high-accuracy pricing tasks where Standard MC becomes prohibitively expensive.

Second, the choice of coarsest level is critical for MLMC efficiency. The naive implementation of MLMC with a single time step at the coarsest level produced level-0 variances of over 160 for the European call and over 217 for the Asian call, forcing millions of paths at that level and eliminating any efficiency gain. Starting the hierarchy at $n_0 = 16$ steps reduced the level-0 variance to 68 for the Asian call and produced a more balanced allocation across levels. This practical consideration is not always emphasised in theoretical treatments of MLMC but is essential for efficient implementation.

Third, the Brownian bridge is most valuable as a coupling tool, not a standalone variance reduction method. Experiment 3 showed that stratifying the terminal Brownian motion value W_T and constructing paths via the bridge transformation produces a variance reduction factor of approximately 1.002 for both European and Asian options

— essentially no gain. The bridge's true role in this study is to couple fine and coarse paths at adjacent MLMC levels through shared Brownian increments, ensuring that the level correction variances decrease rapidly across the hierarchy.

Fourth, the Longstaff–Schwartz algorithm provides reliable American option pricing with a well-behaved exercise boundary. Experiment 4 produced a price of 6.0747 with a tight 95% confidence interval of $[6.0116, 6.1378]$. The estimated exercise boundary has the correct qualitative shape predicted by optimal stopping theory, and the price is stable across Bermudan approximations with $m = 10$ to $m = 200$ exercise dates, confirming rapid convergence of the Bermudan-to-American approximation.

Fifth, naive LSMC with MLMC combination fails to achieve variance decay. Experiment 5 demonstrated that independently training the LS regression at each MLMC level prevents the level correction variances from decaying, violating the key condition for MLMC efficiency. The resulting price estimate of 13.43 diverges significantly from the correct value of 6.07, confirming that a successful integration of LSMC and MLMC requires explicit coupling of the exercise strategies across levels [4].

Chapter 7

Conclusion

7.1 Summary of Contributions

This study has developed a comprehensive simulation framework for option pricing, progressing from the foundations of the Monte Carlo method to the state-of-the-art Multilevel Monte Carlo technique, with American options and Brownian bridge path construction as connecting themes throughout.

Chapter 2 established the theoretical basis of Monte Carlo simulation and demonstrated that variance reduction techniques — antithetic variates, control variates, importance sampling, and stratified sampling — can significantly improve estimation efficiency for European and Asian options. The empirical results confirmed substantial variance reduction factors, with control variates achieving a VRF of 6.93 for European call pricing and importance sampling achieving a VRF of nearly 50 for deep out-of-the-money options.

Chapter 3 addressed the early exercise problem that makes American option pricing fundamentally harder than European pricing. The Longstaff–Schwartz algorithm is presented as the standard Monte Carlo solution, replacing intractable conditional continuation values with linear regression approximations via backward induction. The algorithm is shown to produce a convergent lower bound for the option price, and the Andersen–Broadie primal-dual framework is presented as the theoretical complement that supplies an upper bound.

Chapter 4 introduced the Brownian bridge as a path construction tool whose key property is the conditional interpolation formula: given the values of a Brownian motion at two times, the intermediate value follows a Gaussian distribution with mean equal to the linear interpolation and variance proportional to the distance from both endpoints. The backward dyadic simulation algorithm is presented, and the connection to MLMC path coupling is established.

Chapter 5 developed the Multilevel Monte Carlo method from the telescoping identity, showing how the computational complexity can be reduced from $\mathcal{O}(\epsilon^{-3})$ to $\mathcal{O}(\epsilon^{-2}(\ln \epsilon)^2)$ by distributing simulations across a hierarchy of discretization levels and coupling adjacent levels through shared Brownian increments. The adaptive implementation and the

complexity theorem of [9] are presented in detail with numerical implementation [20].

Chapter 6 provided numerical evidence for all these theoretical results across five experiments covering European, Asian, and American options. The experiments confirmed the theoretical complexity advantage of MLMC in the asymptotic regime, demonstrated the role of the Brownian bridge as a coupling rather than variance reduction tool, validated the Longstaff–Schwartz algorithm through a correctly shaped exercise boundary and stable price estimates, and investigated the combination of LSMC and MLMC for American option pricing, revealing that naive independent regression at each level fails to produce the variance decay required for MLMC efficiency.

7.2 Key Findings

The principal findings of this study can be summarised as follows.

The MLMC method delivers its theoretical computational advantage of $\mathcal{O}(\epsilon^{-2}(\ln \epsilon)^2)$ versus $\mathcal{O}(\epsilon^{-3})$ most clearly for path-dependent options at high precision requirements. For moderate accuracy or path-independent payoffs with exact simulation available, the overhead of the multilevel hierarchy can outweigh the savings at a single fixed ϵ.

The choice of the coarsest discretization level is a practically important implementation decision that is not always emphasised in the theoretical literature. Starting the MLMC hierarchy at a grid fine enough to produce a reasonable coarse approximation — rather than at a single time step — is essential for achieving the efficiency gains predicted by the theory.

The Brownian bridge construction is most valuable in the context of MLMC as a mechanism for coupling fine and coarse paths through shared Brownian increments, rather than as a standalone variance reduction tool. Stratifying only the terminal Brownian motion value W_T provides negligible variance reduction for path-dependent payoffs whose distribution depends on the entire path.

The Longstaff–Schwartz algorithm provides a stable and accurate lower bound for American put option prices. The estimated exercise boundary correctly reflects the theoretical structure of the optimal stopping problem, and the Bermudan approximation converges rapidly as the number of exercise dates increases.

The naive combination of LSMC and MLMC, in which the Longstaff–Schwartz regression is trained independently at each discretization level, fails to produce the variance decay required for MLMC efficiency. The level correction variances remain large across all levels, and the resulting price estimate diverges significantly from the true value. This confirms that a successful integration of the two methods requires explicit coupling of the exercise strategies across levels [4].

7.3 Limitations and Future Directions

Several directions for future research follow naturally from this work.

The MLMC experiments in this study used the Euler–Maruyama scheme for path simulation. Higher-order schemes such as the Milstein method achieve better strong convergence rates, and adaptive timestep methods [8] can further improve stability for non-Lipschitz coefficients which translates directly into faster decay of the level correction variances and improved MLMC efficiency. Implementing the Milstein scheme would be a straightforward extension that could produce more pronounced speedups, particularly for the Asian option case.

The application of MLMC to American options remains an open research challenge. The Longstaff–Schwartz regression must be performed independently at each discretization level, and the interaction between the multilevel hierarchy and the regression-based exercise strategy introduces additional approximation errors that are not yet fully characterised theoretically. Combining the methods of Chapters 3 and 5 into a coherent multilevel American option pricer would be a natural and practically important extension. The numerical investigation in Section 6.6 confirms that naive independent regression at each level fails to produce the variance decay required for MLMC efficiency; a successful implementation requires coupling the exercise strategies across levels, as discussed theoretically in [3, 4].

The Brownian bridge variance reduction experiments suggested that stratifying only the terminal value W_T is insufficient for path-dependent payoffs. A more effective approach would combine the backward dyadic simulation with quasi-Monte Carlo sequences, assigning low-discrepancy numbers to the most important path values first. More sophisticated combinations of MLMC with sparse grids and QMC methods have been explored in [10], achieving computational complexity below $\mathcal{O}(\epsilon^{-2})$ for certain problem classes, and such extensions could provide substantially larger variance reduction factors for Asian option pricing within the MLMC framework.

Finally, the numerical experiments in this study are conducted in the standard Black–Scholes framework. Extending the MLMC and Brownian bridge methods to more realistic models — such as the Heston stochastic volatility model or jump-diffusion models — would require additional work on path coupling across levels and the treatment of discontinuous payoffs, and represents an important direction for practical application.

7.4 Closing Remarks

The Multilevel Monte Carlo method represents one of the most significant advances in computational finance of the past two decades. By distributing computational effort intelligently across a hierarchy of accuracy levels rather than concentrating it all at the finest level, it achieves near-optimal complexity for a broad class of option pricing problems. This study has presented the mathematical foundations, algorithmic details, and numerical evidence for this claim, while also clarifying the practical conditions under which the theoretical advantages materialise. The results confirm that MLMC, when implemented carefully with an appropriate choice of coarsest level and combined with the Brownian bridge path coupling, is a powerful and practically viable tool for high-precision derivative

pricing. At the same time, the numerical investigation of LSMC with MLMC highlights that extending the framework to American options requires careful attention to the coupling of exercise strategies across levels — a challenge that remains an important open direction for future research.

Bibliography

[1] L. Andersen and M. Broadie, Primal-dual simulation algorithm for pricing multidimensional American options, Management Science, 50(9), pp. 1222–1234, 2004.

[2] V. Bally and G. Pagès, A quantization algorithm for solving multidimensional discrete-time optimal stopping problems, Bernoulli, 9(6), pp. 1003–1049, 2005.

[3] D. Belomestny, Pricing Bermudan options by nonparametric regression: Optimal rates of convergence for lower estimates, Finance and Stochastics, 15(4), pp. 655–683, 2011.

[4] D. Belomestny, J. Schoenmakers, and F. Dickmann, Multilevel dual approach for pricing American style derivatives, Finance and Stochastics, 17(4), pp. 717–742, 2013.

[5] G. H. Choe, *Stochastic Analysis for Finance with Simulations*, Springer Finance, Springer, 2016.

[6] E. Clément, D. Lamberton, and P. Protter, An analysis of a least squares regression method for American option pricing, Finance and Stochastics, 6(4), pp. 449–471, 2002.

[7] K. Cliffe, M. Giles, R. Scheichl, and A. Teckentrup, Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients, Computing and Visualization in Science, 14(1), pp. 3–15, 2011.

[8] W. Fang and M. B. Giles, Adaptive Euler–Maruyama method for SDEs with non-globally Lipschitz drift, in *Monte Carlo and Quasi-Monte Carlo Methods 2016*, Springer, 2018, arXiv:1609.08101.

[9] M. B. Giles, Multilevel monte carlo path simulation, Operations Research, 56(3), pp. 607–617, 2008.

[10] M. B. Giles, F. Y. Kuo, and I. H. Sloan, Combining sparse grids, multilevel MC and QMC for elliptic PDEs with random coefficients, in *Monte Carlo and Quasi-Monte Carlo Methods 2016*, Springer, 2018, arXiv:1711.02437.

[11] A.-L. Haji-Ali and M. B. Giles, Sub-sampling and other considerations for efficient risk estimation in large portfolios, Journal of Computational Finance, 26(1), 2022, arXiv:1912.05484.

[12] T. Hironaka, M. B. Giles, T. Goda, and H. Thom, Multilevel Monte Carlo estimation of the expected value of sample information, SIAM/ASA Journal on Uncertainty Quantification, 8(3), pp. 1236–1259, 2020, arXiv:1909.00549.

[13] I. Karatzas and S. E. Shreve, *Methods of Mathematical Finance*, Springer, New York, 1998.

[14] G. Katsiolides, E. Muller, R. Scheichl, T. Shardlow, M. Giles, and D. Thomson, Multilevel Monte Carlo and improved timestepping methods in atmospheric dispersion modelling, Journal of Computational Physics, 354, pp. 320–343, 2018.

[15] R. Korn, E. Korn, and G. Kroisandt, *Monte Carlo Methods and Models in Finance and Insurance*, CRC Press, Taylor & Francis Group, Boca Raton, FL, 2010, ISBN 978-1-4200-7619-6.

[16] C. Lester, R. E. Baker, M. B. Giles, and C. A. Yates, Extending the multi-level method for the simulation of stochastic biological systems, Bulletin of Mathematical Biology, 78(8), pp. 1640–1677, 2016, arXiv:1412.4069.

[17] F. A. Longstaff and E. S. Schwartz, Valuing American options by simulation: A simple least-squares approach, The Review of Financial Studies, 14(1), pp. 113–147, 2001.

[18] L. C. G. Rogers, Monte Carlo valuation of American options, Mathematical Finance, 12(3), pp. 271–286, 2002.

[19] R. U. Seydel, *Tools for Computational Finance*, Springer, 6th edition, 2017.

[20] O. Uğur, *Introduction to Computational Finance*, Imperial College Press, London, 2009.

[21] F. Vidal-Codina, N. Nguyen, M. Giles, and J. Peraire, A model and variance reduction method for computing statistical outputs of stochastic elliptic partial differential equations, Journal of Computational Physics, 2015.