Module 3 Exam Solutions

Koundinya Vajjha

May 2017

1 Introduction

Every time a Monte Carlo simulation is run, we get an estimate of the quantity we are trying to calculate. It is desirable to quantify how precise our estimate is. To do this, we can calculate the "standard error" of our Monte Carlo estimate.

Formally speaking, if X is our underlying random variable or a stochastic process (say the stock price) and g is a function of interest (say the payoff), we need to estimate $\mathbb{E}[g(X)]$ through Monte Carlo simulations. We do this by drawing N random price paths $x_1, ..., x_N$, calculating the mean $\overline{g} = \sum_{i=1}^{N} g_i$ and then declaring $\overline{g}$ to be our estimate for $\mathbb{E}[g(X]$. Now to get an idea of how precise our estimate is, we can use the Central Limit Theorem to come up with a "95% confidence interval" for our estimate:

$$\left(\overline{g} - 1.96 * \frac{s}{\sqrt{N}}, \overline{g} + 1.96 * \frac{s}{\sqrt{N}}\right)$$

Where

$$s^2 = \frac{1}{N-1} \sum_{i=1}^{N} (g(x_i) - \overline{g})$$

Now one obvious way to make our estimate more precise is to increase the number N of random price paths considered, as can easily be seen above. However this is computationally expensive. The goal of this assignment is to investigate different methods whereby the variance, or more precisely, the standard error of the Monte Carlo estimate is minimized. These are called Variance Reduction techniques. Variance reduction techniques increase the efficiency of Monte Carlo estimation.

This report is organized as follows.

1. I first consider the naive Monte Carlo method: simply simulating many price paths through the Euler-Maruyama scheme and calculating the option price as the discounted expected value of the payoff. This is done for both Vanilla European Call Options and also the Fixed and Floating Strike Asian Options with Arithmetic/Geometric Sampling. We find the standard error in each case and see how it behaves under increasing sample size.

2. According to Kemna and Vorst [1], using geometric sampling as a **control variate** results in a smaller standard error for the Asian Option. We investigate if this is true for our case. (The meaning of "control variates" will be explained in the relevant section below.)

3. Next we mention more advanced methods for variance reduction: namely the method of Dingec and Hormann [2], implemented in the 'OptionPricing' package in R, and also the method of "Importance Sampling" for Arithmetic Average Asian Options as described in the thesis by Ferstl [3].

The following data is used.

$$\begin{aligned} \text{Stock price } S_0 &= 100 \\ \text{Strike } E &= 100 \\ \text{Time to expiry } (T-t) &= 1 \text{ year} \\ \text{volatility } \sigma &= 20\% \\ \text{constant risk-free interest rate } r &= 5\% \end{aligned}$$

2 Naive Monte Carlo

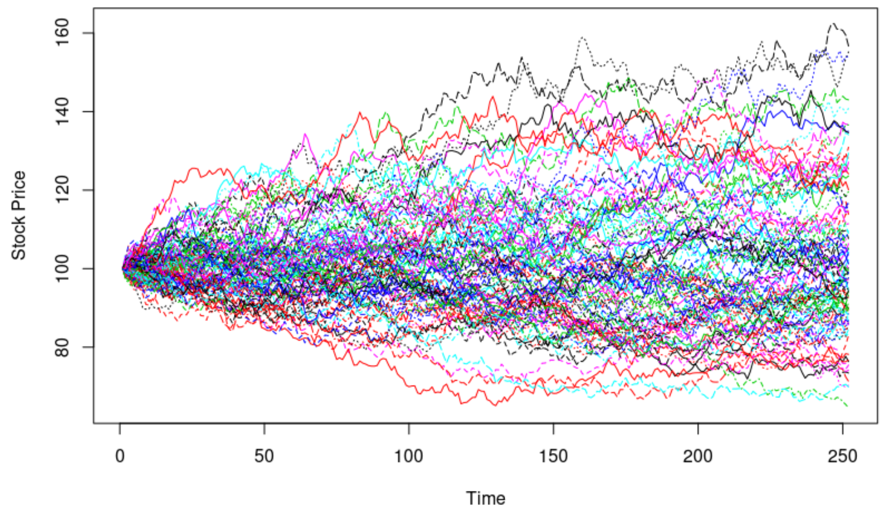
I first describe the naive Monte Carlo approach, where we simulate the price path using the Euler-Maruyama scheme. Here is the general outline of the algorithm used.

1. Set $S_0 = 100, \delta t = 1/252, r = 0.05, \sigma = 0.2$. Generate S_i for $i = 1, ..., 251$ by the following relation.

$$S_i = S_{i-1}(1 + r\delta t + \sigma\sqrt{\delta t}\phi)$$

   Where ϕ is a standard normal random variate, a new one is taken at each step.

2. The above step is repeated N times, so that we get N stock price realization vectors, each of length 252. We denote by $S_i^{(j)}$ the i^{th} day's simulation in the j^{th} realization. The realizations look as follows when plotted:



3. For each price path, we calculate the average value, given by the following formula:
$$\text{Average}_j = \frac{1}{N}\sum_{i=0}^{251} S_i^{(j)} \text{ for } j = 1, ..., N$$

4. Calculate the payoff for each realization for both fixed and floating strike:
$$\text{FixedPayoff}_j = \max\left(\text{Average}_j - E, 0\right) \quad (1)$$
$$\text{FloatPayoff}_j = \max\left(S_{251}^{(j)} - \text{Average}_j, 0\right) \quad (2)$$
for $j = 1, ..., N$.

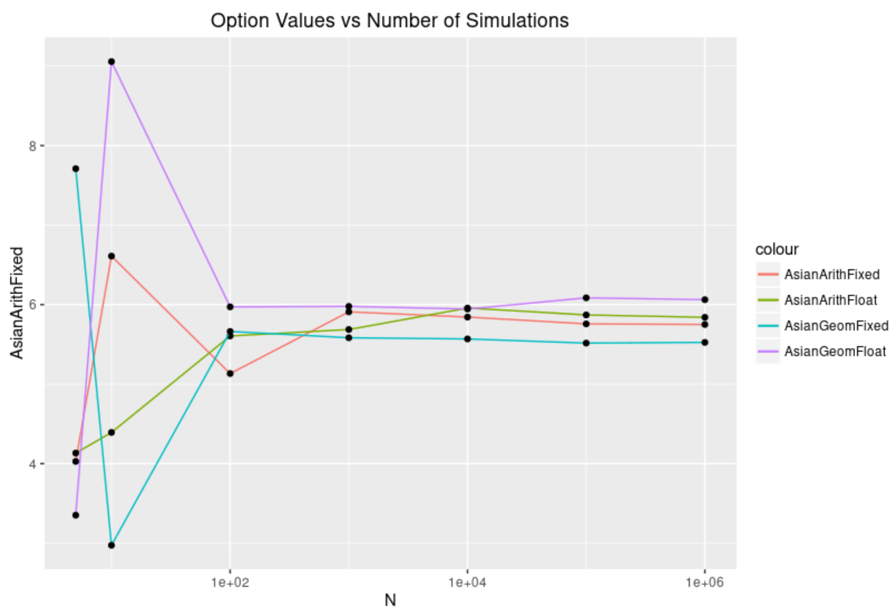
5. Calculate the discounted average of each of the above values over $j = 1, ..., N$:
$$e^{-r}\mathbb{E}_j[\text{FixedPayoff}]$$
or
$$e^{-r}\mathbb{E}_j[\text{FloatPayoff}]$$

6. Calculate the standard error of the vector of the N simulated values.

Now I present the results. Here is a list of calculated option values plotted against the number of simulations used.

N	ArithFixed	ArithFloat	GeomFixed	GeomFloat	VanillaEuro
5	4.03	4.13	7.71	3.35	12.90
10	6.61	4.39	2.97	9.06	11.19
100	5.13	5.61	5.66	5.97	12.16
1000	5.91	5.69	5.58	5.98	9.94
10000	5.84	5.96	5.57	5.94	10.45
100000	5.76	5.87	5.52	6.09	10.43
1000000	5.75	5.84	5.52	6.06	10.42

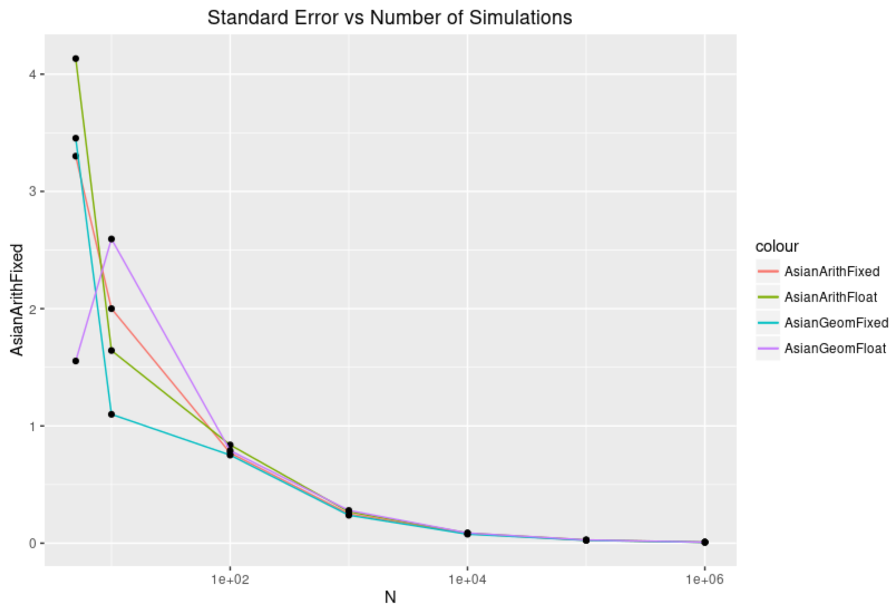
Table 1: Asian Option Values for Various Simulations.



Next have a list of the standard error of the calculations. Naturally, the larger the number of simulations, the more accurate the estimate.

N	ArithFixed	ArithFloat	GeomFixed	GeomFloat	VanillaEuro
5	3.30079	4.13284	3.45385	1.55370	3.43781
10	2.00005	1.64290	1.09878	2.59426	3.75396
100	0.76942	0.83790	0.75125	0.78906	1.59588
1000	0.25019	0.26769	0.23801	0.27927	0.44149
10000	0.08104	0.08596	0.07663	0.08695	0.14607
100000	0.02503	0.02683	0.02425	0.02777	0.04648
1000000	0.00795	0.00846	0.00767	0.00876	0.01467

Table 2: Standard Error for the Monte Carlo estimate for Various Simulations.



3 Using Control Variate Variance Reduction Method for Arithmetic Sampling Asian Options

While researching on this topic, I came across the paper by Kemna and Vorst [1]. While glancing through the results in that paper, I came across the geometric sampling and mistakenly assumed that a smaller standard error is achieved when using geometric sampling rather than arithmetic sampling. However, when I looked at the results in Table 2, I couldn't see any noticeable difference in the standard error.

It was only later that I realized that the authors meant that they used geometric sampling as a **control variate** for the Monte Carlo simulation and this resulted in the smaller variance. I set out to inspect if this was indeed the case. I shall now quickly outline the concept of a control variate and then proceed with the results.

Say desired simulation quantity is $\theta = \mathbb{E}[X]$, and there is another known random variable Y with known expectation $\mu_Y = \mathbb{E}[Y]$. Now for any c once can see easily that $Z = X - c(Y - \mu_Y)$ is an unbiased estimator of θ, by linearity of expectation. Now it can be shown that

$$\mathbb{V}(Z) = \mathbb{V}(X) + c^2\mathbb{V}(Y) + 2c\mathrm{Cov}(X, Y)$$

is minimized when $c = -\frac{\mathrm{Cov}(X,Y)^2}{\mathbb{V}(Y)}$. Y is called a **control variate** of X. To reduce variance, we choose Y correlated with X. In practice, we can choose a Y which is approximately equal to X, which is easy to simulate and whose mean is known. In our case, these requirements are satisfied perfectly when we take

X to be the Asian Option payoff with Arithmetic Sampling, and Y to be the Asian Option payoff with Geometric Sampling.

Here is the outline of the algorithm:

1. Set $S_0 = 100, \delta t = 1/252, r = 0.05, \sigma = 0.2$. Generate S_i for $i = 1, ..., 251$ by the following relation.

$$S_i = S_{i-1}\text{exp}\left(\left(r - \frac{\sigma^2}{2}\right)\delta t + \sigma \delta t \phi\right)$$

   Where ϕ is a standard normal random variate, a new one is taken at each step.

2. The above step is repeated N times, so that we get N stock price realization vectors, each of length 252. We denote by $S_i^{(j)}$ the i^{th} day's simulation in the j^{th} realization.

3. Calculate the following quantities for each j from 1 to N:

$$X_j = e^{-r}\left[\max\left(\frac{1}{N}\sum_{i=0}^{251} S_i^{(j)} - E, 0\right)\right]$$

$$Y_j = e^{-r}\left[\max\left(\left(\prod_{i=0}^{251} S_i^{(j)}\right)^{\frac{1}{N}} - E, 0\right)\right]$$

$$c = -\frac{\sum_{i=0}^{251}\left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sum_{i=0}^{251}\left(X_i - \overline{X}\right)^2}$$

$$d = \left(\ln(S_0/E) + \left(r + \frac{\sigma^2}{6}\right)T/2\right) / \left(\sigma\sqrt{T/3}\right)$$

$$\mu_Y = e^{-r}\left(e^{(r+\frac{\sigma^2}{6})\frac{T}{2}} S_0 \Phi(d) - E\Phi\left(d - \sigma\sqrt{\frac{T}{3}}\right)\right)$$

$$Z_j = X_j + c(Y_j - \mu_Y)$$

4. Finally, calculate the mean and standard error of all Z_j.

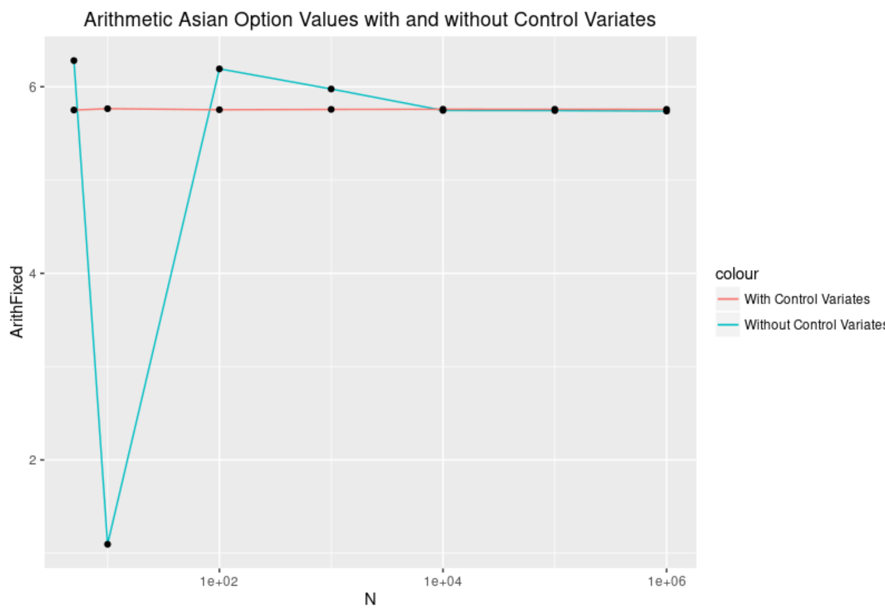
$$\text{Option Price } = \frac{1}{N}\sum_{j=1}^{N} Z_j$$

   and

$$\text{Standard Error } = \sqrt{\mathbb{V}(Z)/N}$$

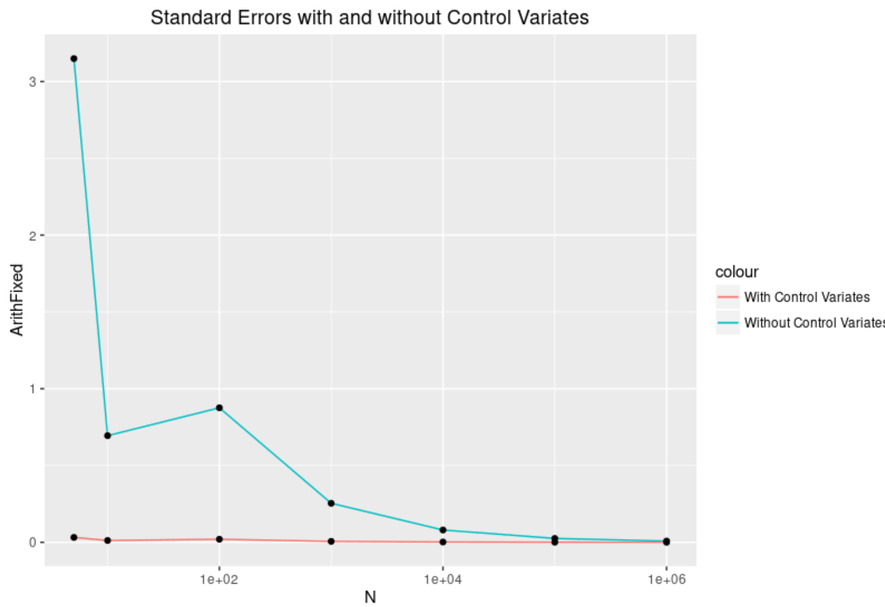
Now we look at the results.

N	ArithFixed	StdError	ArithFixedControl	StdErrorControl
5	6.2804064	3.149390423	5.7504092	0.0318102089
10	1.0957633	0.694199409	5.7647020	0.0118502141
100	6.1917248	0.875800169	5.7537722	0.0198204416
1000	5.9757576	0.253978571	5.7573382	0.0060653180
10000	5.7459812	0.079961232	5.7590264	0.0019584667
100000	5.7443366	0.025132075	5.7587144	0.0006240232
1000000	5.7391779	0.007945063	5.7577610	0.0001958552



One can see that with control variates, the Option value is almost constant at 5.75, whereas the naive implementation tends to jump around before converging. In the next page there is also a plot of the Standard Error. Both these plots show that the Control Variate method for reducing variance is very effective. The downsides are that an effective control variate may not be readily available. Apart from the control variate method, there is also another variance reduction technique called the method of **Antithetic variates**.

In this method, instead of considering N random numbers at each step of the Monte Carlo simulation, we consider $N/2$ random numbers, and take the rest to be the same random numbers except with the opposite sign. After taking the appropriate unbiased estimator from these paths, this clever trick reduces the variance and also the computation time by half.



4 Conclusion and Further methods.

I have seen that the Control Variates form a very effective variance reduction technique. For more advanced methods of variance reduction, one can look at the technique of **Importance Sampling**. There has been work done in this direction in the thesis of Ferstl [3]. Also, there is an R package by the name of 'OptionPricing' which implements an advanced variance reduction technique through 'effective control variates' and 'quasi-Monte Carlo simulations' and 'Koborov Lattices' [2]. I have used this package and the results are extremely fast and with a very low standard error.

References

[1] A.G.Z Kemna and A.C.F. Vorst, *A Pricing Method for Options Based on Average Asset Values*, Journal of Banking and Finance,14 (1990), 113-129,North-Holland

[2] K. D. Dingec and W. Hormann, *Improved Monte Carlo and Quasi-Monte Carlo Methods for the Price and the Greeks of Asian Options*, Proceedings of the 2014 Winter Simulation Conference.

[3] D.Ferstl *Pricing Asian Options by Importance Sampling*,Masters Thesis, Vienna University of Technology.