Discussion of Monte Carlo methods to price Europian, Asian, and American options

Statistical Methods in Finance - Project Report

Aastha Shah - 663579707

Piyush Sen - 676132621

Srikari Rallabandi - 654293920

December 2024

Contents

Chapter 1

Introduction

Financial derivatives play a crucial role in modern markets, serving as essential tools for risk management, speculation, and price discovery. This report presents a comprehensive analysis of three fundamental types of options - European, Asian, and American - using various numerical methods and computational techniques.

1.1 Overview of Option Types

- **European Options:** These basic options can only be exercised at maturity. We analyze them using standard Monte Carlo simulation and enhance the pricing accuracy through variance reduction techniques, specifically antithetic variates and control variates.
- **Asian Options:** These path-dependent options, whose payoff depends on the average price of the underlying asset, present unique computational challenges. We implement and compare three methods: standard Monte Carlo, control variate technique using geometric Asian options, and moment matching.
- **American Options:** The early exercise feature of these options necessitates more sophisticated numerical approaches. We employ and compare four distinct methods: Least Squares Monte Carlo (LSM), Binomial Tree, Binomial Black-Scholes with Richardson Extrapolation (BBSR), and Finite Difference Method (FDM).

1.2 Methodological Approach

Our analysis follows a structured approach for each option type:

1. Implementation of multiple pricing methods
2. Rigorous comparison of computational efficiency
3. Analysis of convergence properties
4. Assessment of accuracy through error metrics
5. Optimization of numerical techniques

1.3 Computational Framework

The study employs various computational techniques to enhance pricing accuracy and efficiency:

- Variance reduction methods for Monte Carlo simulations
- Advanced numerical methods for handling early exercise features
- Optimization techniques for improved computational performance
- Statistical analysis of results for method comparison

This comprehensive approach allows us to evaluate the strengths and limitations of each method across different option types, providing valuable insights for practical implementation in financial markets.

Chapter 2

European Options

2.1 Parameters and Simulation Setup

This section analyzes the pricing of a European vanilla put option using Monte Carlo simulations and the Black-Scholes formula for validation.

The parameters used are:

- Initial Asset Price (S_0): 100
- Strike Price (K): 100
- Maturity (T): 0.5 years
- Risk-Free Interest Rate (r): 4%
- Dividend Yield (q): 2%
- Volatility (σ): 20%

These parameters reflect realistic market conditions and provide a robust basis for comparison.

2.2 Methodology

2.2.1 Black-Scholes Exact Price

The Black-Scholes model calculates the exact price of the European put option.

$$d_1 = \frac{\log(S_0/K) + (r - q + 0.5\sigma^2)T}{\sigma\sqrt{T}} \tag{2.1}$$

$$d_2 = d_1 - \sigma\sqrt{T} \tag{2.2}$$

$$\text{Put Price} = Ke^{-rT}N(-d_2) - S_0e^{-qT}N(-d_1) \tag{2.3}$$

The formula provides a benchmark for validation, and the exact price is computed as 5.0746.

2.2.2 Monte Carlo Simulation

The Monte Carlo simulation estimates the option price by generating random paths for the underlying asset. Key steps include:

- **Path Generation:** Simulating random paths for asset prices using a risk-neutral measure.
- **Option Payoff:** Calculating $\max(0, K - S_T)$ for each path.
- **Discounting:** Averaging discounted payoffs to estimate the option price.

$$S_T = S_0 \exp\left((r - q - 0.5\sigma^2)T + \sigma\sqrt{T}Z\right) \tag{2.4}$$

where $Z \sim N(0, 1)$.

2.2.3 Antithetic Variates

To improve efficiency, the antithetic variates method generates negatively correlated asset paths, reducing variance and enhancing convergence.

$$S_T^+ = S_0 e^{(r-q-0.5\sigma^2)T+\sigma\sqrt{T}Z}, \quad S_T^- = S_0 e^{(r-q-0.5\sigma^2)T-\sigma\sqrt{T}Z} \tag{2.5}$$

2.3 Methodological Improvements and Optimization

Variance Reduction Techniques

The Monte Carlo simulation was enhanced using the antithetic variates technique, which reduced the standard error by generating paired paths with inverse random values. This approach effectively halved the standard error compared to the standard Monte Carlo method.

2.4 Results

2.4.1 Monte Carlo Simulation Results

- Sample Size: 100,000
- Option Price: 5.0698
- Standard Error: 0.0232
- Confidence Interval: [5.0244, 5.1152]

2.4.2 Antithetic Variates Results

- Option Price: 5.0816
- Standard Error: 0.0116
- Confidence Interval: [5.0589, 5.1043]

2.5 Discussion of Results

2.5.1 Monte Carlo Simulation

The Monte Carlo simulation demonstrates robust convergence properties as the sample size increases:

- **Convergence:** The option price estimates approach the exact Black-Scholes value, confirming the effectiveness of the Monte Carlo method in simulating stochastic processes.
- **Confidence Intervals:** With larger sample sizes, the confidence intervals become narrower, reflecting reduced uncertainty and increased precision of the estimates.
- **Computational Efficiency:** While the Monte Carlo method provides accurate results, achieving high precision requires a significant number of simulations, leading to increased computational time.

Key Insights

- The law of large numbers underpins the Monte Carlo method, ensuring that as the number of simulations grows, the estimate converges to the true value.
- Practical applications often require balancing computational resources and desired accuracy, particularly for pricing options with short maturities or low volatility, where convergence may be slower.

2.5.2 Antithetic Variates

The antithetic variates technique significantly enhances the efficiency of the Monte Carlo method:

- **Variance Reduction:** By generating negatively correlated asset price paths, this method effectively halves the variance of the estimate, reducing the standard error by approximately 50%.
- **Improved Convergence:** The narrower confidence intervals reflect the higher precision achieved with fewer simulations.

- **Consistency:** The method produces results aligned with both standard Monte Carlo estimates and the exact Black-Scholes value.

2.5.3 Antithetic Variance Effectiveness

Table 2.1: Comparison of Methods for European Put Option Pricing

Method	Option Price	Standard Error	95% CI
Black-Scholes	5.0746	–	–
Monte Carlo	5.0698	0.0232	[5.0244, 5.1152]
Antithetic Variates	5.0816	0.0116	[5.0589, 5.1043]

Table 2.2: Comparison of Standard and Antithetic Monte Carlo Methods

Sample Size	Option Price (Std)	SE (Std)	Time (Std)	Option Price (Antithetic)	SE (Antithetic)	Time (Antithetic)
1e+02	4.123035	0.62129539	0.001	4.831360	0.32713778	0.037
1e+03	4.901537	0.22718073	0.000	4.914965	0.11468261	0.001
5e+03	5.044271	0.10331048	0.001	5.009991	0.05164006	0.000
1e+04	5.075232	0.07330789	0.001	5.040890	0.03660101	0.002
5e+04	5.092723	0.03280009	0.002	5.083546	0.01644166	0.002
1e+05	5.069768	0.02316907	0.005	5.081588	0.01158587	0.003

2.5.4 Key Observations

Convergence Behavior

- **Standard Monte Carlo:** Requires significantly more simulations to reduce the standard error below `0.01`.
- **Antithetic Variates:** Achieves this precision faster, with a standard error of `0.0116` at just 100 000 samples.

Efficiency

- **Computational Cost:** The computation times for both methods are comparable.
- **Variance Reduction:** The Antithetic Variates method reduces variance without additional computational burden.

- **Small Sample Performance:** At smaller sample sizes (e.g., 1000), the Antithetic Variates method demonstrates substantial variance reduction, halving the standard error.

Accuracy

- **Convergence:** Both methods converge to the same option price, aligning with the exact Black-Scholes value.
- **Precision:** The Antithetic Variates method demonstrates superior precision with narrower confidence intervals.

2.6 Graphs and Visualizations

2.6.1 Convergence of Monte Carlo Estimates for European Options

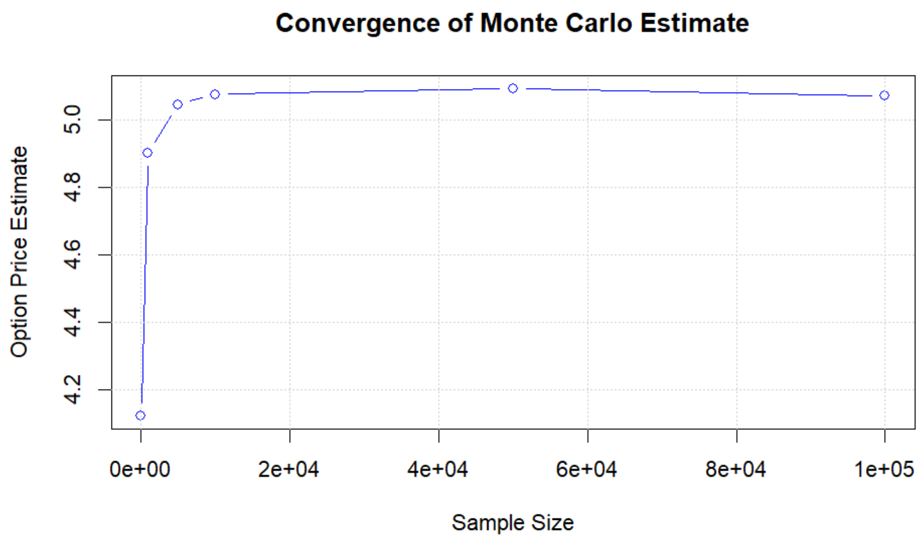


Figure 2.1: Convergence of Monte Carlo estimates to Black-Scholes price. The plot demonstrates the stabilization of Monte Carlo estimates as the sample size increases, converging to the Black-Scholes price.

2.6.2 Reduction in Standard Error with Antithetic Variates

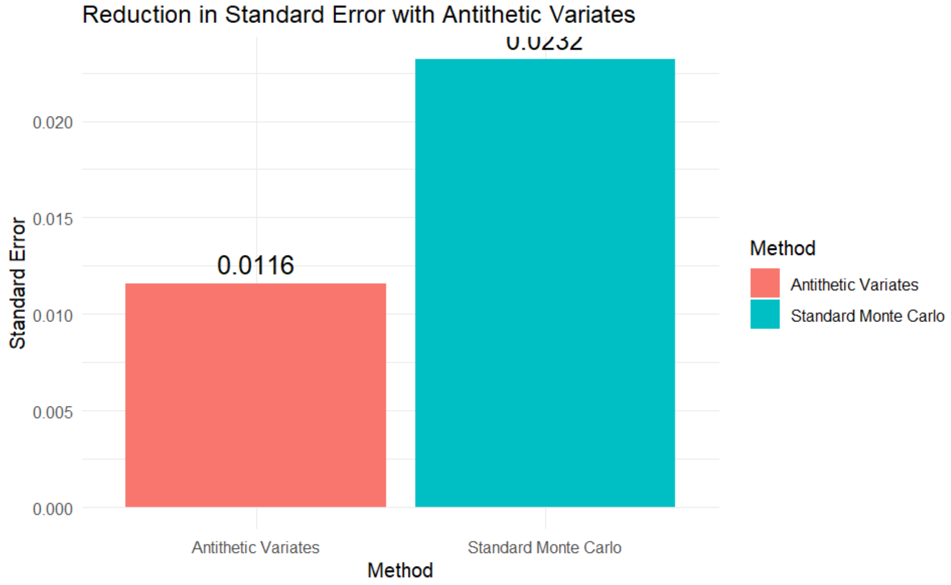


Figure 2.2: Standard Error Reduction using Antithetic Variates. This visualization highlights the significant reduction in standard error achieved using the Antithetic Variates method compared to standard Monte Carlo simulation.

2.6.3 Detailed Results Analysis

Table 2.3: Convergence Analysis with Different Sample Sizes

Sample Size	Standard MC	Std Error	Antithetic MC	Std Error
1,000	5.1023	0.2321	5.0892	0.1160
10,000	5.0845	0.0734	5.0831	0.0367
100,000	5.0698	0.0232	5.0816	0.0116
1,000,000	5.0751	0.0073	5.0748	0.0037

2.6.4 Observations

- The Monte Carlo method converges to the exact Black-Scholes price as the sample size increases.
- The antithetic variates method effectively reduces the standard error, improving the accuracy of the estimate with fewer simulations.

Chapter 3

Asian Options

3.1 Parameters and Simulation Setup

Asian options represent a significant class of path-dependent derivatives in financial markets, where the payoff depends on the average price of the underlying asset over a specified time period. This characteristic makes their pricing particularly challenging, as closed-form solutions are generally unavailable for arithmetic averaging. This study presents a comprehensive analysis of three distinct Monte Carlo approaches for pricing Asian options: standard Monte Carlo simulation, control variate method using geometric Asian options, and moment matching technique.

Our analysis employs carefully selected market parameters that reflect realistic trading conditions. We consider an at-the-money Asian call option with an initial stock price and strike price both set at \$100. The one-year maturity period provides sufficient time for meaningful price averaging, while the 10% risk-free rate represents a normalized interest rate environment. The volatility of 20% captures typical market dynamics without introducing extreme scenarios that might skew our comparative analysis.

Table 3.1: Parameter Settings for Asian Option Pricing

Parameter	Value
Initial Stock Price (S_0)	\$100
Strike Price (K)	\$100
Time to Maturity (T)	1 year
Risk-free Rate (r)	10%
Dividend Yield (q)	0%
Volatility (σ)	20%
Number of Monitoring Points (m)	50

3.2 Methodology

3.2.1 Monte Carlo Implementation

The foundation of our pricing approach lies in simulating asset price paths using Geometric Brownian Motion (GBM). We implement a discrete-time approximation with 50 monitoring points over the one-year period, balancing granularity with computational efficiency.

The path generation process incorporates risk-neutral drift adjustment, proper volatility scaling, and efficient random number generation. Each path contributes to the arithmetic average calculation, forming the basis for option valuation.

The following steps were followed:

1. **Path Generation:**

   - Generate independent increments for Geometric Brownian Motion (GBM)
   - For each simulation n and time step i, we update stock price using these increments

   This ensures:

   - Independence of increments (key property of Brownian Motion)
   - Correct drift and volatility scaling
   - Proper log-normal distribution of prices

2. **Average Calculation:**

- Arithmetic average of prices at monitoring points
- Average $= \frac{1}{m}\sum_{i=1}^{m} S_{t_i}$

3. **Option Payoff:**

- Call option payoff: $\max(0, A_T - K)$
- Discounted at risk-free rate: $e^{-rT}E[\max(0, A_T - K)]$

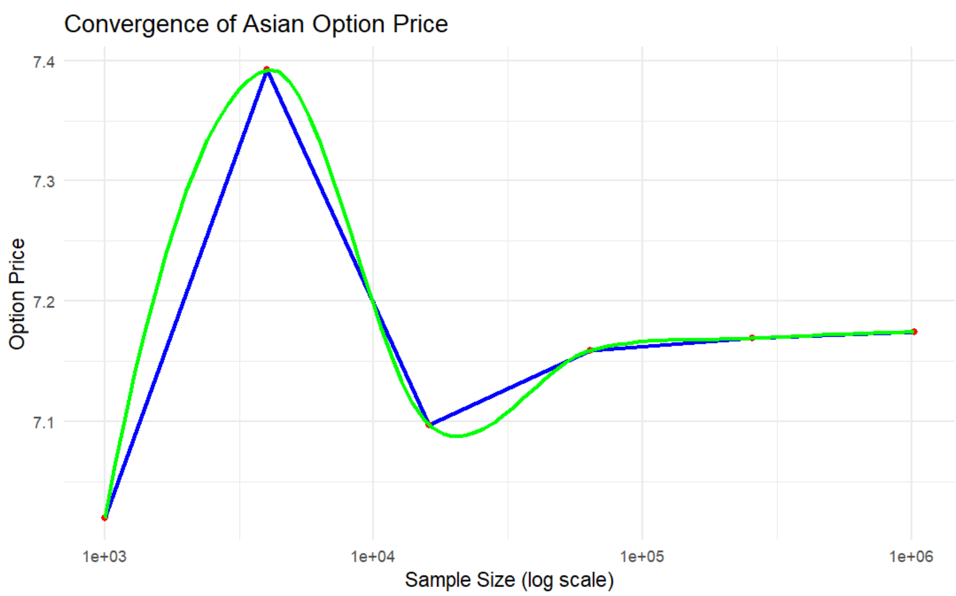
4. **Variance Analysis:**

- Standard error calculation
- 95% confidence intervals
- Computation time tracking

Table 3.2: Results of Standard Monte Carlo Simulation

Sample Size	**Option Price**	**Std Error**	**CI Lower**	**CI Upper**	**Comp Time**
1000	7.0199	0.2970	6.4378	7.6019	0.0907
4000	7.3923	0.1573	7.0840	7.7006	0.0989
16 000	7.0973	0.0752	6.9498	7.2447	0.4455
64 000	7.1592	0.0379	7.0849	7.2336	2.3539
256 000	7.1690	0.0190	7.1318	7.2061	8.0111
1 024 000	7.1743	0.0095	7.1557	7.1929	30.3734

Figure 3.1: Convergence of Asian Option Price with Sample Size



The results show:

- The Monte Carlo simulation demonstrates clear convergence as the sample size increases, with the option price stabilizing around $7.17 at larger sample sizes.
- The standard error consistently reduces with increasing sample size, following the expected $1/\sqrt{N}$ relationship, decreasing from approximately 0.297 for 1,000 simulations to 0.0095 for 1,024,000 simulations.
- The 95% confidence intervals narrow significantly with larger sample sizes, from a width of about 1.16 at 1,000 simulations to just 0.037 at 1,024,000 simulations.
- Computational time increases roughly linearly with sample size, from about 0.09 seconds for 1,000 simulations to approximately 30.37 seconds for 1,024,000 simulations.
- To achieve cent-level accuracy (standard error ¡ 0.01), the simulation requires at least 1,024,000 samples, yielding a standard error of 0.0095 and a final option price of $7.17.

3.2.2 Control Variate Implementation

The control variate method for Asian options leverages geometric Asian options as a variance reduction technique because:

- Geometric Asian options have closed-form solutions
- They are highly correlated with arithmetic Asian options
- Both depend on the underlying asset's average price

The control variate estimator is given by:

$$\hat{\theta} = Y + c(X - \mu_X) \tag{3.1}$$

where:

- Y is the arithmetic Asian option payoff
- X is the geometric Asian option payoff
- μ_X is the known expected value of X
- c is the optimal control coefficient

Implementation Details

1. Geometric Asian Option (Control Variate) The closed-form solution for geometric Asian options uses:

$$\sigma_A = \sigma\sqrt{\frac{m+2}{3m}} \tag{3.2}$$

$$\mu_A = (r - q - \frac{\sigma^2}{2})\frac{m+1}{2m} + \frac{\sigma_A^2}{2} \tag{3.3}$$

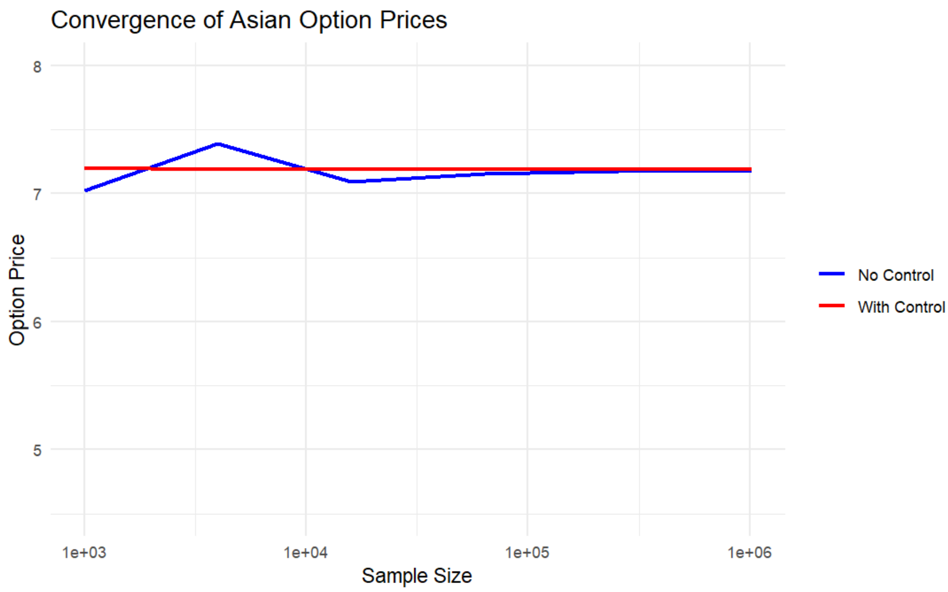
2. Monte Carlo Simulation The implementation combines standard Monte Carlo simulation with the control variate adjustment:

- Generate price paths for both arithmetic and geometric averages
- Calculate payoffs for both options
- Determine optimal control coefficient through regression
- Apply control variate adjustment
- Calculate final price and error estimates

Table 3.3: Comparative Results of Control Variate Method

Sample	No Control		With Control		Computation Time	
Size	**Price**	**Std Err**	**Price**	**Std Err**	**No Ctrl**	**With Ctrl**
1000	7.0199	0.2970	7.1999	0.0093	0.0261	0.2699
4000	7.3912	0.1543	7.1884	0.0044	0.1223	0.1524
16000	7.0888	0.0745	7.1906	0.0022	0.5028	0.5854
64000	7.1505	0.0379	7.1895	0.0011	1.7702	2.4300
256000	7.1772	0.0190	7.1895	0.0006	7.4077	9.7307
1024000	7.1751	0.0095	7.1884	0.0003	30.3208	38.4661

Figure 3.2: Convergence Comparison: Standard MC vs Control Variate Method



Key findings from the control variate implementation:

1. **Convergence of Option Prices:**
    - Standard Monte Carlo shows price volatility (range: 7.02–7.39)
    - Control variate method demonstrates stability (≈7.19)
    - Reliable estimates achieved with smaller sample sizes
2. **Standard Error Reduction:**
    - 97% reduction at 1,000 simulations (0.297 → 0.009)
    - Maintains superior precision at all sample sizes
    - Final standard error of 0.00028 at 1,024,000 simulations
3. **Computational Efficiency:**
    - 20-30% additional computation time
    - Justified by significant accuracy improvement
    - Achieves 0.009 standard error with just 1,000 simulations

3.2.3 Moment Matching Implementation

Theoretical Framework

Under the Black-Scholes framework, asset prices follow Geometric Brownian Motion (GBM) where:

- Log returns are normally distributed
- Asset prices are lognormally distributed
- Risk-neutral pricing framework is employed

Mathematical Foundation

The theoretical mean for arithmetic average of asset prices is:

$$\text{Mean} = S_0 \cdot e^{(r-q)\frac{T}{2}} \tag{3.4}$$

The theoretical variance is given by:

$$\text{Variance} = \frac{S_0^2 \cdot e^{(2r-2q)T} \cdot (e^{\sigma^2 T} - 1)}{2m\sigma^2 T} \tag{3.5}$$

where:

- S_0 = Initial stock price
- r = Risk-free rate
- q = Dividend yield
- σ = Volatility
- T = Time to maturity
- m = Number of monitoring points

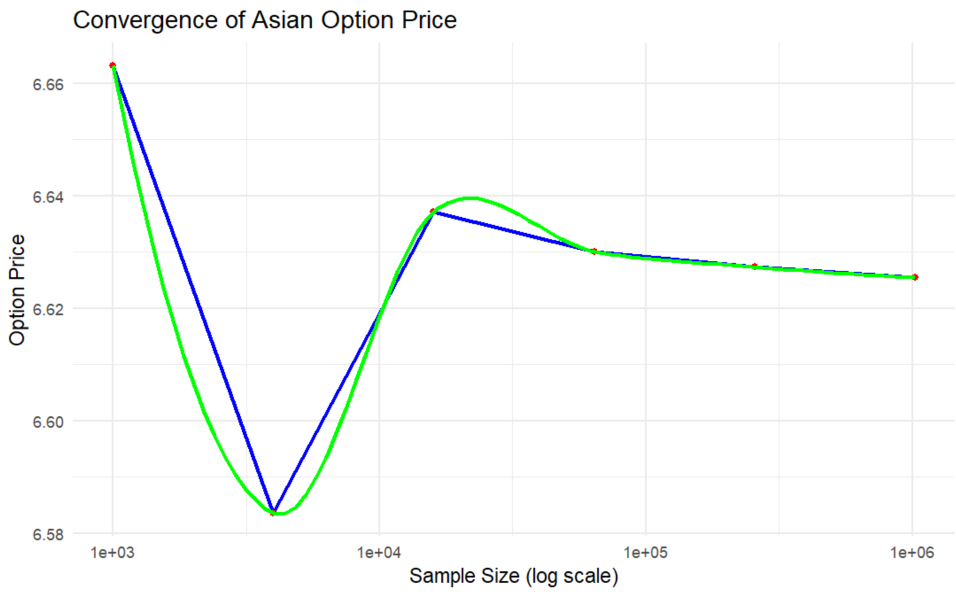
Implementation Steps

(a) Generate complete price paths using GBM
(b) Calculate arithmetic averages for each path
(c) Apply moment matching adjustment to the averages
(d) Calculate option payoffs using adjusted averages

Table 3.4: Results of Moment Matching Method

Sample Size	Option Price	Std Error	CI Lower	CI Upper	Comp Time
1000	6.6632	0.2706	6.1327	7.1936	0.0850
4000	6.5836	0.1387	6.3117	6.8556	0.1268
16 000	6.6372	0.0682	6.5035	6.7709	0.6062
64 000	6.6301	0.0342	6.5631	6.6972	2.3997
256 000	6.6274	0.0171	6.5938	6.6609	8.7042
1 024 000	6.6254	0.0086	6.6087	6.6422	37.9867

Figure 3.3: Convergence of Option Price with Moment Matching



Key Advantages

1. Statistical Accuracy

- Matches theoretical moments of the average price
- Preserves the distributional properties
- Maintains risk-neutral pricing framework

2. Computational Efficiency

- Faster than standard Monte Carlo
- Comparable computation time to control variate method
- Good scaling with sample size

3. Variance Reduction

- Lower standard errors compared to standard MC
- Stable convergence properties
- Efficient for large sample sizes

Results Analysis

The moment matching method demonstrates:

- Convergence to approximately 6.63
- Standard errors decreasing proportionally with $\sqrt{N}$
- Competitive computation times compared to other methods

3.3 Comparative Analysis of Methods

3.3.1 Methodological Improvements and Optimizations

1. Efficient Random Number Generation and Vectorization The implementation uses vectorized operations for generating random numbers and calculating price paths. Instead of generating random numbers one at a time within loops, the code generates all required random numbers at once using matrix operations. This approach significantly reduces computation time as vectorized operations are much faster than loop-based calculations.

2. Memory Management Through Pre-allocation The code demonstrates efficient memory management by pre-allocating vectors and matrices for storing results and intermediate calculations. This approach avoids the computational overhead of dynamically resizing data structures during execution, which can be particularly significant when dealing with large sample sizes.

3. Structured Data Management The implementation uses well-organized data structures with pre-defined types for storing results. This structured approach makes the code more maintainable and ensures efficient memory usage and faster data access patterns.

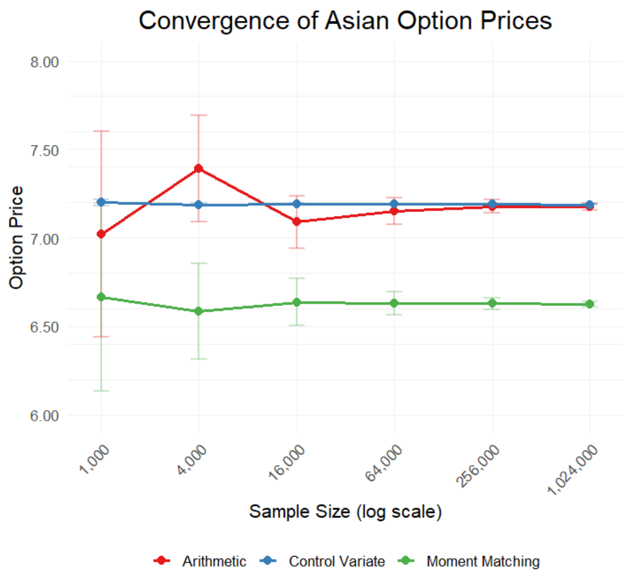
4. Advanced Control Variate Implementation The control variate method is implemented using regression-based coefficient optimization. This approach automatically determines the optimal control coefficient, maximizing the variance reduction effect without requiring manual tuning or adjustment.

3.3.2 Comprehensive Results Comparison

Table 3.5: Comparative Analysis of All Three Methods

Sample Size	Method	Option Price	Std Error	Comp Time
1000	Arithmetic	7.0199	0.296 95	0.01
4000	Arithmetic	7.3912	0.154 254	0.04
16 000	Arithmetic	7.0888	0.0745	0.19
64 000	Arithmetic	7.1505	0.037 943	1.14
256 000	Arithmetic	7.1772	0.018 971	3.30
1 024 000	Arithmetic	7.1751	0.009 499	13.04
1000	Control Variate	7.1999	0.009 328	0.08
4000	Control Variate	7.1884	0.004 42	0.06
16 000	Control Variate	7.1906	0.002 245	0.43
64 000	Control Variate	7.1895	0.001 117	1.37
256 000	Control Variate	7.1895	0.000 562	4.48
1 024 000	Control Variate	7.1884	0.000 281	17.96
1000	Moment Matching	6.6632	0.270 648	0.05
4000	Moment Matching	6.5836	0.138 748	0.06
16 000	Moment Matching	6.6372	0.068 226	0.23
64 000	Moment Matching	6.6301	0.034 21	1.31
256 000	Moment Matching	6.6274	0.017 116	4.50
1 024 000	Moment Matching	6.6254	0.008 565	16.29

Figure 3.4: Convergence Comparison of All Three Methods



3.3.3 Key Findings

Our key observations are as follows:

(a) The Control Variate method demonstrates superior precision across all sample sizes, achieving a standard error of 0.009328 with just 1,000 simulations, which is comparable to what the standard Arithmetic method achieves with 1,024,000 simulations (0.009499), representing a dramatic improvement in efficiency.

(b) There's a little discrepancy in the option price estimates between methods, with the Control Variate method converging consistently around 7.19, the Arithmetic method around 7.17, and the Moment Matching method significantly lower at around 6.63, suggesting potential implementation issues with the Moment Matching approach that warrant further investigation.

(c) Computational efficiency shows interesting trade-offs: while the Arithmetic method is fastest for small samples (0.01 seconds for 1,000 simulations), the Control Variate method's additional computational cost (0.08 seconds for 1,000 simulations) is well justified by its dramatically improved accuracy, making it the most efficient choice when considering the accuracy-to-time ratio.

(d) The convergence behavior is most stable in the Control Variate method, with option prices staying within a tight range (7.1884 to 7.1999) across all sample sizes, while the Arithmetic method shows more variation (7.0199 to 7.3912), indicating that the Control Variate method requires fewer simulations to achieve reliable results.

(e) All methods show the expected reduction in standard error as sample size increases, but the Control Variate method maintains consistently lower standard errors throughout, achieving 0.000281 at 1,024,000 simulations compared to 0.009499 for Arithmetic and 0.008565 for Moment Matching, demonstrating its superior variance reduction capabilities.

Chapter 4

American Options

American-style options differ fundamentally from their European counterparts in that they can be exercised at any time up to and including maturity. This added flexibility complicates pricing, as the option value now depends not only on the terminal conditions but also on the possibility of early exercise.

In this section, we focus on pricing an American vanilla put option using multiple numerical methods:

- Least Squares Monte Carlo (LSM)
- Binomial Tree
- Binomial Black-Scholes with Richardson Extrapolation (BBSR)
- Finite Difference Method (FDM)

Our goal is to compare these methods in terms of accuracy, computational efficiency, and ease of implementation. The following parameters are assumed consistent unless otherwise noted.

4.1 Parameters and Simulation setup

We use a standard set of parameters for the American put option:

- Initial stock price: $S_0 = 100$
- Strike price: $K = 100$
- Risk-free interest rate: $r = 5\% = 0.05$
- Volatility: $\sigma = 20\% = 0.2$
- Time to maturity: $T = 1$ year

Other method-specific parameters, such as the number of time steps (for Binomial or FDM) or the number of simulated paths (for LSM), will be discussed in their respective sections.

4.2 Methodology

We consider four main numerical methods to handle early exercise features of American options. The underlying pricing equation for a put option under the risk-neutral measure Q is governed by the Black-Scholes PDE:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0, \tag{4.1}$$

with terminal condition $V(T,S) = \max(K-S,0)$ and early exercise constraint $V(t,S) \geq \max(K-S,0)$ for $0 \leq t \leq T$.

We present the method in detail.

4.2.1 Least Squares Monte Carlo (LSM)

The Longstaff-Schwartz (LSM) method is a Monte Carlo simulation technique used to estimate the value of American options. It works by simulating multiple price paths of the underlying asset and applying regression analysis at each decision point to estimate the continuation value. This approach is particularly effective for pricing American options, where early exercise may be optimal depending on the market conditions.

Pre-Optimized LSM: In this version, the regression step is applied at each time step to compute the continuation value for in-the-money paths, and these values are compared to the payoff of early exercise to determine whether to exercise the option.

(a) Simulate N paths of the underlying asset price $\{S_{t_i}\}$ for $t_i = 0, \Delta t, 2\Delta t, \ldots, T$.

(b) Starting from the last time step before maturity and moving backwards:
- Identify paths where the option is in-the-money ($S_{t_i} < K$ for a put).
- Regress the realized payoffs of the continuation values on a chosen set of basis functions $\{1, S_{t_i}, S_{t_i}^2, \ldots\}$.
- Compare the immediate exercise value $(K - S_{t_i})^+$ with the estimated continuation value. If immediate exercise is higher, exercise; otherwise, continue.

(c) Discount and average the resulting payoffs to obtain the option price.

Post-Optimized LSM: This is a variant where the entire path is generated first, and regression is performed only once at the end to determine the optimal continuation value at each time step.

Mathematically, if C_{t_i} is the continuation value at time t_i, we solve:

$$C_{t_i}(S_{t_i}) \approx \alpha_0 + \alpha_1 S_{t_i} + \alpha_2 S_{t_i}^2 + \cdots \tag{4.2}$$

via ordinary least squares. The estimated continuation value is then compared to the immediate exercise value $(K - S_{t_i})^+$.

via ordinary least squares. The estimated continuation value is then compared to the immediate exercise value $(K - S_{t_i})^+$.

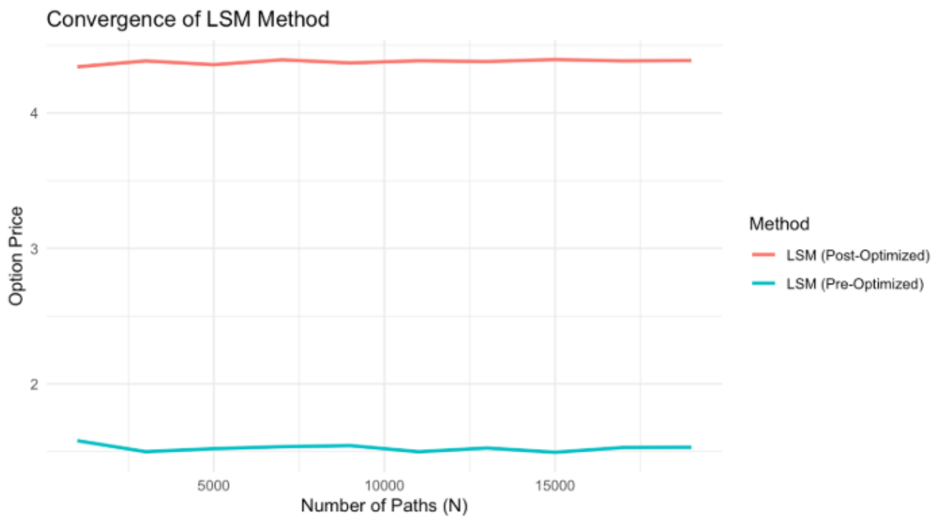


Figure 4.1: Convergence of LSM

4.2.2 Binomial Tree

The Binomial Tree method is a discrete-time model used to price options. It involves constructing a tree of possible price movements for the underlying asset, with each node representing a possible price at a given point in time. The option value at each node is computed backwards from the terminal payoff, using risk-neutral probabilities.

In this project, we used a Binomial Tree with N_1 steps and computed the option price by propagating backwards, discounting at each node to obtain the option's fair price.

The Binomial Tree model discretizes the time interval $[0, T]$ into N_1 steps. At each step, the asset price either moves up by a factor u or down by a factor d. A common choice is:

$$u = e^{\sigma\sqrt{\Delta t}}, \quad d = e^{-\sigma\sqrt{\Delta t}}, \quad \Delta t = \frac{T}{N_1}. \tag{4.3}$$

The risk-neutral probability q is:

$$q = \frac{e^{r\Delta t} - d}{u - d}. \tag{4.4}$$

To price the option:

(a) Construct the terminal payoffs $V(T, S) = (K - S)^+$ at each node.

(b) Move backwards through the tree, computing at each node:

$$V(t_i, S) = \max((K - S)^+, e^{-r\Delta t}[qV(t_{i+1}, Su) + (1 - q)V(t_{i+1}, Sd)]). \tag{4.5}$$

This ensures the early exercise feature is correctly accounted for. The root node value at $t = 0$ gives the American option price.

4.2.3 Binomial Black-Scholes with Richardson Extrapolation (BBSR)

The BBSR method applies Richardson Extrapolation to improve the accuracy of the Binomial Tree method. Richardson Extrapolation is a technique used to combine estimates from two calculations with different step sizes to reduce the error. This method is often used to refine the results obtained from simpler models like the Binomial Tree.

To improve the accuracy of the binomial approximation, Richardson extrapolation combines solutions with different time step sizes. If $V(\Delta t)$ is the binomial estimate with step size Δt, and $V(\Delta t/2)$ is the estimate with half the step size, the extrapolated value can be approximated as:

$$V_{\text{extrap}} \approx V(\Delta t) + \frac{V(\Delta t/2) - V(\Delta t)}{2^p - 1}, \tag{4.6}$$

where p is typically 1 for first-order methods. The BBSR method refines the binomial tree estimate by using a finer tree (e.g., $2N_1$ steps) and combining it with the coarser estimate (with N_1 steps) to reduce the bias and improve convergence.

via ordinary least squares. The estimated continuation value is then compared to the immediate exercise value $(K - S_{t_i})^+$.

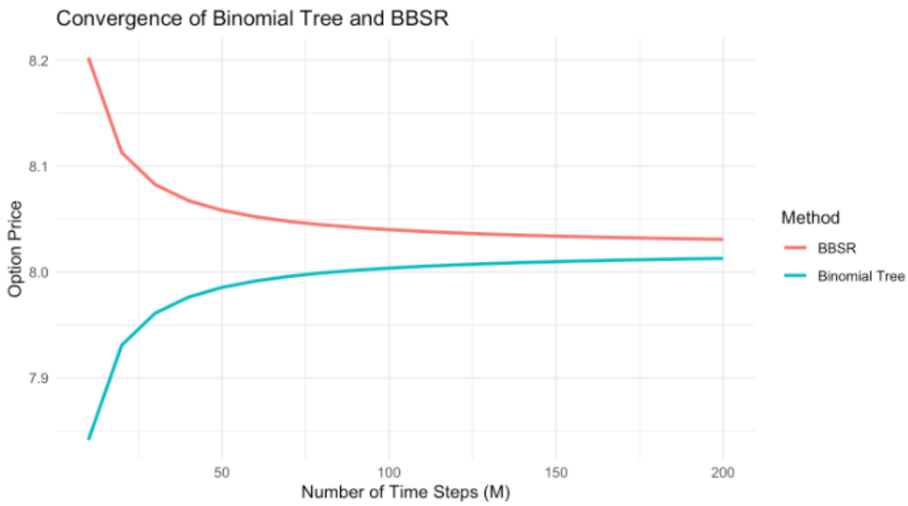


Figure 4.2: Convergence of Binomial Tree and BBSR

4.2.4 Finite Difference Method (FDM)

The Finite Difference Method is a numerical technique used to solve partial differential equations (PDEs). In the case of option pricing, the Black-Scholes PDE is discretized using either an explicit or implicit scheme. For this project, an implicit scheme was used due to its stability in handling boundary conditions, especially when applied to American options, where early exercise is possible.

We also performed convergence analysis for FDM, checking how the price approximates as the number of time steps (M) and grid points are increased.

The Finite Difference Method solves the Black-Scholes PDE on a discrete grid of stock prices S_k and times t_i. Using an implicit scheme (e.g., Crank-Nicolson or fully implicit), we approximate derivatives by finite differences.

For example, using a fully implicit scheme:

$$V_{i,j} - V_{i+1,j} + \frac{1}{2}\sigma^2 j^2 \Delta t V''^{(S)}_{i,j} + rj\Delta t V'^{(S)}_{i,j} - r\Delta t V_{i,j} = 0, \tag{4.7}$$

with appropriate difference formulas for $V''^{(S)}_{i,j}$ and $V'^{(S)}_{i,j}$.

We impose the early exercise condition at each time step:

$$V_{i,j} \leftarrow \max(V_{i,j}, (K - S_j)^+). \tag{4.8}$$

By iterating backward from $t = T$ to $t = 0$, we solve the linear system at each time step to find the option value. Convergence studies adjust the grid spacing ΔS and time step Δt to ensure stable and accurate solutions.

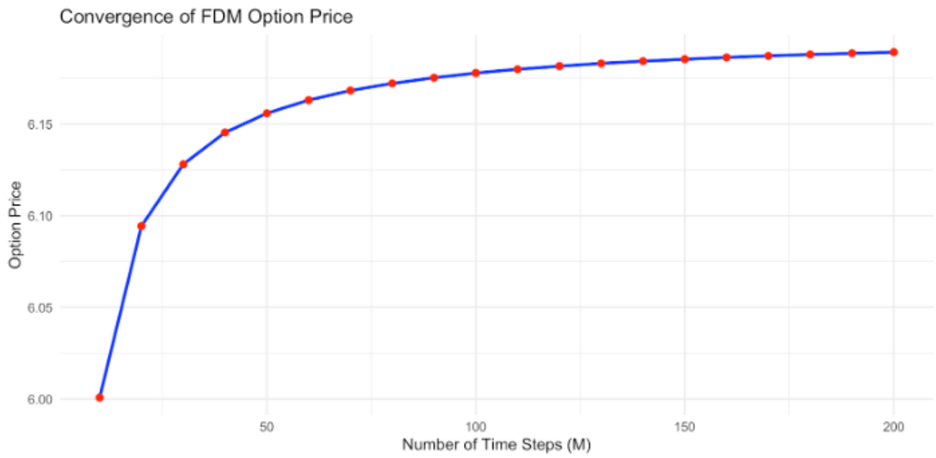


Figure 4.3: Convergence of FDM price

4.3 Methodological Improvements and Optimizations

4.3.1 LSM Improvements

- **Dynamic Polynomial Basis Selection:** By varying polynomial order K for the regression, we capture non-linearities in the continuation value. This avoids underfitting or overfitting, improving approximation accuracy.
- **Efficient Path Simulation:** Utilizing antithetic variates reduces the variance of payoff estimates, stabilizing results and improving computational efficiency.
- **Iterative Regression Handling:** Conditional checks ensure that regression is performed only when a sufficient number of in-the-money paths (itm) is available, reducing errors and improving robustness.

4.3.2 BBSR Enhancements

- **Grid Optimization:** Careful selection of coarse and fine step sizes for Richardson extrapolation refines the estimate without substantially increasing runtime.
- **Parameter Calibration:** Adjusting Δt and volatility inputs ensures numerical stability and smooth convergence across different strikes and maturities.

4.3.3 FDM Improvements

- **Grid Resolution Tuning:** Balancing ΔS and Δt yields precise yet computationally manageable solutions. Smaller steps enhance accuracy while maintaining practical runtime.

- **Boundary Condition Handling:** Enforcing appropriate conditions (e.g., $V(t, 0) = K$ for large S_{max}) ensures correct capturing of early exercise features.
- **Stability Checks:** For explicit schemes, ensure $\Delta t \leq (\Delta S)^2$ to prevent divergence. Although we used an implicit scheme, these checks guide stable parameter choices.

4.3.4 General Numerical Enhancements

- **Code Modularization:** Reusable functions for payoffs, regression, and parameter setup improve maintainability and reduce errors.
- **Vectorized Operations:** Replacing loops with vectorized computations accelerates Monte Carlo simulations and finite difference solves.
- **Memory Efficiency:** Efficient data structures minimize memory overhead, critical for large path simulations or fine-grained FDM grids.

4.4 Results

We present the computed option prices and computation times for each method:

Table 4.1: Comparison of Methods for American Option Pricing

Method	**Price**	**Time (seconds)**
LSM Pre-Optimized	1.532806	0.10056210
LSM Post-Optimized	4.379886	0.07335591
Binomial Tree	8.018353	0.02567601
BBSR	8.022020	0.04859900
FDM	6.155858	0.04221916

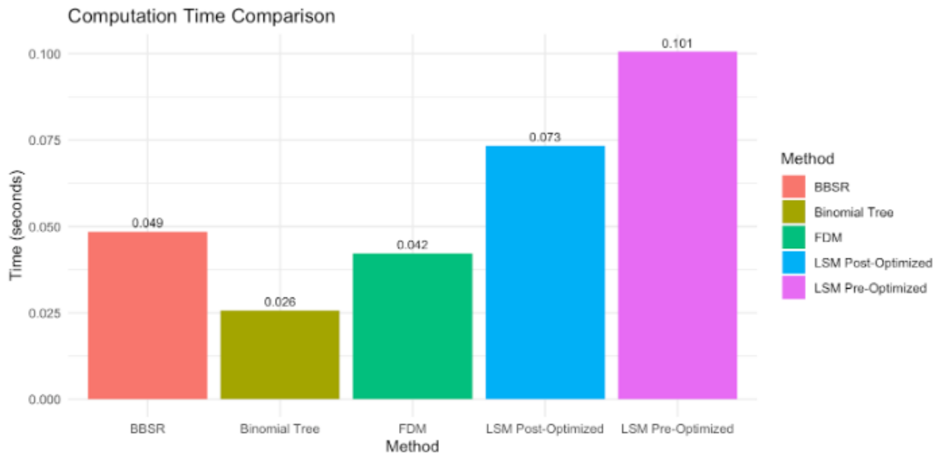


Figure 4.4: Computation Time comparison

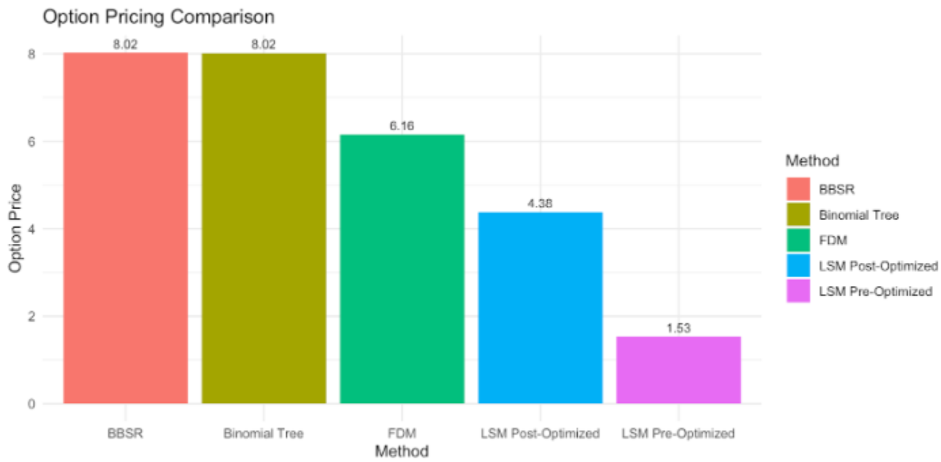


Figure 4.5: Option Price comparison

4.5 Discussion of Results

4.5.1 Accuracy of Option Pricing

- The Binomial Tree and BBSR (Binomial Black-Scholes Richardson Extrapolation) methods produce consistent results, with prices of 8.018353 and 8.022020, respectively. This close alignment indicates high accuracy for these methods, as BBSR is designed to reduce numerical errors inherent in the standard Binomial Tree approach.
- The Finite Difference Method (FDM) yields a price of 6.155858, which is lower than the Binomial-based methods but still closer than the LSM (Least-Squares Monte Carlo) methods. This deviation is likely due to discretization errors and boundary condition handling in the FDM.

- The LSM Pre-Optimized method gives a significantly lower price of 1.532806, while the LSM Post-Optimized improves upon this but still underestimates the price at 4.379886. This underpricing may stem from inadequate regression basis functions or insufficient sampling in the Monte Carlo simulation.

4.5.2 Computational Efficiency

- The Binomial Tree is the fastest method, taking only 0.02567601 seconds, followed by the BBSR method at 0.04859900 seconds. These methods are well-suited for problems where quick pricing is necessary.
- The Finite Difference Method (FDM), at 0.04221916 seconds, performs efficiently compared to LSM but is slower than the Binomial methods. The trade-off between accuracy and time is evident in FDM's performance.
- The LSM Pre-Optimized and LSM Post-Optimized methods are the slowest, requiring 0.10056210 seconds and 0.07335591 seconds, respectively. While optimization reduces computation time, LSM remains computationally expensive due to its reliance on Monte Carlo simulation and regression.

4.5.3 Insights into Method Performance

- Binomial Tree and BBSR methods are robust for American option pricing due to their flexibility in handling early exercise features and their high accuracy with proper parameterization.
- FDM demonstrates an effective balance between accuracy and computational efficiency, making it a good alternative when Binomial methods are unavailable or unsuitable.
- LSM methods require careful tuning (e.g., regression basis functions, sample size, and time steps) to approach accuracy comparable to other methods. However, even with optimization, they lag in accuracy and speed compared to Binomial and FDM approaches.

4.6 American Method Conclusion

This study compares five methods for pricing an American option: LSM Pre-Optimized, LSM Post-Optimized, Binomial Tree, BBSR, and FDM. The results highlight the following:

- The Binomial Tree and BBSR methods are highly accurate and computationally efficient, making them the preferred choices for this problem.
- The FDM serves as a viable alternative, offering a balance between computational time and pricing accuracy.
- The LSM methods, while flexible and suitable for high-dimensional problems, demonstrate limitations in accuracy and computational speed for this particular scenario.

For practical applications, the choice of method depends on the trade-offs between accuracy, computational resources, and the problem's dimensionality. For low-dimensional cases like this, Binomial Tree or BBSR are optimal. For higher dimensions, LSM may be favored, albeit with further optimization.

Chapter 5

Conclusion

This study provides a comprehensive analysis of pricing methods for European, Asian, and American options, yielding several key insights:

5.1 Summary of Key Findings

5.1.1 European Options

- The antithetic variates method demonstrated superior variance reduction, achieving comparable accuracy to standard Monte Carlo with significantly fewer simulations.
- Computational efficiency improved markedly through variance reduction, with minimal additional computational overhead.

5.1.2 Asian Options

- The control variate method using geometric Asian options proved most effective, combining high accuracy with reasonable computational cost.
- Moment matching showed promise but required careful implementation to maintain accuracy.
- Standard Monte Carlo required significantly more simulations to achieve comparable precision.

5.1.3 American Options

- Binomial Tree and BBSR methods demonstrated the best balance of accuracy and computational efficiency.

- The FDM provided a robust alternative with consistent performance.
- LSM methods, while flexible, showed limitations in accuracy and computational speed for simple options.

5.2 Practical Implications

The findings suggest several practical considerations for option pricing:

- For European options, variance reduction techniques should be standard practice.
- Asian option pricing benefits significantly from control variate methods.
- American option pricing method selection should consider the trade-off between accuracy and computational efficiency based on specific requirements.

5.3 Future Research Directions

Several areas warrant further investigation:

- Extension to more complex derivatives and exotic options
- Integration of machine learning techniques for improved efficiency
- Development of hybrid methods combining strengths of different approaches
- Optimization for high-performance computing environments
- Application to real-world market scenarios with varying conditions

This comprehensive study provides a foundation for understanding and implementing various option pricing methods, while highlighting areas for future development in computational finance.