FE 621 Computational Methods in Finance: Lecture 5
Monte Carlo Simulation Method

1 Plain vanilla Monte Carlo method

You have discovered for yourself in the first assignment that some options are hard to price using even Binomial or Trinomial tree approximations.

With the Monte Carlo simulations you can price complex options and even depart from the regular geometric Brownian motion assumption for the stock price.

The general idea is to simulate many paths of the underlying asset price. For every such path say path i one can see what the simulated price of the option would be. All you have to do is look at the path i and, depending on the evolution of the underlying asset, calculate the value of the option at the maturity T. Call this value corresponding to path i as $C_{T,i}$. Then calculate the value of the option at the present time $t = 0$ by discounting back as: $C_i = \exp(-rT)C_{T,i}$.

The Monte Carlo option price will be just the average of all the values calculated for each path:

$$C = \frac{1}{M}\sum_{i=1}^{M} C_i,$$

where M is the number of paths you chose to simulate.

The question is, if this method is so simple then why are we not using it all the time? I will let you find the problems of this method.

Please read Section 4.2. about the details of using this method to simulate from the regular geometric Brownian motion.

Also read about standard deviation and standard error. To summarize in a short description of the Central Limit Theorem: assume that we are given

$X_1, X_2, \ldots, X_M$ values from some distribution with mean μ and standard deviation σ. Then the CLT says that the distribution of the average of these values denoted $\overline{X}$ is approximately normal with mean μ and standard deviation $\sigma/\sqrt{M}$. This is the result that makes the Monte Carlo method possible. Thus to approximate μ we use the average of the values $\overline{X}$ and to approximate the standard deviation σ we use:

$$\widehat{\sigma} = \sqrt{\frac{\sum_{i=1}^{M} \left(X_i - \overline{X}\right)^2}{M-1}}$$

Note that the formula in your book [1] on page 83 is the same with the one above, but when applied to an example on page 87 it contains an error.

To estimate the standard deviation of $\overline{X}$ we calculate what your book calls the standard error as: $\widehat{\sigma}/\sqrt{M}$.

To summarize: The precision of your estimate C is of the order $\hat{\sigma}/\sqrt{M}$. Even more precisely we can give a 95% confidence interval for the true option price when M is large as:

$$\left[C - 1.96\frac{\hat{\sigma}}{\sqrt{M}}, C + 1.96\frac{\hat{\sigma}}{\sqrt{M}}\right]$$

This is all from the CLT.

This standard error is a measure of how precise is your estimation. Ideally you would like a small standard error and consequently a precise confidence interval. However, this is not likely to happen with the plain Monte Carlo method.

2 Generating random numbers distributed as $N(0,1)$

In your Monte Carlo simulation you will have to generate many Brownian Motion paths. For this you will need a generator of increments which are normally distributed with mean 0 and standard deviation $\sqrt{\Delta t}$ where Δt is the time increment in your simulation. If you can generate standard normal random variables (with mean 0 and standard deviation 1) say Z, it is a simple matter to make it normal with mean 0 and standard deviation $\sqrt{\Delta t}$ by just taking $X = \sqrt{\Delta t} Z$.

To generate the standard normal variables use any method you wish. You can use libraries or included functions in the software if they exist. Alternatively, you can use either of the two methods presented in your book [1] in the Section 4.11: the Box-Muller transformation or the polar rejection method. Section 4.12 has a nice discussion of the pseudo-random numbers (uniform numbers generated by any computer algorithm) versus quasi-random numbers. Please read this section.

When you will turn in your homework please specify which method you used to generate random numbers.

3 Extension to more general processes

Monte Carlo technique is the first approximation method we study that is applicable to more general processes (not only the geometric BM). One simply generates paths from this more complex process and then calculates the option price exactly as in Section 1. The difficulty lies in generating paths following the process one wants to approximate. For more details about this simulation please see Lecture 5Xtra: Simulating SDE's using Euler method.

4 Variance Reduction

We have seen earlier that one of the problems with the Monte Carlo simulations is the big variance (standard error) when computing the average of the simulated values. This variance can be decreased by increasing the number of simulated values, however this will also increase the run time of the whole simulation.

An alternative approach is to use variance reduction techniques, two of which your book discuss in detail in sections 4.3-4.5. I will present a brief summary of these sections.

4.1 Antithetic variates

The main idea of this technique is to look at the asset equation that you are trying to simulate:

$$dS_t = rS_t dt + \sigma S_t dz_t$$

and recognize that since z_t is a standard Brownian motion so will be $-z_t$ and they will have the same exact distribution. This means that the equation:

$$dS_t = rS_t dt - \sigma S_t dz_t$$

will also generate paths of the same asset.

So, how is this fact used in practice? When you generate the increments for the Brownian motion in your simulation of the paths say $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$ you will use these increments to generate one path and also use the negative of these increments $-\varepsilon_1, -\varepsilon_2, \ldots, -\varepsilon_n$ to generate an antithetic path. Both of these paths are simulating the same asset, thus you can use them to find two instances of the call option. Then, when using both in the calculation of the final Monte Carlo value the variance of the estimate will be reduced. Not only you generate $2M$ values instead of M with the plain Monte Carlo method, but there are also theoretical reasons why the variance is smaller than if you would generate $2M$ plain Monte Carlo simulations.

4.2 Control Variates

For this part you will have to know the details of the Delta hedging so please read Hull's textbook [2] for details. Please read carefully sections 4.4 and 4.5 in [1] and consult Hull's book as well.

Delta hedging can be summarized succinctly in the following way: Suppose that at time $t = 0$ we receive C_0 the price of an option that pays C_T at time T. The price of this option at any time t is a function $C(t, S)$. Then, if we hold at any moment in time $\frac{\partial C}{\partial S}(t, S) = \frac{\partial C_t}{\partial S}$ units of stock then we will be able to replicate the payout of this option C_T at time T. This is in theory since of course we cannot trade continuously. So in practice we perform a partial hedge where we only recalibrate at some discrete moments in time say $t_1, t_2, \ldots, t_N$. For simplicity of writing the formulas we take $t_i = i\Delta t$ equidistant in time and $t_N = N\Delta t = T$. Table 1 writes the adjustments we make at all of these times. We use the notations $\frac{\partial C_t}{\partial S} = \frac{\partial C}{\partial S}(t, S)$ and $S_i = S_{t_i} = S_{i\Delta t}$

So looking at the table if we bring all the money in their value at time T by multiplying each row with the corresponding factor $e^{(T-i\Delta t)r}$ and we take the received money with plus and the money we pay with minus we should obtain approximately a zero sum. Thus we should have:

$$C_0 e^{rT} - \frac{\partial C_0}{\partial S} S_0 e^{rT} + \frac{\partial C_0}{\partial S} S_1 e^{r(T-\Delta t)} - \frac{\partial C_1}{\partial S} S_1 e^{r(T-\Delta t)} + \cdots + \frac{\partial C_{N-1}}{\partial S} S_N - C_T = \eta$$

Table 1: Calibrating the Delta hedging

At time	Receive from before	Pay to hold $\frac{\partial C}{\partial S}(t, S)$ units of stock
$t_0 = 0$	C_0	$\frac{\partial C_0}{\partial S} S_0$
$t_1 = \Delta t$	$\frac{\partial C_0}{\partial S} S_1$	$\frac{\partial C_1}{\partial S} S_1$
$t_2 = 2\Delta t$	$\frac{\partial C_1}{\partial S} S_2$	$\frac{\partial C_2}{\partial S} S_2$
$\vdots$	$\vdots$	$\vdots$
$t_{N-1} = (N_1)\Delta t$	$\frac{\partial C_{N-2}}{\partial S} S_{N-1}$	$\frac{\partial C_{N-1}}{\partial S} S_{N-1}$
$t_N = T$	$\frac{\partial C_{N-1}}{\partial S} S_N$	C_T

where η is the pricing error. We rewrite the expression above in the following way (we factor similar derivatives):

$$C_0 e^{rT} + \sum_{i=0}^{N-1} \frac{\partial C_i}{\partial S} \left(S_{i+1} - S_i e^{r\Delta t}\right) e^{r(T-(i+1)\Delta t)} = C_T + \eta$$

Thus, rewriting we obtain:

$$C_0 = C_T e^{-rT} - \left(\sum_{i=0}^{N-1} \frac{\partial C_i}{\partial S} \left(S_{i+1} - S_i e^{r\Delta t}\right) e^{r(T-(i+1)\Delta t)} \right) e^{-rT} + \eta e^{-rT} \quad (1)$$

The sum in parenthesis in equation (1) represents the control variate. The error η is further multiplied with a small number and is going to be averaged in the end so it is neglected.

So, how do we use this formula? In order to use it we need an equation for the dynamics of S_t. Supposedly, we have such equation and we use the

Euler method for example to produce a path of prices $S_0, S_1, \ldots, S_N$. The payoff value C_T can be determined based on the sequence of prices.

BUT, and this is a big but, in order to use this formula we need also to have some means to calculate or approximate $\frac{\partial C_i}{\partial S}$ for any time t_i. For complex price dynamics this task is hopeless (it is much harder than finding the price of the option itself). Back to the Black-Scholes world this can be accomplished because we have a formula for the price of the option at any moment in time.

Exemplification for the European Call in Black Scholes world. Recall the Black-Scholes formula for European call option when the Stock price is modeled as a geometric Brownian motion:

$$C(S,t) = SN(d_1) - Ke^{-r(T-t)}N(d_2), \tag{2}$$

where

$$\begin{aligned} d_1 &= \frac{\ln\left(\frac{S}{K}\right) + \left(r + \frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{T-t}} \\ d_2 &= d_1 - \sigma\sqrt{T-t} = \frac{\ln\left(\frac{S}{K}\right) + \left(r - \frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{T-t}}. \end{aligned}$$

Note that d_1 and d_2 are themselves functions of S thus we need to apply the chain rule when calculating the derivative $\frac{\partial C}{\partial S}(t,S)$. Note that we have:

$$\begin{aligned} \frac{\partial d_1}{\partial S} &= \frac{1}{\sigma\sqrt{T-t}}\frac{1}{S} \\ \frac{\partial d_1}{\partial S} &= \frac{\partial d_1}{\partial S} = \frac{1}{S\sigma\sqrt{T-t}}. \end{aligned}$$

Thus:

$$\begin{aligned} \frac{\partial C}{\partial S}(t,S) &= N(d_1) + SN'(d_1)\frac{\partial d_1}{\partial S} - Ke^{-r(T-t)}N'(d_2)\frac{\partial d_2}{\partial S} \\ &= N(d_1) + \left(SN'(d_1) - Ke^{-r(T-t)}N'(d_2)\right)\frac{1}{S\sigma\sqrt{T-t}} \end{aligned}$$

In the formula above $N(x)$ is the CDF of a normal (0,1) calculated at x and $N'(x)$ is the pdf of a normal calculated at x, i.e.:

$$N'(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}.$$

Thus, at time say $i\Delta t$ one plugs into the formula above $t = i\Delta t$ and $S = S_{i\Delta t}$, the stock price generated according to the current path.

The Control Variates method is supposedly much faster than any other Monte Carlo method. One can use also a Gamma variate methodology but I will not bother explain it here (it is very similar with the above).

5 Hedge sensitivities and Multiple asset valuation

Please finish the chapter by reading sections 4.6 and 4.7 in your textbook.

In particular read carefully 4.7 since it contains examples of interest for practical applications.

Finally 4.8 and 4.9 describe in detail ways of pricing path dependent options using Monte Carlo simulations.

References

[1] L. Clewlow and C. Strickland. *Implementing Derivatives Models*. Wiley, 1998.

[2] John C. Hull. *Options, Futures and Other Derivatives*. Prentice Hall, 6 edition, 2005.