TEACHING NOTE 96-03:

MONTE CARLO SIMULATION

Version date: July 18, 2008 C:\CLASSES\TEACHING NOTES\TN96-03.DOC

Simulation is a procedure in which random numbers are generated according to probabilities assumed to be associated with a source of uncertainty, such as a new product's sales or, more appropriately for our purposes, stock prices, interest rates, exchange rates or commodity prices. Outcomes associated with these random drawings are then analyzed to determine the likely results and the associated risk. Oftentimes this technique is called Monte Carlo simulation, being named for the city of Monte Carlo, which is noted for its casinos.

The gambling analogy notwithstanding, Monte Carlo simulation is a legitimate and widely used technique for dealing with uncertainty in many aspects of business operations. For our purposes, it has been shown to be an accurate method of pricing options and particularly useful for path-dependent options and others for which no known formula exists.

To facilitate an understanding of the technique, we shall look at how Monte Carlo simulation has been used to price standard European options. Of course, we know that the Black-Scholes model is the correct method of pricing these options so Monte Carlo simulation is not really needed. It is useful, however, to conduct this experiment because it demonstrates the accuracy of the technique for a simple option of which the exact price is easily obtained from a known formula.

The assumptions of the Black-Scholes model imply that for a given stock price at time t, simulated changes in the stock price at a future time t +.t can be generated by the following formula:

$$\Delta S = S r_c \Delta t + S\sigma\varepsilon\sqrt{\Delta t}$$

where S is the current stock price,)S is the change in the stock price, r_c is the continuously compounded risk-free rate, Φ is the volatility of the stock and)t is the length of the time interval over which the stock price change occurs. The variable , is a random number generated from a standard normal probability distribution. Recall that the standard normal random variable has a mean of zero, a standard deviation of 1.0 and occurs with a frequency corresponding to that associated with the famous bell shaped curve.

Generating future stock prices according to the above formula is actually quite easy. A standard normal random variable can be approximated with a slight adjustment to Excel's Rand() function. The Rand() function produces a uniform random number between 0 and 1, meaning that it generates numbers between 0 and 1 with equal probability. A good approximation for a standard normal variable is obtained by the Excel formula "= Rand() + Rand() + Rand() + Rand() + Rand() + Rand() + Rand() + Rand() + Rand() + Rand() + Rand() + Rand() - 6.0", or simply 12 uniform random numbers minus 6.0.[1] Excel also has a random number generator in its Data Analysis Tools.

After generating one standard normal random variable, you then simply insert it into the right hand side of the above formula for)S. This gives the price change over the life of the option, which is then added to the current price to obtain the price of the asset at expiration. You then compute the price of the option at expiration according to the standard formulas, Max(0,S_T - X) for a call or Max(0,X - S_T) for a put, where X is the exercise price and S_T is the asset price at expiration. This produces one possible option value at expiration. You then repeat this procedure many thousands of times, take the average value of the call at expiration and discount that value at the risk-free rate. Some users compute the standard deviation of the call prices in order to obtain a feel for the possible error in estimating the price.

Let us price a call option. The stock price is 164, the exercise price is 165, the risk-free rate is 0.0521, the volatility is 0.29 and the time to expiration is 0.0959. Inserting the above approximation formula for a standard normal random variable in any cell in an Excel spreadsheet produces a random number. Suppose that number is 0.733449. Inserting into the formula for)S gives $164(.0521)(.0959) + 164(.29)(.733449)\sqrt{.0959} = 11.62$. Thus, the simulated value of the stock at expiration is 164 + 11.62 = 175.62. At that price, the option will be worth Max(0, 175.62 - 165) = 10.62 at expiration. We then draw another random number. Suppose we get -0.18985. Inserting into the formula for)S, we obtain $164(.0521)(.0959) + 164(.29)(-.18985)\sqrt{.0959} = -1.98$, which gives us a stock price at expiration of 164 -1.98 = 162.02, leading to an option price of Max(0, 162.02 - 165) = 0. We repeat this procedure several thousand times, after which we take an average

---

[1] This approximation is based on the fact that the distribution of the sum of twelve uniformly distributed random numbers between 0 and 1 will have a mean of six and a standard deviation of 1. By subtracting 6.0, we adjust the mean to zero without changing the standard deviation. What we obtain is technically not normally distributed but is symmetric with mean zero and standard deviation of 1.0, which are three properties associated with the normal distribution. The procedure is widely accepted as a quick and reasonable approximation but might not pass the most demanding tests for normality. On the other hand, many other random number generators would also not pass the most demanding tests either. If you generate a large enough number of random draws, my experience has been that the procedure converges nicely to the Black-Scholes model, which gives a great deal of confidence in it.

of the simulated option prices and then discount that average to the present using the present value formula $e^{-.0521(.0959)}$.

Naturally every simulation is different because each set of random numbers is different. A Monte Carlo procedure written in Excel's Visual Basic produced the following values for this call, whose actual Black-Scholes price is 5.79, where the number of random drawings is the sample size, n.

n	Call Price
1,000	5.58
10,000	5.51
50,000	5.83
100,000	5.75

It would appear that a sample of at least 50,000 is required for the simplest case of a standard European option.

The option price obtained from a Monte Carlo simulation is a sample average. Thus, its standard deviation is the standard deviation of the sample divided by the square root of the sample size.[2] Consequently, the error reduces at the rate of 1 over the square root of the sample size. Notice what this means for increasing the sample accuracy by increasing the sample size. Suppose Φ is the standard deviation of the sample. We first conduct a Monte Carlo simulation using n_1 random drawings. Since the option value is a sample mean, the standard deviation of our estimate of the option value is $\sigma/\sqrt{n_1}$. Now suppose we wanted to reduce that standard deviation in half. How much larger must the sample be? Let this new sample size be n_2. Then its standard deviation of the estimate of the option price is $\sigma/\sqrt{n_2}$. Now note that,

$$\frac{1}{2}\frac{\sigma}{\sqrt{n_1}} = \frac{\sigma}{\sqrt{n_2}}, \text{if and only if}$$
$$n_2 = 4n_1.$$

Thus, to achieve a 50% reduction in error, i.e., a 50% increase in accuracy, we must quadruple the number of random drawings. That is, the standard error reduces only at the rate of the square root of

[2]Although this is a point from the most elementary principles of statistics, it is worthwhile to distinguish between the standard deviation of a sample and the standard deviation of the sample mean. The latter is the former divided by the square root of the sample size. Hence, the sample mean is much less volatile than the sample values themselves.

the sample size, not at the rate of the sample size itself.

With that in mind, it behooves the user of Monte Carlo simulation to attempt to achieve greater accuracy through other means. One method of doing so is quite simple and automatically doubles the sample size with only a minimum increase in computational time. This is called the *antithetic variate* method. Remember that in the standard Monte Carlo simulation, we are generating observations of a standard normal random variable. The standard normal random variable is distributed with a mean of zero, a variance of 1.0, and is symmetric. Thus, for each value we draw, there is an equally likely chance of having drawn the observed value times -1. Consequently, for each value of ˛ we draw, we can legitimately create an artificially observed companion observation of -,. This is the antithetic variate. We simply use it the same way we use the value we drew, i.e., in computing an asset price change from which we compute an option price. This procedure automatically doubles our sample size without having increased the number of random drawings.

In the case of Black-Scholes, which converges rapidly in a simulation, this may not matter that much. Shown below is a set of simulations using the standard and antithetic variate methods:

	Call Price	
n	Standard Monte Carlo	Antithetic Variate
1,000	5.87	5.70
10,000	5.73	5.69
50,000	5.83	5.79
100,000	5.75	5.78

The advantages of the antithetic variate method are likely to be greater the smaller the sample size.

Another method that can be used with certain types of options is called the *control variate* method. A control variate in this context is a somewhat similar option whose true value is known. We then obtain a simulated value of that option. The difference between the true value of the control variate and its simulated value is then added to the simulated value of the option we are trying to price. In this manner, the error in the control variate is added to the simulated value of the option of interest. Let us see how this method works.

Let c_s be the simulated price of the option we are trying to price. Let v_t be the true value of another similar option and v_s be its simulated value. Our control variate estimate is then found as

$$c_s + (v_t - v_s).$$

What we are doing is running a simulation of c_s - v_s and adding v_t. Note the following result:

$$\mathrm{Var}(c_s - v_s) = \mathrm{Var}(c_s) + \mathrm{Var}(v_s) - 2\mathrm{cov}(c_s,v_s).$$

This will be less than $\mathrm{Var}(c_s)$ if $\mathrm{Var}(v_s) < 2\mathrm{cov}(c_s,v_s)$, meaning that the control variate method relies on the assumption of a large covariance between c_s and v_s. The control variate chosen should be one that is very highly correlated with the option we are pricing.

Applying the Monte Carlo technique to more complicated options such as path-dependent options requires a partitioning of the option's life into time periods, as in the binomial model. For example, suppose you wanted to price an Asian option, which is an option in which the terminal asset price is replaced with the average price of the asset over a period of time. Let us say that the contract terms stipulate that the average price be computed by collecting the daily closing price over the life of the option. Ignoring holidays and weekends, let us say that the option has a 90 day life. Then a run would consist of 90 random drawings, each used to simulate the stock price at the end of each of the 90 days. The formula for each)S would be based on the previous day's closing price. The value of)t would be 1/365. Then the average of the 90 stock prices would determine the option payoff at expiration. You would then need to repeat the procedure at least 50,000 times. The more complex options, however, would probably require at least 100,000 runs.

Asian options are ideally suited for the control variate technique. Although Asian options are common in practice, the average is nearly always computed arithmetically. It is possible, however, to structure Asian options so that they pay off based on the geometric average, which is the j^{th} root of the product of the j prices used to compute the average. For a geometric average price option, there is indeed a closed-form solution for the option price. A common approach to pricing an arithmetically averaged Asian option is to use the geometrically averaged Asian option as the control variate.

We shall not get into the details of Asian options in this document, but we shall show the results of a Monte Carlo simulation using the standard method, the control variate method and the antithetic variate method for our example option, now in Asian form with 50 prices used in calculating the average. The following results are for a put options with the same terms as the call.

		Call Price	
n	Monte Carlo	Control Variate	Antithetic Variate

1,000	3.33	3.11	3.14
10,000	3.19	3.11	3.16
50,000	3.16	3.11	3.17
100,000	3.16	3.11	3.16

Differences in prices obtained using the control variate method do not show until several digits past the decimal. Note, however, that in this case, the control variate method does not appear to help much. For small samples, however, it can be of great help. Note that it appears to be more accurate than the standard approach for the small sample. Of course for large samples, the standard approach is accurate and, naturally, is theoretically correct in the limit. In such cases, these improvements to the standard Monte Carlo method may not help much.

There are many more methods for improving the accuracy of Monte Carlo simulation. Some methods improve the sampling process so that random but unusual results are balanced with more typical outcomes. These methods belong to a family called *low discrepancy sequences*. For example, if a random sample is generated, it is quite possible that numbers that ought to appear with a given frequency within a specific range might not appear with the expected frequency. Some common techniques from this family are *Halton sequences* and *Sobol numbers* are used to supply random outcomes that result in the sample of random numbers being much closer to a truly random sample with observed outcomes as expected.

References

The original work on Monte Carlo simulation in option pricing was

Boyle, Phelim P. "Options: A Monte Carlo Approach." *Journal of Financial Economics* 4 (May, 1977), 323-338.

Research on Monte Carlo methods in option pricing is a major topic in the literature today. See for example,

Lehoczky, J. P. "Simulation Methods for Option Pricing," Chapter 26 in *Mathematics of Derivative Securities*, ed. M. A. H. Dempster and S. Pliska. Cambridge: Cambridge University Press (1997).

Newton, N. J. "Continuous-Time Monte Carlo Methods for Variance Reduction," in *Numerical Methods in Finance*, ed. L. C. G. Rogers and D. Talay. Cambridge: Cambridge University Press (1997).

An excellent reference containing many useful articles is

Dupire, B. *Monte Carlo: Methodologies and Applications for Pricing and Risk Management*. London: Risk Books (1998).

One article in particular that gives an excellent overall survey is

Boyle, P., M. Broadie, and P. Glasserman. "Monte Carlo Methods for Security Pricing." *Monte Carlo Methods for Security Pricing* 21 (1997), 1267-1321.