Monte Carlo Techniques for Options Pricing

Kamin Atsavasirilert

2023.12.12

Introduction

This study investigates various Monte-Carlo-based and variance reduction techniques to value European, American-style and Asian options. The research begins by applying Monte-Carlo simulation in pricing European vanilla put options and examining the evolution of the absolute pricing error and computing time as the number of samples in the simulation increases when using and not using an antithetic approach. Subsequently, we study the convergence of Asian option prices using Monte Carlo simulation with and without control variate approach when the sample size increases. Lastly, we implement Least-Squares Monte Carlo and study how the main parameters, sample sizes, number of timesteps, number of regressors, affect approximated American-style put option prices while using the prices obtained from Binomial Black-Scholes with Richardson Extrapolation (BBSR) as a benchmark.

Theory

Geometric Brownian Motion of Asset Prices

A geometric Brownian motion of asset prices is a stochastic process for asset price movement modeling, which has the analytical solution shown below under Black-Scholes-Merton model assumptions.

$$S_t = S_0 exp((r - q - 0.5\sigma^2)t + \sigma B_t)$$

Where S_t is the price of the asset at timestep t.

r is the risk-free rate

σ is the yearly volatility of the underlying asset

q is the continuous dividend yield of the underlying asset

B_t is the standard Brownian motion

Black-Scholes-Merton formula for European option pricing

Black-Scholes-Merton formula is a very well known close-formed solution for European option pricing. The formula is often used as a benchmark to validate other pricing algorithms. The formula for a European put option pricing is:

$$P = exp(-rt)kN(-d2) - S_0 exp(-qt)N(-d1)$$

$$d1 = \frac{ln(\frac{S_0}{k}) + (r - q - 0.5\sigma^2)T}{\sigma\sqrt{T}}$$

$$d2 = d1 - \sigma\sqrt{T}$$

Where T is the remaining time to maturity date in years.

N is the standard normal cumulative distribution function.

k is the strike price of the put option

Monte Carlo Simulation for European put options pricing

The price of a put European option can be obtained by solving the equation below.

$$p = exp(-rt)E[max(0, k - S_t)]$$

Where p is the price of the put option

As the future underlying asset price S_t is stochastic, Monte Carlo simulation is a powerful and widely-used approach to estimate the expectation term $E[max(0,\ k\ -\ S_t)]$ because of its simplicity and flexibility. However, to obtain a precise estimation of the expectation, Monte Carlo simulation requires a large number of samples, which increases the computing time significantly. Several techniques, such as antithetic and control variate, attempt to address the issue by using statistical properties to reduce the variance of the estimated value.

Antithetic technique

Antithetic technique is a well-known technique to reduce the variance of an estimator obtained by Monte Carlo simulation. An example of estimating the expectation of a random variable Y with an estimator Θ is shown below.

$$\Theta\ =\ E[Y]$$

Considering a case of generated 2 samples Y_1 and Y_2, we therefore have:

$$\widehat{\theta} = \frac{Y_1+Y_2}{2}$$

and

$$Var(\widehat{\theta})\ = \frac{Var(Y_1)+Var(Y_2)+2Cov(Y_1,Y_2)}{4}$$

The variance $Var(\widehat{\theta})$ can be reduced when the $Cov(Y_1,\ Y_2)$ is negative, which can be done by utilizing the negative part of B_t to generate another price path during simulation. With antithetic technique, we can efficiently reduce the variance of the estimator while the mean remains.

Control variate technique

Control variate technique is similar to antithetic technique. The main goal is to estimate $\Theta\ =\ E[Y]$ and reduce the variance of the estimator by utilizing the statistical properties of the estimator. A new random variable X is created using the following equation.

$$X\ =\ Y\ -\ b(C\ -\ E[C])$$

Where b is a constant and C is another random variable. The idea is, instead of finding $\Theta\ =\ E[Y]$, to find $E[X]$, which share the same mean with $E[Y]$ but the variance becomes:

$$Var(X)\ =\ Var(Y)\ +\ b^2Var(C)\ -\ 2bCov(Y,C)\ \text{—}\ (1)$$

To minimize $Var(X)$, we differentiate the above equation with respect to b and set it to 0. So, we obtain:

$$b^*\ = \frac{Cov(Y,C)}{Var(C)}$$

We then substitute b^* in (1). So, we have:

$$Var(X)\ =\ Var(Y)(1\ -\ \rho(Y,C)^2)$$

The result shows the opportunity to reduce $Var(X)$ by selecting appropriate random variable C that is highly correlated with Y.

Least Squares Monte Carlo (LSMC) for American-style put options pricing

Least Squares Monte Carlo is a simple and elegant approach to value American-style options. The idea is to make a decision to or not to early exercise the option based on the prediction of Ordinary Least Squares regression (OLS). Put option prices obtained from the technique are proved to be as shown below:

$$P(X) \geq \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} LSMC(\Theta_i; M, K)$$

Where $P(X)$ is the actual put price, N is the number of samples, M is the order of the basis function to estimate future cash flow, K is the number of timesteps. The equation above shows that the put prices obtained from LSMC will alway be less than or equal to their actual prices. The difference however can be reduced as N, M and K increase.

Binomial Black and Scholes method with Richardson extrapolation (BBSR) for American-style put options pricing

BBSR is an effective way to approximate the value of an American-style put option as it requires a few modifications on top of the Binomial Black and Scholes method (BBS), which is a modified binomial tree algorithm that replaces the continuation value with the Black-Scholes-Merton option pricing formula at one step before maturity. The technique yields much more stable output than the traditional binomial tree algorithm, which uses the continuation value at one step before maturity as an option payoff function. As the output prices are stabilized, Richardson extrapolation is applied to obtain more accurate prices by the following equation.

$$P = 2P_{n1} - P_{n2}$$

Where P is the BBSR approximated option price, P_n is the BBS approximated option price with n samples, and $n1 > n2$.

Results

Our study can be divided into 3 main experiments. The first experiment is to apply Monte-Carlo simulation in pricing European vanilla put options and examining the evolution of the absolute pricing error and computing time as the number of samples in the simulation increases when using and not using an antithetic approach. Secondly, we study the convergence of Asian option prices using Monte Carlo simulation with and without control variate approach when the sample size increases. Lastly, we implement Least-Squares Monte Carlo and study how the main parameters, sample sizes, number of timesteps, number of regressors, affect approximated American-style put option prices while using the prices obtained from Binomial Black-Scholes with Richardson Extrapolation (BBSR) as a benchmark.

Experiment 1: Antithetic technique for European put option pricing

The code is written is C++ and the corresponding code file is European_Option_Pricing_Monte_Carlo.cpp, which does not require input as all the parameters are already hard coded. The outputs of the code are shown in the figure below.

During price path generation, we use not only Z_i which is a standard random normal number but also use $-Z_i$ to generate 2 paths, which are highly negatively correlated to each other. By doing so, we are generating highly negative $Cov(Y_1, Y_2)$ when Y_1 and Y_2 are different price paths, and expect the standard error to decrease from using a plain Monte Carlo simulation. This method however is more time consuming as more computation is required as shown in Figure b.

```
C:\Users\kamin\source\repos\European_Option_Pricing_Monte_Carlo\x64\Debug>European_Option_Pricing_Monte_Carlo.exe
n: 50000 with variance reduction technique, computing time: 0.004 BSM_put_value: 5.07463
option_value: 5.06088 standard_error: 0.0160353
 95% confidence interval: [5.02945, 5.09231] with Absolute Error:0.0137563
----------------
n: 50000 without variance reduction technique, computing time: 0.004 BSM_put_value: 5.07463
option_value: 5.08068 standard_error: 0.0320524
 95% confidence interval: [5.01786, 5.1435] with Absolute Error:-0.00604718
---------------------------------------------------------
n: 100000 with variance reduction technique, computing time: 0.009 BSM_put_value: 5.07463
option_value: 5.05576 standard_error: 0.01134
 95% confidence interval: [5.03354, 5.07799] with Absolute Error:0.0188725
----------------
n: 100000 without variance reduction technique, computing time: 0.007 BSM_put_value: 5.07463
option_value: 5.06122 standard_error: 0.0226519
 95% confidence interval: [5.01683, 5.10562] with Absolute Error:0.0134104
---------------------------------------------------------
n: 500000 with variance reduction technique, computing time: 0.046 BSM_put_value: 5.07463
option_value: 5.07419 standard_error: 0.00509041
 95% confidence interval: [5.06421, 5.08416] with Absolute Error:0.000448741
----------------
n: 500000 without variance reduction technique, computing time: 0.037 BSM_put_value: 5.07463
option_value: 5.07862 standard_error: 0.010167
 95% confidence interval: [5.05869, 5.09854] with Absolute Error:-0.00398268
---------------------------------------------------------
n: 1000000 with variance reduction technique, computing time: 0.092 BSM_put_value: 5.07463
option_value: 5.07193 standard_error: 0.00359492
 95% confidence interval: [5.06489, 5.07898] with Absolute Error:0.00270061
----------------
n: 1000000 without variance reduction technique, computing time: 0.074 BSM_put_value: 5.07463
option_value: 5.06802 standard_error: 0.00717769
 95% confidence interval: [5.05395, 5.08209] with Absolute Error:0.00661503
---------------------------------------------------------
n: 5000000 with variance reduction technique, computing time: 0.454 BSM_put_value: 5.07463
option_value: 5.07366 standard_error: 0.00160744
 95% confidence interval: [5.07051, 5.07681] with Absolute Error:0.000977257
----------------
n: 5000000 without variance reduction technique, computing time: 0.367 BSM_put_value: 5.07463
option_value: 5.07285 standard_error: 0.00321056
 95% confidence interval: [5.06656, 5.07914] with Absolute Error:0.00178592
---------------------------------------------------------
n: 10000000 with variance reduction technique, computing time: 0.909 BSM_put_value: 5.07463
option_value: 5.07509 standard_error: 0.00113683
 95% confidence interval: [5.07286, 5.07732] with Absolute Error:-0.00045212
----------------
n: 10000000 without variance reduction technique, computing time: 0.731 BSM_put_value: 5.07463
option_value: 5.0764 standard_error: 0.00227153
 95% confidence interval: [5.07194, 5.08085] with Absolute Error:-0.00176092
---------------------------------------------------------
n: 50000000 with variance reduction technique, computing time: 4.514 BSM_put_value: 5.07463
option_value: 5.07512 standard_error: 0.000508557
 95% confidence interval: [5.07412, 5.07612] with Absolute Error:-0.00048365
----------------
n: 50000000 without variance reduction technique, computing time: 3.641 BSM_put_value: 5.07463
option_value: 5.07541 standard_error: 0.00101606
 95% confidence interval: [5.07342, 5.0774] with Absolute Error:-0.000776621
---------------------------------------------------------
n: 100000000 with variance reduction technique, computing time: 9.029 BSM_put_value: 5.07463
option_value: 5.07466 standard_error: 0.000359631
 95% confidence interval: [5.07395, 5.07536] with Absolute Error:-2.45135e-05
----------------
n: 100000000 without variance reduction technique, computing time: 7.273 BSM_put_value: 5.07463
option_value: 5.07495 standard_error: 0.000718462
 95% confidence interval: [5.07355, 5.07636] with Absolute Error:-0.000319268
---------------------------------------------------------
n: 500000000 with variance reduction technique, computing time: 49.809 BSM_put_value: 5.07463
option_value: 5.07485 standard_error: 0.000160834
 95% confidence interval: [5.07454, 5.07517] with Absolute Error:-0.000218285
----------------
n: 500000000 without variance reduction technique, computing time: 36.451 BSM_put_value: 5.07463
option_value: 5.07473 standard_error: 0.000321305
 95% confidence interval: [5.0741, 5.07536] with Absolute Error:-9.26603e-05
---------------------------------------------------------
```

Figure a: The results of our code implementation (experiment 1)

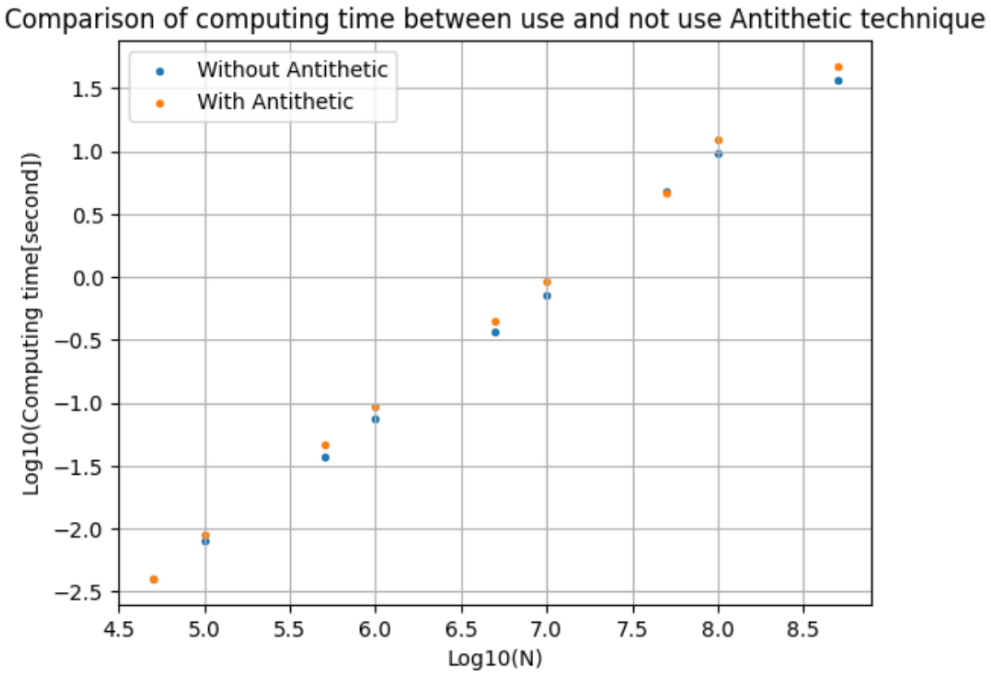

Figure b: The computing time when varying N

As shown in Figure a, our program outputs sample size, calculated put option value, estimated standard errors, 95% confidence interval, and the absolute pricing error. We then compare the prices obtained by using and not using the antithetic technique with the price obtained by the Black-Schole-Merton formula in Figure c. Moreover, as we expect lower standard error when using antithetic technique, Figure d shows that our expectation is in line with the experimental results.

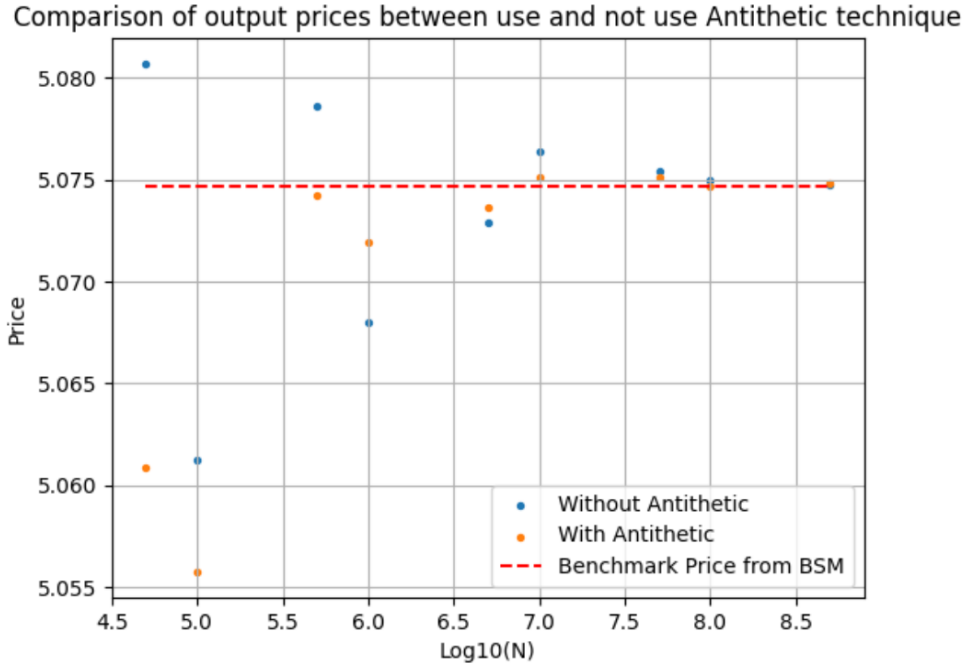

Figure c: The convergence of the output prices

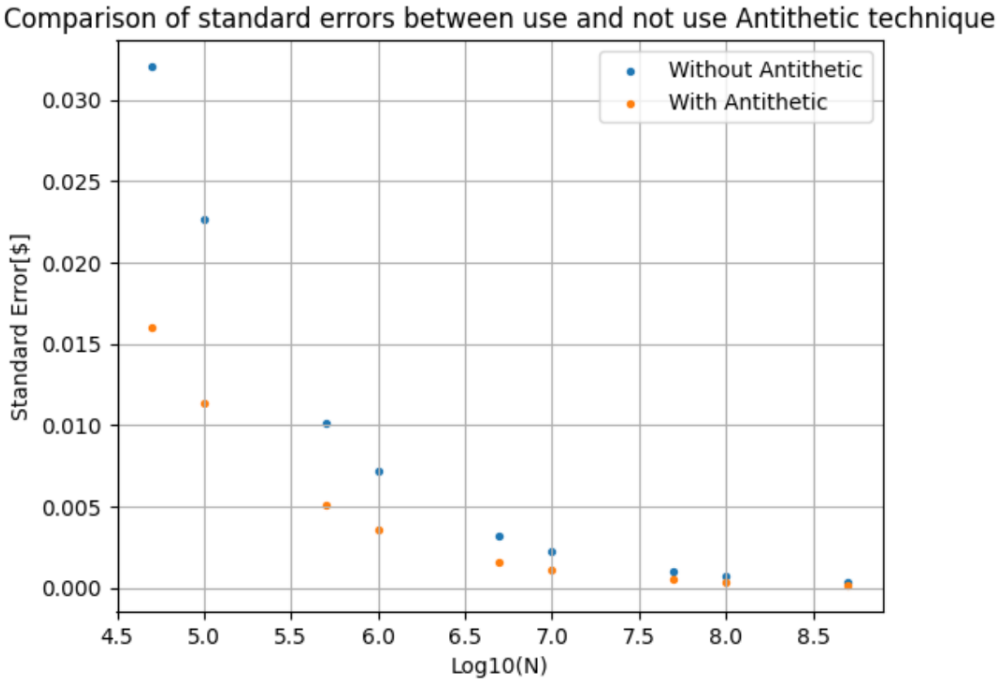

Figure d: The standard error when varying N

Experiment 2: Control variate approach in pricing Asian put option pricing

The code is written is C++, and the corresponding code file is Asian_option_with_variance_reduction.cpp, which does not require input as all the parameters are already hard coded. The outputs of the code are shown in the figure below. With control variate approach, we use a geometric asian option as out control variable C as its payoffs should be highly correlated to that of an arithmetic asian option with the same parameters. Therefore, we should expect small standard errors when the technique is applied. One main drawback of the control variate approach is that it requires b^* and $E[C]$, and they need to be computed before the simulation. So, we run a simulation with a small number of samples, in our case n=1,000,000, to obtain those parameters beforehands. Otherwise, it may hurt computing time. Another major drawback of the approach is because we substitute $E[Y]$ by $E[X]$ where $X = Y - b(C - E[C])$. In theory, $E[Y]$ and $E[X]$ should be the same, but it is not true in practice as we need to estimate b^* and $E[C]$ using a limited number of samples. With the limitation, estimating $E[Y]$ by using $E[X]$ can be biased. Therefore, the first simulation to obtain b^* and $E[C]$ is critical. In contrast, this drawback does not occur in both plain Monte Carlo simulation with and without the antithetic technique. The standard errors when using control variate technique are indeed smaller than that of the plain Monte Carlo method, but the confidence intervals may not contain the true value although the confidence level is set as high as possible. This is because of the biasedness of the estimator as mentioned before.

```
C:\Users\kamin\source\repos\Asian_option_with_variance_reduction\x64\Debug>Asian_option_with_variance_reduction.exe
b_star: 1.03692 avg_c: 7.62584 rho_xc: 0.999566n: 50000
without control variable
option_value: 7.10274 standard_error: 0.0384534 95% confidence interval: [7.02738, 7.17811] execution time: 0.139
----------------------
with control variable
option_value: 7.1718 standard_error: 0.00115654 95% confidence interval: [7.16953, 7.17406] execution time: 3.334
-----------------------------------------------------------------
b_star: 1.03684 avg_c: 7.61124 rho_xc: 0.999566n: 100000
without control variable
option_value: 7.18544 standard_error: 0.0275657 95% confidence interval: [7.13141, 7.23947] execution time: 0.27
----------------------
with control variable
option_value: 7.1586 standard_error: 0.000811317 95% confidence interval: [7.15701, 7.16019] execution time: 3.455
-----------------------------------------------------------------
b_star: 1.03697 avg_c: 7.60557 rho_xc: 0.999563n: 500000
without control variable
option_value: 7.16679 standard_error: 0.0123039 95% confidence interval: [7.14267, 7.19091] execution time: 1.351
----------------------
with control variable
option_value: 7.15322 standard_error: 0.000362339 95% confidence interval: [7.15251, 7.15393] execution time: 4.815
-----------------------------------------------------------------
b_star: 1.03682 avg_c: 7.61467 rho_xc: 0.999566n: 1000000
without control variable
option_value: 7.17397 standard_error: 0.00868939 95% confidence interval: [7.15694, 7.191] execution time: 2.736
----------------------
with control variable
option_value: 7.16141 standard_error: 0.000256712 95% confidence interval: [7.16091, 7.16191] execution time: 6.423
-----------------------------------------------------------------
b_star: 1.03696 avg_c: 7.62513 rho_xc: 0.999564n: 5000000
without control variable
option_value: 7.16676 standard_error: 0.00388291 95% confidence interval: [7.15915, 7.17437] execution time: 14.345
----------------------
with control variable
option_value: 7.17112 standard_error: 0.000114637 95% confidence interval: [7.17089, 7.17134] execution time: 20.14
-----------------------------------------------------------------
b_star: 1.03699 avg_c: 7.62407 rho_xc: 0.999564n: 10000000
without control variable
option_value: 7.16464 standard_error: 0.00274709 95% confidence interval: [7.15926, 7.17003] execution time: 27.399
----------------------
with control variable
option_value: 7.1702 standard_error: 8.11819e-05 95% confidence interval: [7.17004, 7.17036] execution time: 35.368
-----------------------------------------------------------------
```

Figure e: The results of our code implementation (experiment 2)

For the convergence of the method, we try using 2 the number of samples, N_b = 10,000 and 1,000,000, for the simulation to obtain b^* and $E[C]$. From Figure f, as expected when N_b is too small, in this case N_b = 10,000, the output can be extreme because of the biasedness even though the standard errors (see Figure g) are close to the case of N_b = 1,000,000.

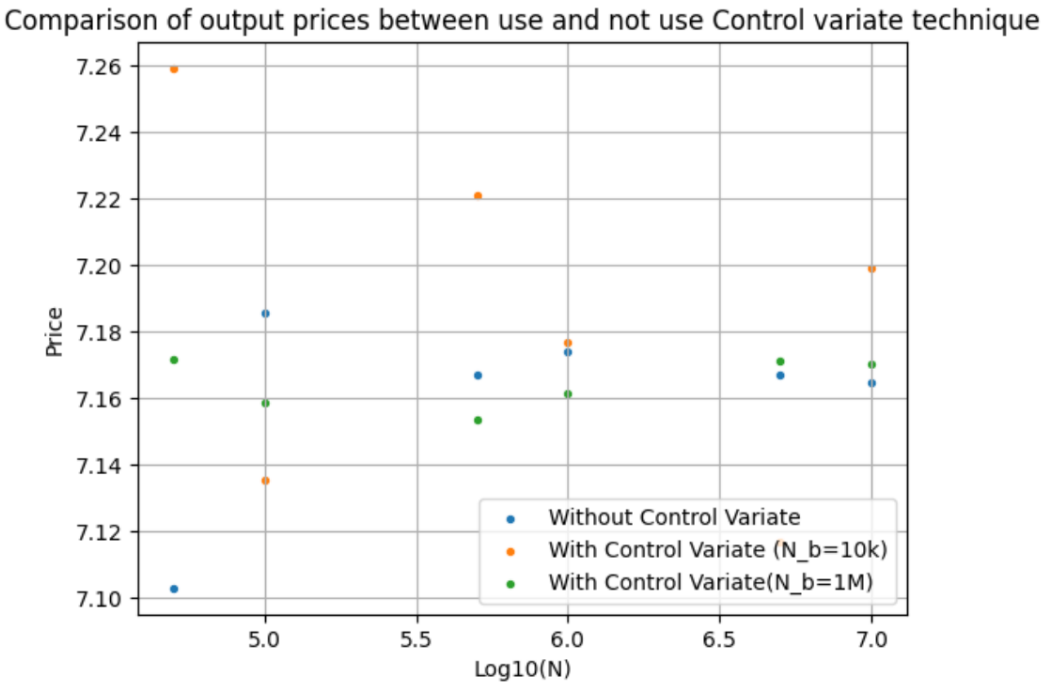

Figure f: The convergence of the output prices

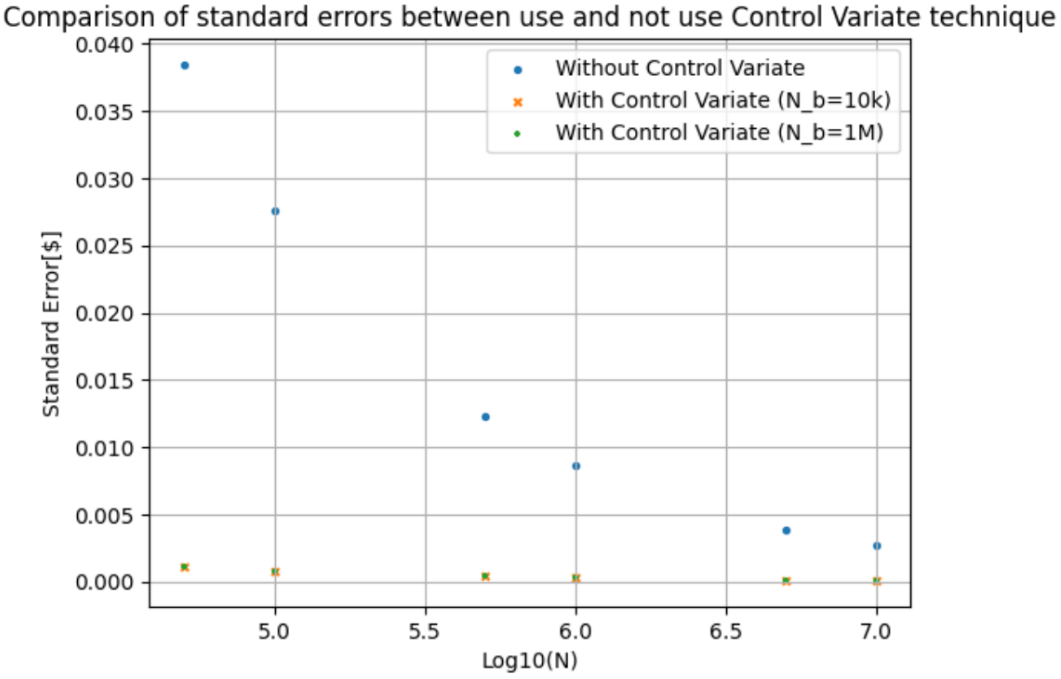

Figure g: The standard error when varying N and N_b

Experiment 3: Least Squares Monte Carlo (LSMC) for American-style put option pricing

In this experiment, we implement the LSMC to price an American-style put option with $S_0 = k = 100, r = 0.04, q = 0.02$ and $\sigma = 0.2$. The implementation of LSMC is done using R , and the corresponding code file is Least_squares_monte_carlo.html (R markdown). The put option prices obtained from LSMC are then compared with a benchmark which are the results obtained from the BBSR approach. The implementation of the BBSR approach is done using C++, and the corresponding code file is Binomial_Black_and_Scholes_method.cpp.

Because the BBSR approach is used as the benchmark in this experiment, we firstly introduce our implementation and its results. The code takes 1 input which is the number of states of the binomial tree used in estimating the option price. The code is implemented using recursion and memoization techniques to speed up the runtime. If the number of states were to set > 2,500, we recommend increasing the stack size of the program to avoid stack overflow. Figure h shows the results of the code, which yield the benchmark option price of 2.22591.

```
C:\Users\kamin\source\repos\Binomial_Black_and_Scholes_method\x64\Debug>Binomial_Black_and_Scholes_method.exe 2000
n_state: 2000
dt= 4.16667e-05 u= 1.00129 pu= 0.5 discount_factor= 0.999998
BBS put_price (n=2000): 2.22603 time used: 0.069
n_state: 1000
dt= 8.33333e-05 u= 1.00183 pu= 0.5 discount_factor= 0.999997
BBS put_price (n=1000): 2.22615 time used: 0.018
BBSR put_price =  2.22591 total time used: 0.105
```

Figure h: The results of BBSR in pricing the given put option

For the LSMC method, we investigate how the number of samples (N), number of timesteps (m) and highest polynomial order when fitting OLS (K) affect the option prices. We firstly vary N = [1,000, 2,000, 4,000, 8,000, 16,000, 36,000, 64,000, 128,000] while m = 5 and K = 2 are fixed. According to Figure i and Table a, the estimated prices approach the benchmark price as N increases. The standard errors also reduce by nearly $1/\sqrt{2}$ times when N increases by 2 times.

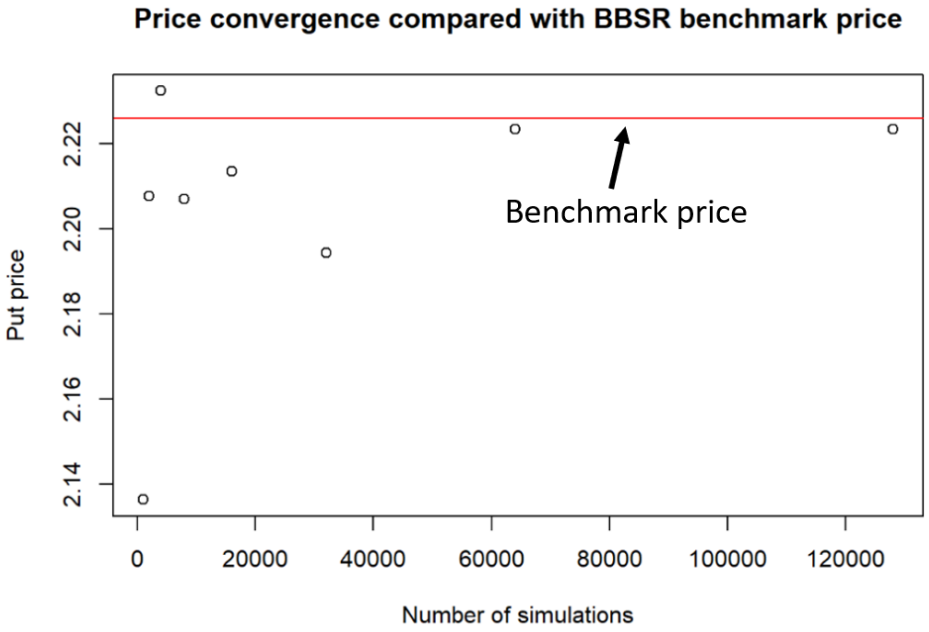

Figure i: The convergence of the estimated option price obtained from LSMC when varying the number of samples (N)

Table a: The standard errors of the estimated option prices obtained from LSMC when varying the number of samples (N)

Number of samples (N)	Standard Error
1,000	0.08264
2,000	0.06539
4,000	0.04854
8,000	0.03483
16,000	0.02331
32,000	0.01613
64,000	0.01149
128,000	0.00821

To investigate the effect of the number of timesteps (m), We then vary m = [5, 10, 20, 40] while N = 20,000 and K = 2 are fixed. According to Figure j and Table b, we do not observe a trend when varying m.

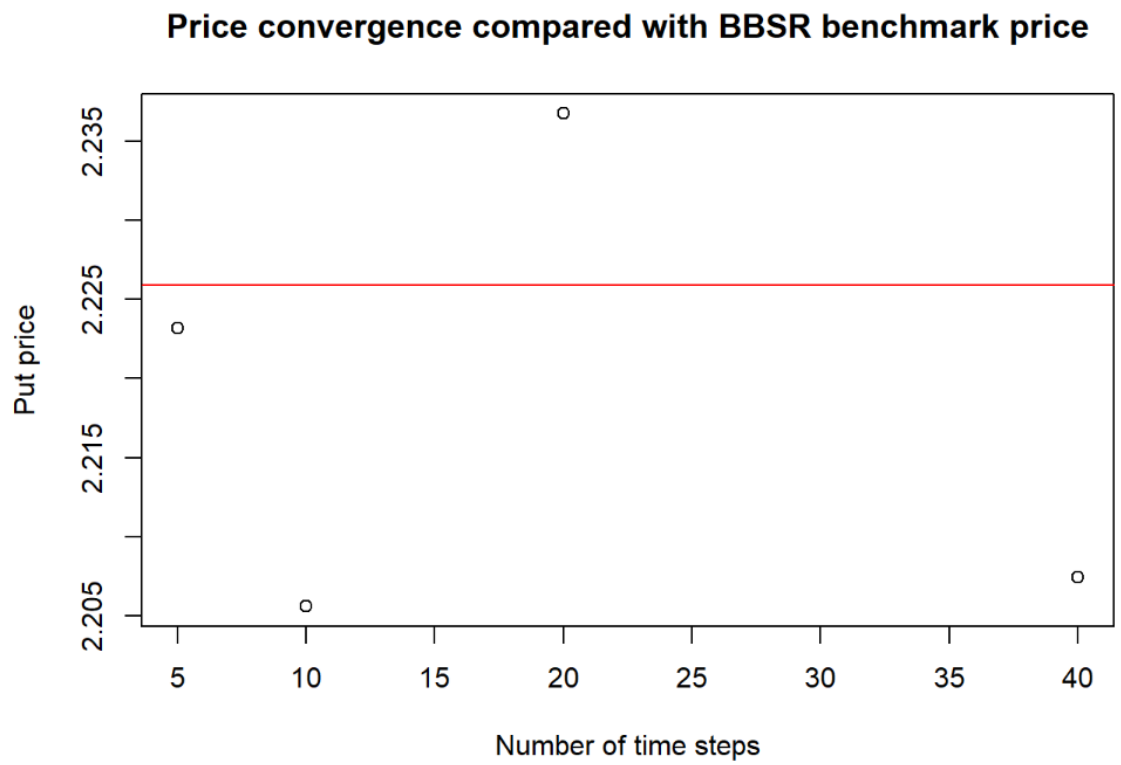

Figure j: The convergence of the estimated option price obtained from LSMC when varying the number of timesteps (m)

Table b: The standard errors of the estimated option prices obtained from LSMC when varying the number of timesteps (m)

Number of timesteps (m)	Standard Error
5	0.02082797
10	0.01995568
20	0.01988777
40	0.01907113

To investigate the effect of the highest polynomial order when fitting OLS (K), We then vary K = [1, 2, 3, 4, 5, 6] while N = 20,000 and m = 5 are fixed. According to Figure h and Table c, we do not observe a trend when varying K.

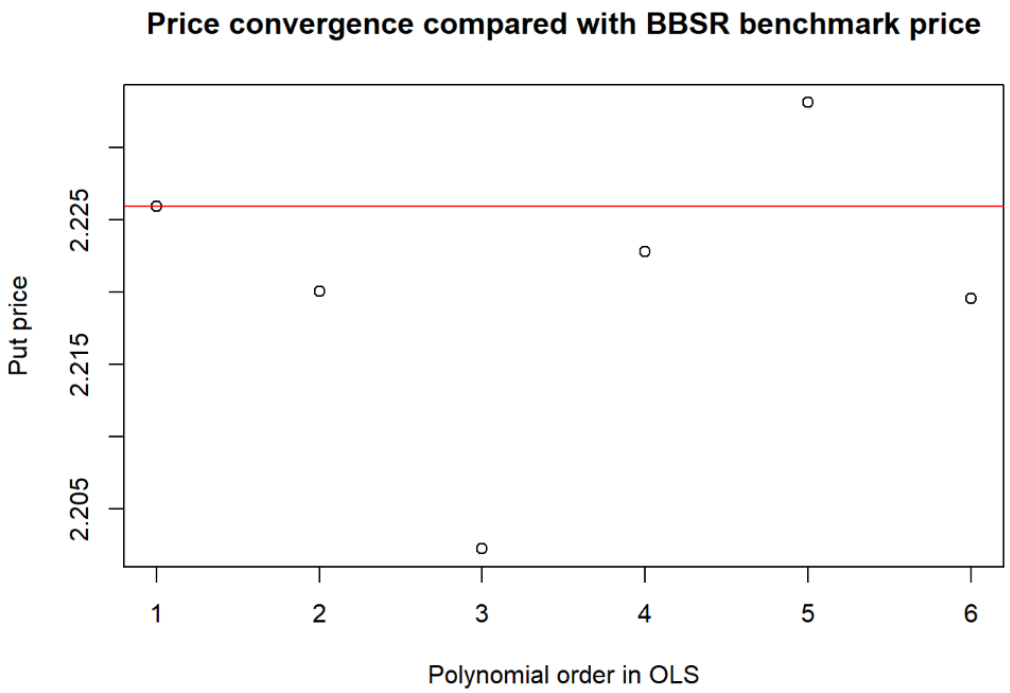

Figure k: The convergence of the estimated option price obtained from LSMC when varying the highest polynomial order when fitting OLS (K)

Table c: The standard errors of the estimated option prices obtained from LSMC when varying the highest polynomial order when fitting OLS (K)

the highest polynomial order when fitting OLS (K)	Standard Error
1	0.02110
2	0.02093
3	0.02086
4	0.02131
5	0.02163
6	0.02000

Although we do not observe any trend when m and K are varied, we observe some trend of all the prices predicted by LSMC, which is most of the predicted prices are equal or below the benchmark price. This observation is in line with what we already discussed in the LSMC session.

Conclusion

We study Monte Carlo simulation with variance reduction techniques, antithetic and control variate approaches, and Least Squares Monte Carlo (LSMC) to price European, American-style and Asian options. As the number of samples increases, the estimation of Monte-Carlo-based techniques that we study become more precise. The point estimations also gradually converge to their benchmark value. The antithetic and control variate approaches are effective in reducing the number of samples while keeping the predicted price confidence interval small. However, we pointed out the issue that the estimator when using the control variate approach may be biased although the standard error is significantly reduced if the number of samples to examine initial variables is too small. Lastly, we study LSMC, and observe that only the number of samples affects both the predicted prices and standard errors. The predicted prices converge to its benchmark price obtained from Binomial Black and Scholes method with Richardson extrapolation (BBSR), and the standard errors establish a reduction trend when the number of samples increase. We also observe that the majority of the predicted prices by LSMC approach are underpricing, which is in line with its mathematical theory.

References

[1] Broadie, M. and Detemple, J., 1996. American option valuation: new bounds, approximations, and a comparison of existing methods. *The Review of Financial Studies*, *9*(4), pp.1211-1250.
[2] Longstaff, F.A. and Schwartz, E.S., 2001. Valuing American options by simulation: a simple least-squares approach. *The review of financial studies*, *14*(1), pp.113-147.
[3] Karl Sigman, Introduction to reducing variance in Monte Carlo simulations, https://www.columbia.edu/~ks20/4703-Sigman/4703-07-Notes-ATV.pdf