IE 522 Statistical Methods in Finance Project

Pricing Options Using Monte Carlo

Chong Shen, Heyang Cui, Junru Wang, Yaoxi Tan

Abstract: In this paper, we price European, American and Asian options using Monte Carlo method, and apply different variance reduction techniques to improve the accuracy of simulation. We further analyze the effectiveness among these approaches. Additionally, we compare the computation time of R and C++, then accelerate the simulation using the vectorization.

Keyword: Option Pricing, Black Sholes Merton Model, Monte Carlo, Antithetic, Control Variate, Moment Matching, Least Squares, Binomial, Richardson Extrapolation

Teamwork Contributions: Chong Shen: Q1, Q2 using R, report of Q2

Heyang Cui: Q1, Q2 using C++, report of Q1, Q2

Junru Wang: Q2, Q3 using C++ , report of Q1

Yaoxi Tan: Q1, Q3 using C++ and R, report of Q3

1.European Vanilla Put Option Pricing with Monte Carlo Simulation

1.1. Monte Carlo Simulation under the Black-Scholes-Merton Model

We aim to price a European vanilla put option using Monte Carlo simulation under the Black-Scholes-Merton model. The underlying asset price St follows the stochastic differential equation:

$$S_t = S_0 \exp\left((r - q - \frac{1}{2}\sigma^2)t + \sigma B_t\right), \quad 0 \le t \le T,$$

where:

- S_0: Initial asset price
- K: Strike price
- T: Maturity (in years)
- r: Risk-free interest rate (continuous compounding)
- q: Continuous dividend yield
- σ: Volatility (annualized)
- B_t: Standard Brownian motion

Inputs:

- Initial asset price: S0 =100
- Strike price: K = 100
- Time to maturity: T = 1 year
- Risk-free interest rate: r = 0.04
- Dividend yield: q = 0.02
- Volatility: σ = 20%
- Number of time steps for averaging: m = [50, 100, 1000, 5000, 10000]

Outputs:

- Monte Carlo option price.
- Estimated standard error:

$$\mathrm{SE} = \frac{\sigma_{\mathrm{payoff}}}{\sqrt{N}},$$

- CI = [Price – 1.96 *SE, Price + 1.96 *SE]

- Absolute pricing error (using the exact Black-Scholes formula for comparison).
- Computational time.

1.2. Result Analysis

1. Accuracy of the Estimated Price:
   - Both C++ and R produce estimates that converge as the sample size increases.
   - For large sample sizes (e.g., N=1,000,000N = 1,000,000N=1,000,000), the estimated prices are very close:
     - C++: 5.064735.064735.06473
     - R: 5.0726885.0726885.072688
2. Computation Time:
   - C++ is significantly faster than R, especially as the sample size increases:
     - At N=1,000,000N = 1,000,000N=1,000,000:
       - C++: 0.0527 seconds
       - R: 1.2923 seconds
   - The difference arises because C++ is compiled and optimized for performance, while R is interpreted and less efficient for intensive computations.

Table 1: Output Result using C++

Sample size	Estimated Price	Standard Error	95% Confidence Interval	Computation Time (s)
10000	4.96025	0.0708442	[4.8214, 5.0991]	0.000411
100000	5.05823	0.0273966	[5.01384, 5.10263]	0.0073321
1000000	5.06473	0.0071800	[5.05065, 5.0788]	0.0526986

Table2: Output Result using R

Sample size	Estimated Price	Standard Error	95% Confidence Interval	Computation Time (s)
10000	5.116893	0.02285328	[5.07211, 5.16168]	0.02213
100000	5.075539	0.00719449	[5.06148, 5.08964]	0.193411

1000000	5.072688	0.00227203	[5.06823, 5.07714]	1.292304

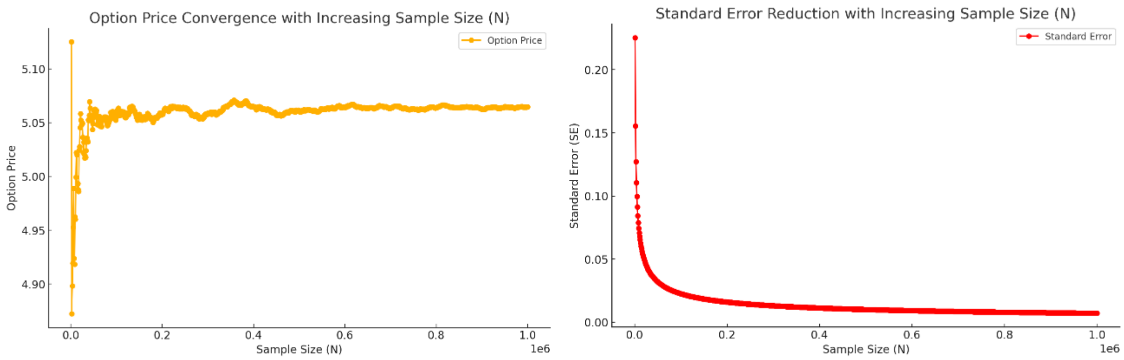

Figure 1: Option price against Samples Size(N)

1.3. Monte Carlo Simulation in the Antithetic Variates Method

The Antithetic Variates method is a variance reduction technique used in Monte Carlo simulations to improve the efficiency and accuracy of the estimates. It is particularly useful for pricing financial options, where simulations can suffer from high variance and require a large number of samples for convergence.

The idea of **Antithetic Variates** is to reduce variance by leveraging the symmetry of random variables. For a standard normal random variable Z∼N(0,1), we simulate two paths:

Original Path: Using Z.

Antithetic Path: Using −Z.

The payoff values from these two paths are averaged to obtain a more stable estimate of the expected value. Since Z and −Z are symmetric and have the same distribution, the expectation remains unchanged, but the variance is often reduced.

For a payoff function g(ST), where ST is the terminal stock price, the estimate using the Antithetic Variates method is:

$$\hat{p} = \frac{1}{2N} \sum_{i=1}^{N} \left[g(S_T(Z_i)) + g(S_T(-Z_i)) \right],$$

1.4 Comparison with Antithetic Variates Method

Table 3: Output Result using C++ and Antithetic

Sample size	Estimated Price	Standard Error	95% Confidence Interval	Computation Time (s)
10000	5.09832	0.036227	[5.027313, 5.169328]	0.00011
100000	5.0917	0.011401	[5.069353, 5.114048]	0.03122
1000000	5.07962	0.0035994	[5.072572, 5.086681]	0.172200

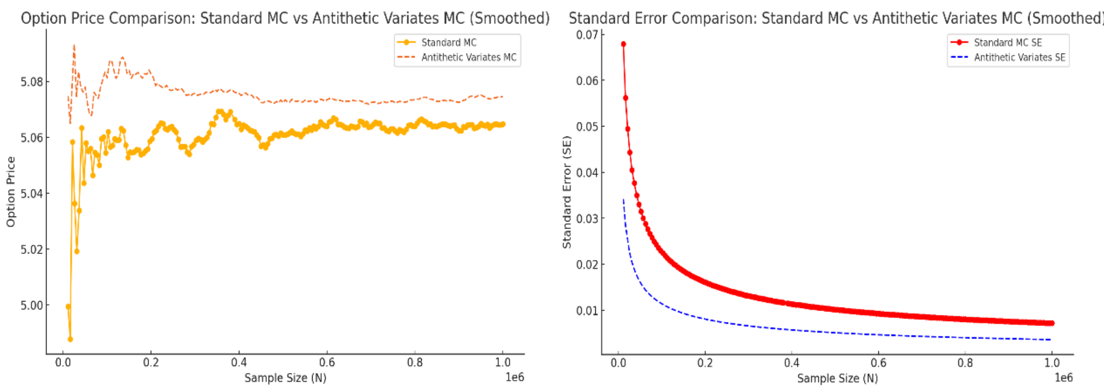

Figure 2: Comparison with Antithetic Variates Method

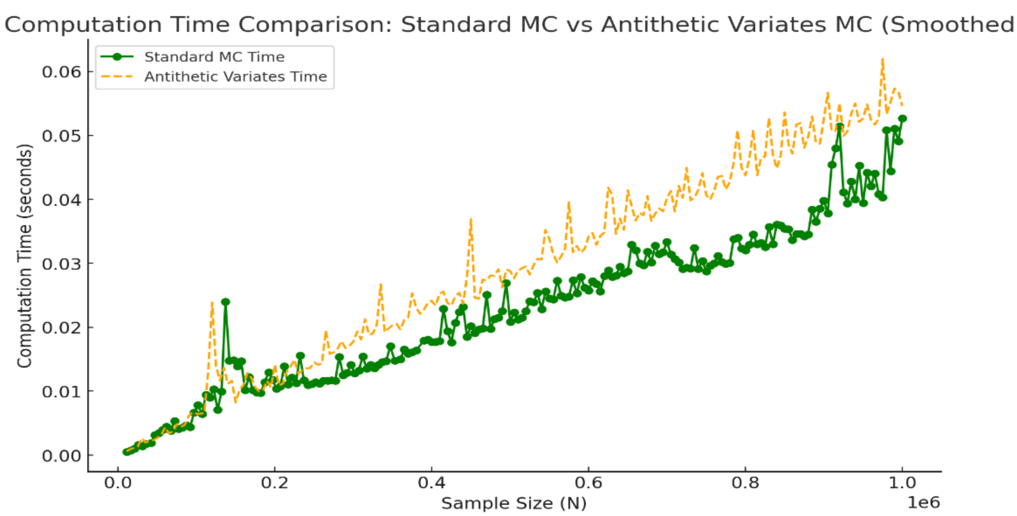

Figure 3: Comparison with Antithetic Variates Method

By removing outliers and reducing sample points, the comparison between Standard Monte Carlo and Antithetic Variates methods is now clearer and smoother.

Option Price:

- Both methods converge to a similar option price as the sample size N increases.
- The Antithetic Variates method (dashed line) stabilizes faster than the Standard Monte Carlo.

Standard Error:

- The Antithetic Variates method achieves consistently lower standard error compared to the Standard Monte Carlo method for the same sample size.

Computation Time:

- The computational time is slightly higher for the Antithetic Variates method due to the additional calculations for paired paths. However, the variance reduction makes it more efficient overall.

2. Asian Call Option Pricing with Monte Carlo Simulation

2.1. Asian Option

An Asian option is a type of exotic derivative whose payoff depends on the average price of the underlying asset over a certain time interval, rather than its price at a single point in time. In this report, we focus on an Asian call option. For a call option, the payoff at maturity is given by the positive part of the difference between the underlying asset's terminal price and the strike price. In the case of an Asian call, the terminal price is replaced by the average of the underlying asset's price over the life of the option. More formally, the arithmetic average price at maturity T is:

$$\bar{S}_T = \frac{1}{m}\sum_{i=1}^{m} S_{t_i}$$

The payoff of an Asian call option with strike K is:

$$\text{Payoff} = \max(\bar{S}_T - K, 0)$$

This smoothing effect makes the option less sensitive to price spikes and can reduce the option's volatility compared to a standard European call.

2.2. Monte Carlo Simulation under the Black-Scholes-Merton Model

2.2.1. Pricing Method

To price an Asian call option under the Black-Scholes-Merton (BSM) framework, we assume the underlying asset price follows a geometric Brownian motion:

$$dS_t = S_t(r - q)dt + S_t \sigma dW_t$$

where r is the risk-free interest rate, q is the dividend yield, and σ is the volatility of the underlying asset. Under the risk-neutral measure, the discounted expected payoff of the option is:

$$\text{Option Price} = e^{-rT}\mathbb{E}[\max(\bar{S}_T - K, 0)]$$

The Monte Carlo approach involves simulating a large number N of possible price paths for the underlying asset. Each path is generated by discretizing the time interval [0, T] into m steps and using random draws from the normal distribution to increment the asset's log-price. For each simulated path, we compute the arithmetic average, the payoff, and then discount it back to present. Averaging these discounted payoffs over the N simulations provides an estimate of the option's fair value. We also compute the standard error and construct confidence intervals to quantify the statistical uncertainty of the estimates. As N increases, the law of large numbers and the central limit theorem ensure that the Monte Carlo estimate converges to the true option price with reduced variance.

2.2.2. Simulation Results

We implemented a Monte Carlo simulation for an Asian call option with the following parameters:

- Initial asset price: S0 =100
- Strike price: K = 100
- Time to maturity: T = 1 year
- Risk-free interest rate: r = 10%
- Dividend yield: q = 0
- Volatility: σ = 20%
- Number of time steps for averaging: m = 50

We ran simulations with different sample sizes N to investigate the convergence of our price estimate. The table below summarizes the results obtained for varying N:

Table4: simulation result

Sample size	Estimated Price	Standard Error	95% Confidence Interval	Computation Time (s)

10000	7.19116	0.0853402	[7.0239, 7.35843]	0.0686281
50000	7.15206	0.0386672	[7.07627, 7.22785]	0.337003
100000	7.14719	0.0273873	[7.09351, 7.20087]	0.680632
1000000	7.16189	0.00868503	[7.14487, 7.17891]	8.09343

Discussion:

- The estimated price stabilizes around a value about 7.16.
- The standard error decreases, indicating improved precision in our estimate.
- The 95% confidence interval narrows significantly as N grows.

This behavior is consistent with the theoretical properties of Monte Carlo simulation, where increasing the number of simulations reduces the sampling variance and leads to a more accurate and stable estimate of the true option price.

In this Monte Carlo simulation, our objective was to estimate the price of an Asian call option while ensuring that the relative error of the estimate remains within 1%. However, through an initial small-scale simulation of 10,000 iterations, we discovered that achieving this level of precision requires a significantly large number of simulations, approximately 68,765. In practical applications, such a high number of simulations can lead to substantial computational costs and increased processing time.

To address this challenge, we adopted an adaptive simulation approach aimed at reducing the required number of simulations without compromising the accuracy of the option price estimate. The process began with a preliminary pilot run involving a modest number of simulations (e.g., 10,000) to obtain an initial estimate of the option price and its standard error. This initial step allows for a rapid assessment of the price's volatility using limited computational resources.

2.2.3. Accuracy-speed balance

To calculate the price within 1% error. Initially, we increase the sample size N to a substantial number, such as 1000,000 iterations, allowed the estimated price to converge towards the theoretical value of approximately $7.16. However, this approach demanded significant computational resources and time. To address this challenge, we adopted an accuracy-speed balance strategy : aimed at reducing the prediction error to within 1% of the estimated price, thereby optimizing both computational speed and estimation accuracy.

The methodology involved a two-step approach, small-scale stimulation and final stimulation with sample size calculated.

We commenced with a preliminary simulation comprising 10,000 iterations. This initial run provided an estimate of the Asian call option price and its associated standard error (SE). The results from this pilot run were as follows:

Table 5: small-scale stimulation result

Sample Size (N)	10,000
Estimated Price	7.2034
Standard Error (SE)	0.0867

These preliminary estimates serve as the basis for calculating the required sample size to achieve the desired precision. The variability observed in the initial simulation informs the subsequent determination of N.

Utilizing the results from the small-scale simulation, we applied the following statistical formula to estimate the necessary sample size N to achieve a target relative error of 1% at a 95% confidence level:

$$N = \left(\frac{z * \ \sigma_{est}}{target \ \ error * initial \ estimate}\right)^2$$

Here, Z = 1.96 corresponds to the 95% confidenc level, = 0.0867 is te standard error from the initial run, and the target relative error is set at 1%. Substituting the values, we calculated that a sample size of approximately 68,765 iterations is necessary to achieve the desired precision. Armed with this estimated sample size, we conducted a final simulation with N = 68,765 iterations. The outcomes of this run were recorded as follows:

Table 6:final simulation result

Sample Size (N)	68,765	95% Confidence Interval	[7.1432, 7.2735]
Estimated Price	7.2083	Computation Time (s)	0.3410
Standard Error (SE)	0.0333	Relative Error	0.46%

This adaptive approach ensures that the simulation is both accurate and computationally efficient, avoiding the excessive computational burden that would result from arbitrarily increasing N without consideration of diminishing returns in precision.

2.2.4. Conclusion

The Monte Carlo results indicate that the Asian call option price under the given parameters converges to approximately 7.16 as N becomes large. The required sample size to approach less than 1% standard error is around 60,000, the confidence interval in this case is: [7.1432, 7.2735].

2.3. Control Variate Technique

2.3.1. Pricing Method

The geometric Asian call option's payoff:

$$Payoff_{\,geo} = e^{-rT} \cdot max\left(S_{geo} - K, 0\right)$$

Where:

$$S_{geo} = exp\left(\frac{1}{m}\sum_{\{j=1\}}^{\{m\}} lnS_j\right)$$

admits a closed-form solution. This exact geometric price can be used as a control variate due to its high correlation with the arithmetic average price.

If is the arithmetic Asian call payoff and is the geometric Asian call payoff with a known expectation, we define:

$$X^* = X_{arith} - \beta(Y_{geo} - \mathrm{E}\left(Y_{geo}\right))$$

With:

$$\beta = \frac{cov(X_{arith}, Y_{geo})}{var(Y_{geo})}$$

This adjustment reduces the variance of the estimator X^*, often dramatically, if X_{arith} and Y_{geo} are closely correlated.

2.3.2. Simulation Results

We generate N independent price paths for the underlying using the Euler discretization of the geometric Brownian motion:

$$S_{\{t+\Delta t\}} = S_t exp\left(\left(r - q - \frac{1}{2}\sigma^2\right)\Delta t + \sigma * \sqrt{\Delta t} Z\right)$$

where Z is a Gaussian Normal distribution, for each path, we compute both the arithmetic mean and the geometric mean, then calculate the discounted payoffs.

Table 7: Results without Control Variate

Arithmetic Asian Call Price	7.17343
Standard Error (SE)	0.00869251

Table 8: Result with Control Variate

Geometric Exact Price	6.89321
Geometric Mean	6.90164
beta	1.03689
Controlled Arithmetic Price	7.16469
Controlled SE	0.000256233
Variance Reduction (VRF)	1150.85
Computation Time(s)	5.08

The results demonstrate the profound impact of the control variate technique. The uncontrolled Monte Carlo simulation produces a reliable estimate but with a relatively high SE (~0.0087). After applying the control variate with the geometric Asian call option, the SE drops by a factor of more than 30, resulting in a standard error of roughly 0.000256.

This improvement in precision, quantified by a VRF of approximately 1150, confirms the strong correlation between the arithmetic and geometric averages. Such a high VRF indicates that, for the same computational effort, we achieve a drastically more accurate and stable price estimate, or equivalently, we could reduce the number of simulations significantly while maintaining the same level of precision.

The computation time remains efficient, just over 5 seconds for one million simulations, which is practical in a real-world setting. This underscores the scalability and effectiveness of the control variate method.

2.3.3. Conclusion

The experiment confirms that using the geometric Asian option as a control variate significantly enhances the efficiency of Monte Carlo simulation when pricing arithmetic Asian call options. The resulting dramatic reduction in variance and standard error enables traders and risk managers to obtain more reliable estimates at a fraction of the computational cost or to achieve extraordinarily high precision in a reasonable time frame.

This work reinforces the importance of variance reduction techniques in computational finance and exemplifies how theoretical insights (closed-form geometric pricing) can translate into practical numerical benefits.

2.4. Moment Matching Technique

2.4.1. Pricing Method

The moment matching (MM) technique aims to reduce variance in Monte Carlo simulations by adjusting the generated random variables so that their sample moments match the theoretical moments. By enforcing that each time step's set of generated standard normal draws has zero mean and unit variance, the simulation more closely aligns with the ideal Gaussian distribution, potentially improving the stability and convergence of the estimates.

This approach does not alter the underlying pricing model or the payoff calculation. Instead, it refines the sampling process. After generating all random normal variables for each simulation path, we adjust each time step column of samples by subtracting their sample mean and scaling by their sample standard deviation. This ensures that across all simulated paths, the random shocks for each time step collectively have the correct moments.

2.4.2. Simulation Results

The tables below compare the results of the Asian call option pricing simulation with and without moment matching. The simulation parameters remain unchanged from the previous experiment.

Table 9: Result with Moment Maching

Sample size	Estimated Price	Standard Error	95% Confidence Interval	Computation Time (s)
10000	7.1360	0.0850185	[6.96945, 7.30272]	0.0816463
100000	7.15741	0.0273966	[7.10371, 7.21111]	0.860963
1000000	7.16443	0.00868654	[7.1474, 7.18145]	9.1142

2.4.3. Conclusion

The results show that while the moment matching technique yields similar estimates for the option price, it does not provide a substantial reduction in variance compared to the standard simulation. In fact, the computational overhead of applying moment matching at each time step slightly increases the total computation time. Thus, under these particular settings, the additional complexity and computational cost of moment matching do not significantly improve the precision of the Monte Carlo estimate.

3. American Put Option Pricing with Monte Carlo Simulation

3.1. American Option

American option gives investors the right but not necessarily the obligation to buy or sell the underlying assets at any time before the expiration date. The payoff of an American option at time $t \in (0, T]$ is:

$$payoff = \begin{cases} max(S_t - K, 0), & call\ option \\ max(K - S_t, 0), & put\ option \end{cases}$$

where T is the maturity, K is the strike price, S_t is the price of underlying assets.

In real world, available early exercise time is continuous before the final expiration date T, but it is impractical if we try to simulate the price path. We assume that the option is exercisable M times before the maturity, and this discretization is termed a Bermuda exercise feature.

3.2. Pricing methods

3.2.1. Least Squares Monte Carlo

Longstaff and Schwartz (2001) proposed a method which combines simulating multiple possible price paths of underlying assets using Monte Carlo and estimating the continuation prices for the paths where the option is in the money at time t_{k-1} by regressing the discounted option values at time t_k, that is why it is called Least Squares Monte Carlo. At each time step, we compare the continuation value with the immediate exercise value or so-called intrinsic value to determine whether we want to exercise this option early or keep holding it. Finally, the option's expected value is obtained at time 0 after we work above process backward recursively.

Choices in the regression techniques are worth discussing since the accuracy of continuation value is relevant to these:

(1) Basis functions and their overall power. Longstaff and Schwartz (2001) calculate continuation value based on the first three weighted Laguerre polynomials as basis functions. They argue that numerical results are robust to different basis functions, no matter they are orthogonal polynomials including the Hermite, Legendre, Chebyshev, Gegenbauer and Jacobi polynomials, Fourier series, trigonometric series or just simple powers. And when it comes to the power, they argue that few basis functions are needed to approximate the continuation value closely. Rodrigues and Armada (2006) suggest that weighted Laguerre polynomials produce more accurate estimates.

(2) The choice of state variables. Longstaff and Schwartz (2001) use underlying asset price as the regressor and rescale it by dividing to strike price. Renormalization avoids computational underflows caused by directly applying the weighted Laguerre polynomials. So they recommend normalizing appropriately to avoid numerical errors resulting from scaling problems.

(3) Regression methods. A near singular cross-moment matrix can raise numerical inaccuracy in some least squares algorithms. Longstaff and Schwartz (2001) suggest use alternative algorithms like Cholesky or QR decomposition. Additionally, they mention that methods such as weighted least squares, generalized least squares or even GMM may be more efficient than ordinary least squares.

On the other hand, Rodrigues and Armada (2006) point out the quality of the simulation can be improved by using variance reduction techniques such as antithetic variates and moment matching techniques, or low-discrepancy sequences.

To sum up, LSMC provides a simple and elegant way to simulate early exercise strategy and approximate value of American-style option.

3.2.2. Binomial Black and Scholes method with Richardson extrapolation

Binomial tree is also a useful method for American option pricing. At each time step, we assume the price of underlying asset moves up or down in equal proportion, if we determine the price change factor and corresponding risk neutral probability, we can derive option price by discounting the expected value of every step through backward induction.

Here we introduce some different forms of binomial tree:

(1) CRR tree proposed by Cox et al. (1979):

$$\mathrm{u} = \mathrm{e}^{\sigma\sqrt{\Delta t}}, d = \mathrm{e}^{-\sigma\sqrt{\Delta t}}, p = \frac{e^{r\Delta t} - d}{u - d}$$

Where σ is the volatility of underlying asset, r is the risk free rate, Δt is the length of each time step.

(2) JR tree proposed by Jarrow and Rudd (1983):

$$u = e^{\left(r-\frac{1}{2}\sigma^2\right)\Delta t+\sigma\sqrt{\Delta t}}, d = e^{\left(r-\frac{1}{2}\sigma^2\right)\Delta t-\sigma\sqrt{\Delta t}}, p = \frac{1}{2}$$

which is not a risk-neutral tree, but can be modified to a Jarrow-Rudd risk-neutral tree (JRRN) using $p = \frac{e^{r\Delta t}-d}{u-d}$.

(3) Tian tree proposed by Tian (1993):

$$\hat{r} = e^{r\Delta t}, \hat{\sigma} = e^{\sigma^2 \Delta t}, p = \frac{\hat{r} - d}{u - d}$$

$$u = \frac{\hat{r}\hat{\sigma}}{2}\left(\hat{\sigma} + 1 + \sqrt{\hat{\sigma}^2 + \hat{\sigma} - 3}\right), d = \frac{\hat{r}\hat{\sigma}}{2}\left(\hat{\sigma} + 1 - \sqrt{\hat{\sigma}^2 + \hat{\sigma} - 3}\right), p = \frac{\hat{r} - d}{u - d}$$

which considers the exact solution to the moment matching conditions.

Although CRR, JR and Tian models adapt the different probability measure, all of them lead to the same result as time steps approaches the limit of infinity.

Broadie and Detemple (1996) propose a binomial method with two modifications. First improvement is called Binomial Black and Scholes (BBS) method, which calculates continuation value using Black-Scholes formula instead of discounted expected value at the time step just before maturity. It is an effective way to reduce price oscillation and smooth convergence. Since no early exercise opportunity exists within the final step, the option is effectively European-style, then a more accurate price can be obtained through BSM equation for this step. Second improvement is Richardson extrapolation, which improves accuracy of numerical method. Consider an option price P(M) derived from a M steps binomial tree, we can expand the equation with a constant F:

$$P(M) = Exact\ price + \frac{F}{M} + o\left(\frac{1}{N}\right)$$

then,

$$2P(M) - P\left(\frac{M}{2}\right) = Exact\ price + o\left(\frac{1}{N}\right)$$

In this way we cancel out the leading error term.

3.3. Numerical results

We try to price an American put option with $S0 = K = 100$, $T = 1/12$, $r = 4\%$, $q = 2\%$, $\sigma = 20\%$. Random seed is 42 and generated by "mt19937_64" algorithm. All calculation is based on Eigen.

3.3.1. BBSR

Set time steps $M = 50$, we get option price is 2.22673 and simulation time is 0.0000948s under CRR tree.

3.3.2. LSMC

Following Longstaff and Schwartz (2001), we generate stock price paths using antithetic approach, choose weighted Laguerre polynomials as basis functions, renormalize stock price dividing to strike price and adopt QR decomposition to calculate OLS coefficients.

Set sample size $N = 10000$, time steps $M = 50$, number of regressors $k = 3$, we get estimated price is 2.24528, standard error is 0.00732206, 95% confidence interval is [2.23093, 2.25964] and computation time is 0.0967992s.

Now change sample size, time steps and number of regressors, we analyze how these parameters affect the approximate option price.

(1) Fix $M = 50$, $k = 3$, set N from 100 to 10000 with a step size of 100. As N increases, option prices drop quickly and when N is larger than 500, estimated prices begin to oscillate around the real price.

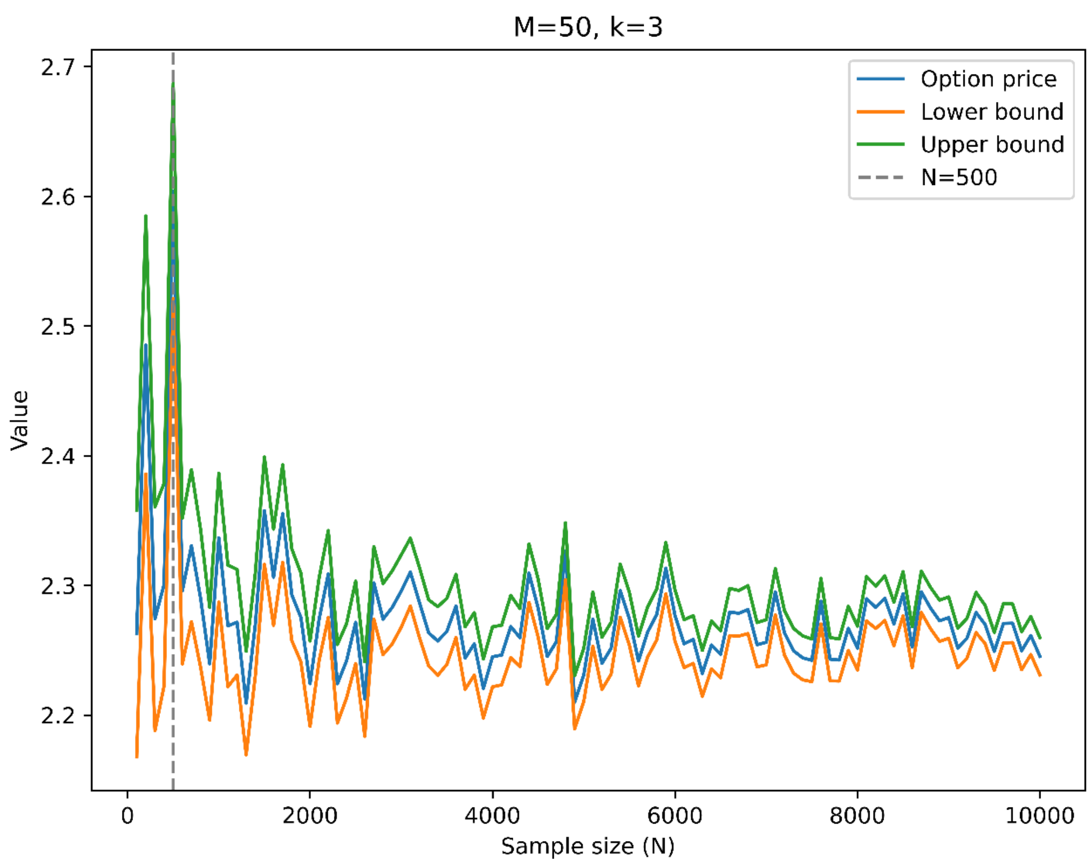

Figure 4: Option price against Sample size (N)

(2) Fix $N = 10000$, $k = 3$, set M from 5 to 500 with a step size of 1. It is weird when M is less than 200 and larger than 200, estimated prices seem to oscillate around different

“real prices”. It is a pity that we have not found any literature explaining this phenomenon, but in practice (for example, Longstaff and Schwartz (2001) set $M = 50$ for $T = 1$ or 2 and $\sigma = 0.2$ or 0.4), it is appropriate to choose time steps around 50 for an option with short maturity or low volatility. And when compared with BBSR, we advise against choosing too large time steps. Here we have a conjecture, only stock price as state variable is insufficient for the regression, then accumulated inaccuracy result in this strange performance as time steps increase.

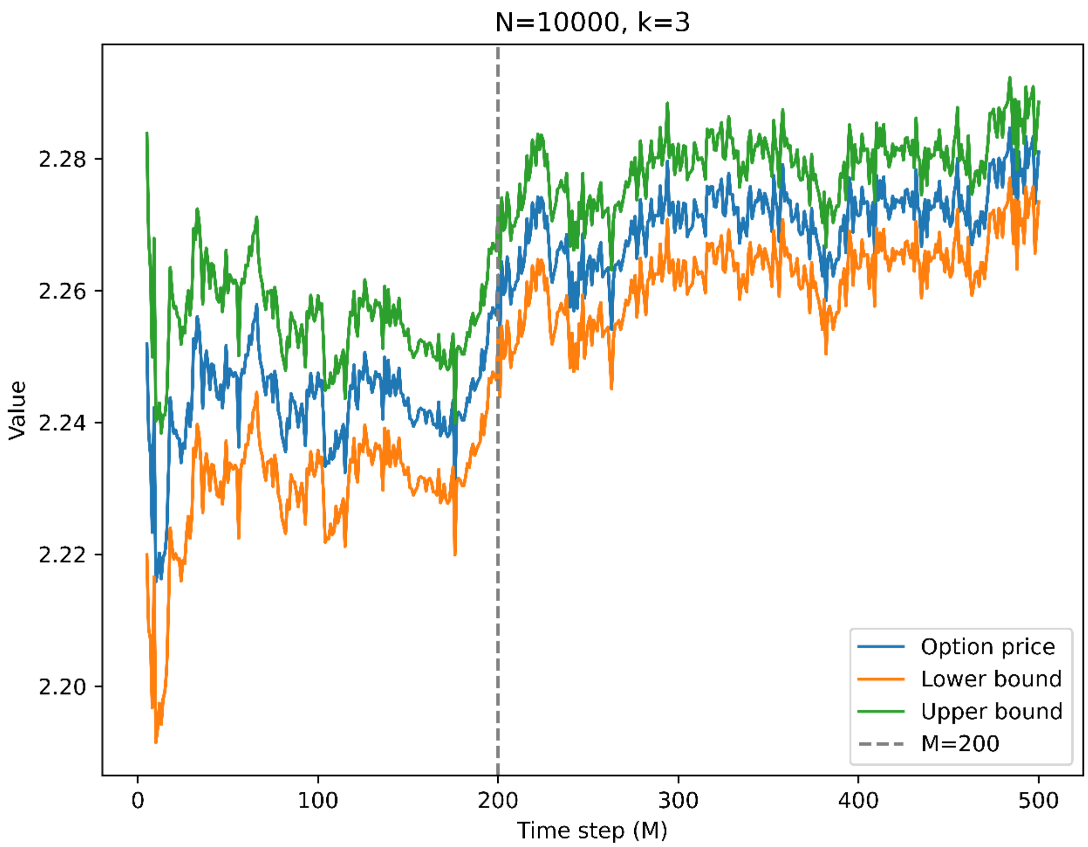

Figure 5: Option price against Time step (M)

(3) Fix $N = 10000$, $M = 50$, set k from 2 to 20 with a step size of 1. As k increases, estimated prices oscillate around the real price. When k is larger than 12, the results stay the same, suggesting that under the restriction of sample size and time steps, the effect of increasing the number of regressors is also limited.

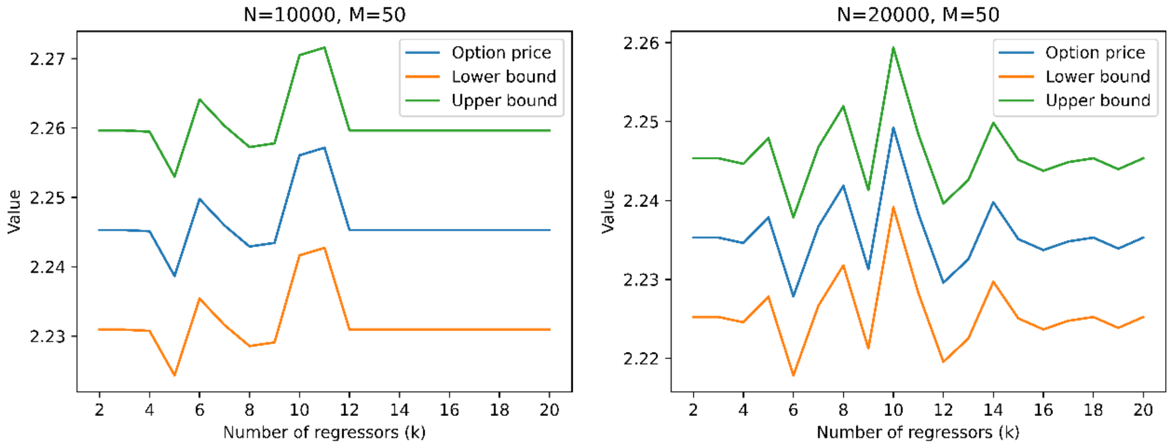

Figure 6: Option price against Number of regressors (k)

If we set $N = 20000$, estimated prices still oscillate around the real price, more regressors do not bring more power to explain. It is consistent with Longstaff and Schwartz (2001) that few basis functions are needed to approximate the continuation value closely.

Overall, LSMC is a robust method to price American option. There is a speed-accuracy trade-off among parameter setting, but appropriate size is beneficial to quick convergence of estimates without loss of accuracy.

References

Broadie, Mark, and Jerome Detemple. "American option valuation: new bounds, approximations, and a comparison of existing methods." *The Review of Financial Studies* 9.4 (1996): 1211-1250.

Cox, John C., Stephen A. Ross, and Mark Rubinstein. "Option pricing: A simplified approach." *Journal of financial Economics* 7.3 (1979): 229-263.

Longstaff, Francis A., and Eduardo S. Schwartz. "Valuing American options by simulation: a simple least-squares approach." *The review of financial studies* 14.1 (2001): 113-147.

Robert A. Jarrow and Andrew Rudd. Option Pricing. Irwin Professional, 1983.

Rodrigues, Artur, and Manuel J. Rocha Armada. "The valuation of real options with the least squares monte carlo simulation method." *Available at SSRN 887953* (2006).

Tian, Yisong. "A modified lattice approach to option pricing." *The Journal of Futures Markets (1986-1998)* 13.5 (1993): 563.