INTRODUCTION TO MONTE CARLO SIMULATION

Samik Raychaudhuri

Oracle Crystal Ball Global Business Unit
390 Interlocken Crescent, Suite 130
Broomfield, C.O. 80021, U.S.A.

ABSTRACT

This is an introductory tutorial on Monte Carlo simulation, a type of simulation that relies on repeated random sampling and statistical analysis to compute the results. In this paper, we will briefly describe the nature and relevance of Monte Carlo simulation, the way to perform these simulations and analyze results, and the underlying mathematical techniques required for performing these simulations. We will present a few examples from various areas where Monte Carlo simulation is used, and also touch on the current state of software in this area.

1 INTRODUCTION

Monte Carlo simulation is a type of simulation that relies on repeated random sampling and statistical analysis to compute the results. This method of simulation is very closely related to random experiments, experiments for which the specific result is not known in advance. In this context, Monte Carlo simulation can be considered as a methodical way of doing so-called *what-if* analysis. We will emphasis this view throughout this tutorial, as this is one of the easiest ways to grasp the basics of Monte Carlo simulation.

We use mathematical models in natural sciences, social sciences, and engineering disciplines to describe the interactions in a system using mathematical expressions (Wikipedia 2008c). These models typically depend on a number of input parameters, which when processed through the mathematical formulas in the model, results in one or more outputs. A schematic diagram of the process is shown in Figure 1.
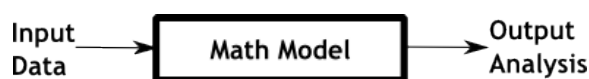

Figure 1: Mathematical models.

The input parameters for the models depend on various external factors. Because of these factors, realistic models are subject to risk from the systematic variation of the input parameters. A deterministic model, which does not consider these variations, is often termed as a *base case*, since the values of these input parameters are their most likely values. An effective model should take into consideration the risks associated with various input parameters. In most circumstances, experimenters develop several versions of a model, which can include the base case, the best possible scenario, and the worst possible scenario for the values of the input variables (Figure 2).
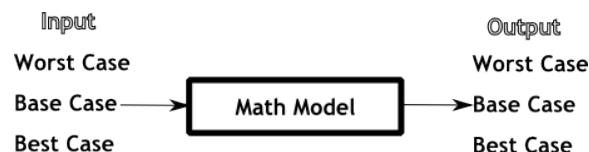

Figure 2: Case-based modeling.

This approach has various disadvantages. First, it might be difficult to evaluate the best and worst case scenarios for each input variable. Second, all the input variables may not be at their best or worst levels at the same time. Decision making tends to be difficult as well, since now we are considering more than one scenario. Also, as an experimenter increases the number of cases to consider, model versioning and storing becomes difficult. An experimenter might be tempted to run various ad-hoc values of the input parameters, often called *what-if* analysis, but it is not practical to go through all possible values of each input parameter. Monte Carlo simulation can help an experimenter to methodically investigate the complete range of risk associated with each *risky* input variable.

In Monte Carlo simulation, we identify a statistical distribution which we can use as the source for each of the input parameters. Then, we draw random samples from each distribution, which then represent the values of the input

variables. For each set of input parameters, we get a set of output parameters. The value of each output parameter is one particular outcome scenario in the simulation run. We collect such output values from a number of simulation runs. Finally, we perform statistical analysis on the values of the output parameters, to make decisions about the course of action (whatever it may be). We can use the sampling statistics of the output parameters to characterize the output variation.

The remainder of this paper is arranged as follows. In the next section, we start with a few terms which are associated with Monte Carlo simulation. In section 3, we discuss the general methodology for performing Monte Carlo simulation analysis. Next, we discuss each of the steps separately. In section 4, we discuss how to identify input distributions from historical data. That is followed by discussions on generating random variates in section 5 and analyzing output of Monte Carlo simulation in section 6. We also discuss various application areas for Monte Carlo simulation in section 7 and software for performing Monte Carlo simulation in section 8, before concluding in section 9.

2 TERMINOLOGIES

In this section, we discuss a few terms which are used in the context of Monte Carlo simulation.

Statistical distributions Statistical distributions or *probability distributions* describe the outcomes of varying a random variable, and the probability of occurrence of those outcomes. When the random variable takes only discrete values, the corresponding probability distributions are called discrete probability distributions. Examples of this kind are the binomial distribution, Poisson distribution, and hypergeometric distribution. On the other hand, when the random variable takes continuous values, the corresponding probability distributions are called continuous probability distributions. Examples of this kind are normal, exponential, and gamma distributions.

Random sampling In statistics, a finite subset of individuals from a population is called a *sample*. In random sampling, the samples are drawn at random from the population, which implies that each unit of population has an equal chance of being included in the sample.

Random number generator (RNG) A random number generator is a computational or physical device designed to generate a sequence of numbers that appear to be independent draws from a population, and that also pass a series of statistical tests (Law and Kelton 2000). They are also called *Pseudo-random number generators*, since the random numbers generated through this method are not actual, but simulated. In this article, we will consider RNG's which generate random numbers between 0 and 1, also called uniform RNG's.

3 METHODOLOGY

The following steps are typically performed for the Monte Carlo simulation of a physical process.

Static Model Generation Every Monte Carlo simulation starts off with developing a deterministic model which closely resembles the real scenario. In this deterministic model, we use the most likely value (or the base case) of the input parameters. We apply mathematical relationships which use the values of the input variables, and transform them into the desired output. This step of generating the static model closely resembles the schematic diagram in Figure 1.

Input Distribution Identification When we are satisfied with the deterministic model, we add the risk components to the model. As mentioned before, since the risks originate from the stochastic nature of the input variables, we try to identify the underlying distributions, if any, which govern the input variables. This step needs historical data for the input variables. There are standard statistical procedures to identify input distributions, which we discuss in section 4.

Random Variable Generation After we have identified the underlying distributions for the input variables, we generate a set of random numbers (also called random variates or random samples) from these distributions. One set of random numbers, consisting of one value for each of the input variables, will be used in the deterministic model, to provide one set of output values. We then repeat this process by generating more sets of random numbers, one for each input distribution, and collect different sets of possible output values. This part is the core of Monte Carlo simulation. We will discuss this step in detail in section 5.

Analysis and Decision Making After we have collected a sample of output values in from the simulation, we perform statistical analysis on those values. This step provides us with statistical confidence for the decisions which we might make after running the simulation. We will discuss this step briefly in section 6.

4 IDENTIFICATION OF INPUT DISTRIBUTION

In this section, we will discuss the procedure for identifying the input distributions for the simulation model, often called distribution fitting. When there are existing historical data for a particular input parameter, we use numerical methods to fit the data to one theoretical discrete or continuous distribution. Fitting routines provide a way to identify the most suitable probability distribution for a given set of data. Each probability distribution can be uniquely identified by its parameter set, so, distribution fitting is essentially the same as finding the parameters of a distribution that would generate the given data in question. From this perspective, fitting routines are nothing but nonlinear optimization problems,

where the variables are parameters of the distributions. There are a few standard procedures for fitting data to distributions. We will discuss them briefly in the following sections.

4.1 Methods for Distribution Fitting

4.1.1 Method of Maximum Likelihood (ML)

The following discussion is a summary of the article at Wikipedia (Wikipedia 2008b). ML estimation (MLE) is a popular statistical method used to make inferences about parameters of the underlying probability distribution from a given data set. If we assume that the data drawn from a particular distribution are independent and identically distributed (iid), then this method can be used to find out the parameters of the distribution from which the data are most likely to arise. For a more detailed analysis, refer to (Law and Kelton 2000, Cohen and Whitten 1988).

Let θ be the parameter vector for f, which can be either a probability mass function (for discrete distributions) or a probability density function (for continuous distributions). We will denote the pdf/pmf as f_θ. Let the sample drawn from the distribution be $x_1, x_2, \ldots, x_n$. Then the likelihood of getting the sample from the distribution is given by the equation (1).

$$L(\theta) = f_\theta(x_1, x_2, \ldots, x_n | \theta) \qquad (1)$$

This can be thought of as the joint probability density function of the data, given the parameters of the distribution. Given the independence of each of the datapoints, this can be expanded to the equation (2).

$$L(\theta) = \prod_{i=1}^{n} f_\theta(x_i | \theta) \qquad (2)$$

In MLE, we try to find the value of θ so that the value of $L(\theta)$ can be maximized. Since this is a product of probabilities, we conveniently consider the log of this function for maximization, hence the term 'loglikelihood'. So, the MLE method can be thought of as a nonlinear unconstrained optimization problem as given below in equation (3):

$$\max LL(\theta) = \sum_{i=1}^{n} \ln f_\theta(x_i | \theta), \quad \theta \in \Theta \qquad (3)$$

Here, Θ represents the domain of each of the parameter of the distribution. For some distributions, this optimization problem can be theoretically solved by using differential (partial differential, if there are more than one parameter) equations w.r.t. the parameters and then solving them.

Advantages and Disadvantages: MLE method is by far the most used method for estimating the unknown parameters of a distribution. It has certain advantages.

- Although the bias of ML estimators can be substantial, MLE is asymptotically unbiased, i.e., its bias tends to zero as the number of samples increases to infinity.
- The MLE is asymptotically efficient, which means that, asymptotically, no unbiased estimator has lower mean squared error than the MLE.

4.1.2 Method of Moments (ME)

The method of moments is a method of estimating population parameters such as mean, variance, median, and so on (which need not be moments), by equating sample moments with unobservable population moments (for which we will have theoretical equations) and then solving those equations for the quantities to be estimated. For a detailed discussion, see (Cohen and Whitten 1988).

Advantages and Disadvantages: MLE method is considered a better method for estimating parameters of distributions, because ML estimates have higher probability of being close to the quantities to be estimated. However, in some cases, the likelihood equations may be intractable, even with computers, whereas the ME estimators can be quickly and easily calculated by hand. Estimates by ME may be used as the first approximation to the solutions of the likelihood equations. In this way the method of moments and the method of maximum likelihood are symbiotic. Also, in some cases, infrequent with large samples but not so infrequent with small samples, the estimates given by the method of moments are outside of the parameter space; it does not make sense to rely on them then. That problem never arises in the method of maximum likelihood.

4.1.3 Nonlinear Optimization

We can also use nonlinear optimization for estimating the unknown parameters of a distribution. The decision variables are typically the unknown parameters of a distribution. Different objective functions can be used for this purpose, such as: minimizing one of the goodness-of-fit statistics, minimizing the sum-squared difference from sample moments (mean, variance, skewness, kurtosis), or minimizing the sum-squared difference from the sample percentiles (or quartiles or deciles). Additional constraints can be added to facilitate the formulation of the optimization problem. These constraints can be constructed from the relations between distribution parameters. This method is typically less efficient, and often takes more time. The value of the parameter depends on the algorithm chosen to solve the nonlinear optimization problem.

4.2 Goodness-Of-Fit Statistics

Goodness-of-fit (GOF) statistics are statistical measures that describe the correctness of fitting a dataset to a distribution. Other than visual indications through graphs, like p-p plots or q-q plots (Law and Kelton 2000), these are mostly used by various software to automate the decision of choosing the best fitting distribution. In this section, we will discuss three of the most common GOF statistics used. For more information on these statistics, refer to (D'agostino and Stephens 1986, Law and Kelton 2000).

4.2.1 Chi-square Test

The Chi-square test can be thought of as a formal comparison of a histogram of the data with the density or mass function of the fitted distribution. To compute the chi-square test statistic in either the continuous or discrete case, we must first divide the range of the fitted distribution into k adjacent intervals $[a_0, a_1), [a_1, a_2), \ldots, [a_{k-1}, a_k)$. It is possible that $a_0 = -\infty$ or $a_k = +\infty$, or both. Then we tally N_j = Number of X_i's in the jth interval $[a_{j-1}, a_j)$, for $j = 1(1)k$. Note that, $\sum_{j=1}^{k} N_j = n$. Next, we compute the expected proportion p_j of the X_i's that would fall in the jth interval if we were sampling from the fitted distribution. Naturally,

$$p_j = \begin{cases} \int_{a_{j-1}}^{a_j} \hat{f}(x)dx & \text{where, } \hat{f} \text{ is the p.d.f.} \\ \sum_{a_{j-1} \leq x_i \leq a_j} \hat{p}(x_i) & \text{where, } \hat{p} \text{ is the p.m.f.} \end{cases}$$

The test statistic is given by the following equation.

$$\hat{\chi}^2 = \sum_{j=1}^{k} \frac{(N_j - np_j)^2}{np_j}$$

4.2.2 EDF Statistics

Given the random sample (the data corresponding to the input variable in the simulation model that needs to be fitted), let $X_{(1)} < X_{(2)} < \ldots < X_{(n)}$ be the ordered statistics. The empirical cumulative distribution function (ECDF) $F_n(x)$ is given by:

$$F_n(x) = \begin{cases} 0 & \text{for } x < X_{(1)} \\ \frac{i}{n} & \text{for } X_{(i)} \leq x \leq X_{(i+1)} \\ 1 & \text{for } x \geq X_{(n)}. \end{cases}$$

This is a right-continuous step function. If $F(x)$ is the CDF of the fitted distribution, then any statistic measuring the difference between $F_n(x)$ and $F(x)$ are called an EDF statistic

Kolmogorov-Smirnov Statistic (KS) Kolmogorov-Smirnov test (KS test) compares an EDF with the fitted

distribution function. Define $D^+ = \sup_x \{F_n(x) - F(x)\}$ and $D^- = \sup_x \{F(x) - F_n(x)\}$. Then, the KS statistic D is defined as:

$$D = \sup_x |F_n(x) - F(x)| = max(D^+, D^-)$$

The Quadratic Statistics The quadratic statistics are given by the following general form:

$$Q = n \int_{-\infty}^{\infty} \{F_n(x) - F(x)\}^2 \psi(x) dF(x)$$

Cramer-von Mises Statistic When $\psi(x) = 1$ in the above equation, the statistic is called Cramer-von Mises Statistic, and is usually represented by W^2.

Anderson-Darling Statistic (AD) When $\psi(x) = [\{F(x)\}\{(1 - F(x))\}]^{-1}$, the statistic is called Anderson-Darling Statistic, and is usually represented by A^2.

4.3 An Example of Distribution Fitting

Consider having the following dataset consisting of 30 numbers representing the weights of students in a class (Table 1). We will fit the dataset to normal and lognormal distributions. For deriving the MLE equations for fitting a dataset to the normal distribution, refer to (Law and Kelton 2000). For the corresponding equations for lognormal distribution, refer to (Cohen and Whitten 1988).

Table 1: Weight data for 30 students in pounds.

116.33	129.50	141.44
128.17	136.68	152.64
151.82	135.48	164.43
161.36	132.56	156.87
160.92	162.19	164.49
101.52	114.36	147.95
104.90	192.11	143.19
196.93	172.94	176.94
123.43	165.28	125.37
180.91	136.09	197.85

For the normal distribution, we estimate the mean (μ_n) as the sample mean and the standard deviation (σ_n) as the sample standard deviation. Representing the dataset as $\{x_i, i = 1 \ldots 30\}$, we get the parameters as follows.

$$\mu_n = \frac{1}{n} \sum_i x_i = \frac{4474.66}{30} = 149.16$$

$$\sigma_n = \sqrt{\frac{\sum_i (x_i - \mu_n)^2}{n-1}} = \sqrt{\frac{19565.14}{30}} = 25.97$$

Thus, the underlying distribution for the data can be a normal distribution with mean 149.16 and standard deviation 25.97.

For fitting to the two-parameter lognormal distribution with log-mean μ_l and log-standard-deviation σ_l, we use two equations as follows.

$$\mu_l = \frac{1}{n} \sum_i \ln x_i = \frac{149.70}{30} = 4.99$$

$$\sigma_l = \sqrt{\frac{1}{n} \sum_i (\ln x_i - \mu_l)^2} = \sqrt{\frac{0.9117}{30}} = 0.18$$

Thus, the underlying distribution for the data can also be a lognormal distribution with log-mean 4.99 and log-standard-deviation of 0.18. To decide the better fit among the above two distributions, we will calculate the Anderson-Darling (AD) statistic for the dataset and the two fitted distributions. This can be easily done by a spreadsheet software like Microsoft Excel. We obtain the AD statistic for the normal distribution as 0.1763, and that for the lognormal distribution as 0.1871. Since a lower GOF statistic indicates a better fit, we choose normal distribution as the better one among the above two fits.

One can also look at the *p-value* (also called *critical value*) for the above fitted distributions, using the p-value tables for the corresponding distributions. For a detailed discussion on p-values, refer to (Law and Kelton 2000). For the fitted distributions above, we obtain a p-value of 0.919 for normal distribution and 0.828 for the lognormal distribution, using the distribution fitting feature of *Oracle ® Crystal Ball* (Gentry, Blankinship, and Wainwright 2008), which is an add-in for the Microsoft® Excel software. Since a larger p-value indicates a better fit, we can conclude that normal distribution is a better fit to the data.

5   RANDOM VARIABLE GENERATION

After we have identified the underlying distributions for the input parameters of a simulation model, we generate random numbers from these distributions. The generated random numbers represent specific values of the variable. For example, we have determined that the normal distribution is the best fit for the weights of students in the previous example (section 4.3). If we want to use this information in a model which has weight as an input parameter, we would generate a random number from the distribution, and use that number as one representative weight.

In this section, we will discuss the most common method for generating random variates (RV's) from discrete and continuous distributions. We will also discuss the case of generating random numbers when an input distribution is not available. We will not discuss the generation of random numbers between 0 and 1 for a uniform distribution, we will

assume that a steady stream of uniform random numbers are available. We use these numbers for the methods discussed below. For a detailed discussion on generating uniform U(0,1) random numbers, refer to (Law and Kelton 2000, Fishman 1995).

5.1 Generating RV's from a Distribution Function

5.1.1 Inverse Transformation Method

The inverse transformation method provides the most direct route for generating a random sample from a distribution. In this method, we use the inverse of the probability density function (PDF) (for continuous distributions) or probability mass function (PMF) (for discrete distributions), and convert a random number between 0 and 1 to a random value for the input distribution. The process can be mathematically described as follows.

Let X be a continuous random variate (which we want to generate) following a PDF function f. Let the cumulative probability distribution function (CDF) for the variate be denoted by F, which is continuous and strictly increasing in (0,1). Let F^{-1} denote the inverse of the function F, which is often called inverse CDF function. Then, the following two steps will generate a random number X from the PDF f.

1. Generate $U \sim U(0,1)$.
2. Return $X = F^{-1}(U)$.

Note that, since $0 \leq U \leq 1$, $F^{-1}(U)$ always exists. The schematic diagram (Figure 3) below depicts the process. We show the curve of a CDF of a certain lognormal distribution in the right hand side. In the left hand side, we show an uniform distribution. A randomly generated number U(0.1)number (say 0.65), corresponds to 160 at the lognormal CDF curve. So, this number is a random variate from the lognormal distribution. If we generate 100 such U(0,1)numbers and replicate the process using the same curve, we will obtain 100 random variates from this distribution.

The inverse transformation method can also be used when X is discrete. For discrete distributions, if $p(x_i)$ is the probability mass function, the cumulative pmf is given by:

$$F(x) = P(X \leq x) = \sum_{x_i \leq x} p(x_i)$$

The cumulative pmf is a step function with discrete jumps. Then, the second step of the algorithm mentioned above for generating random variates from continuous distributions
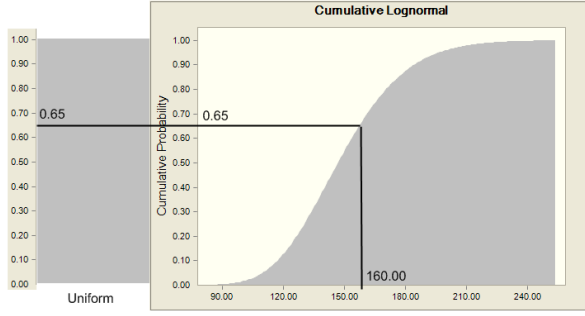
Figure 3: Generation of random variates.

can be replaced by the following: determine the smallest positive integer I such that $U \leq F(x_I)$, and return $X = x_I$.

An important advantage of the inverse transformation method is that, it can be used for generating random numbers from a truncated distribution. Also, since this method preserves the monotonicity between the uniform variate U and the random variable X, negative correlation can be successfully induced between two random variables. This method can also be used for any general type of distribution function, including functions which are a mixture of discrete and continuous distributions. One disadvantage arises from the fact that this method becomes difficult to implement if there is no closed-form inverse CDF for a distribution. If no closed form is available but $F(U)$ can be calculated easily, an iterative method (like bisection or Newton-Raphson) can be used. Note that, in addition to the numerical error inherent in working on any finite precision computer, the iterative methods induce an additional error based on the specified error tolerances. Fore more details, refer to (Devroye 1986).

There are a couple of other methods for generating random variates from distributions, for example, composition method, convolution method and acceptance-rejection method. For a more detailed treatment of these methods, and a list of formulas and methods for specific distributions, refer to (Law and Kelton 2000, Fishman 1995).

5.2 Generating RV's from a DataSet: Bootstrapped Monte Carlo

Often it is not possible to obtain an underlying distribution for an input variable in a simulation model. This can be because of the complicated shape of the original distribution (like non-convex or multi-modal), scarcity of data (for example, destructive testing or costly data) and so on. In those cases, we might end up with nothing more than a few historical values for the input parameter. In those cases, bootstrapped Monte Carlo (MC) simulation (often called bootstrapping) can be used to generate random variates. In bootstrapping, we do not really *generate* random variates. Instead, we

repeatedly sample the original dataset to choose one of the data points from the set (choose a number with replacement). For many datasets, this method provides good result for simulation purposes. For detailed reference, refer to (Efron and Tibshirani 1993, Wikipedia 2008a). For bootstrapped MC simulation, one has to still use an uniform RNG, specifically an RNG to generate integer random numbers among the indices of an array, which is being used for storing the original dataset.

Bootstrapped simulation can be a highly effective tool in the absence of a parametric distribution for a set of data. One has to be careful when performing the bootstrapped MC simulation, however. It does not provide general finite sample guarantees, and has a tendency to be overly optimistic. The apparent simplicity may conceal the fact that important assumptions are being made when undertaking the bootstrap analysis (for example, independence of samples) where these would be more formally stated in other approaches. Failure to account for the correlation in repeated observations often results in a confidence interval that is too narrow and results in a false statistical significance. Therefore, the intrinsic correlation in repeated observations must be taken into account to draw valid scientific inference.

5.2.1 Example of Bootstrapped MC

Let us assume that we are interested in using the following 20 data points in a simulation. The numbers in Table 2 are from a bimodal distribution.

Table 2: 20 samples from a bimodal distribution

7.58	-2.16
-13.08	-2.56
-13.70	-3.84
-1.27	-7.28
-17.03	-8.18
2.44	-12.30
-15.56	-18.32
-13.00	0.62
3.79	-4.45
0.28	-14.60

The following figure (Figure 4) shows a comparison between the original bimodal distribution and the bootstrapped MC simulation. In this figure, the histogram at the top shows 1000 samples randomly drawn with replacement from the original 20 numbers shown in Table 2. The samples are also called bootstrapped samples. The histogram at the bottom shows 1000 numbers randomly generated from the original bimodal distribution. Table 3 compares the basic statistics of these two sets of samples, where the first column refers to the bootstrapped MC sample, and the second column refers to the general MC sample. We notice that the parameters

are not drastically different, considering the fact that the bootstrap was done from only 20 samples. If we had more data points to perform bootstrap sampling, the result would be even better.
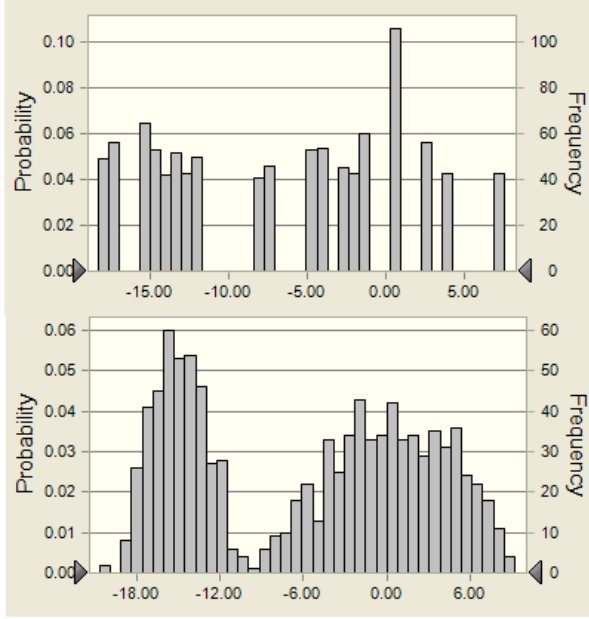

Figure 4: Bootstrap simulation and general Monte Carlo.

Table 3: Basic statistics comparison between bootstrapped MC sample and general MC sample

	BS Sample	MC Sample
Number of Trials	1000	1000
Mean	-6.79	-5.76
Median	-4.45	-4.02
Mode	-15.56	—
Standard Deviation	7.51	8.28
Variance	56.39	68.58
Skewness	0.08	-0.04
Kurtosis	1.74	1.53
Coeff. of Variability	-1.11	-1.44
Minimum	-18.32	-20.62
Maximum	7.58	9.21
Range Width	25.90	29.83
Mean Std. Error	0.24	0.26

6 MONTE CARLO SIMULATION OUTPUT ANALYSIS

The result of the Monte Carlo simulation of a model is typically subjected to statistical analysis. As mentioned before, for each set of random numbers (or trials) generated for each of the random variable, we use the model formula to arrive at a trial value for the output variable(s). When the trials are complete, the stored values are analyzed (Schuyler 1996). Averaging trial output values result in an expected value of each of the output variables. Aggregating the output values into groups by size and displaying the values as a frequency histogram provides the approximate shape of the probability density function of an output variable. The output values can themselves be used as an *empirical distribution*, thereby calculating the percentiles and other statistics. Alternatively, the output values can be fitted to a probability distribution, and the theoretical statistics of the distribution can be calculated. These statistics can then be used for developing confidence bands. The precision of the expected value of the variable and the distribution shape approximations improve as the number of simulation trials increases.

6.1 Formulas for Basic Statistical Analysis

In this section, we show the formulas for the basic statistical analysis for a set of output values. Let us assume that we have N values for each of the output parameters, each value represented as $x_i, i = 1(1)N$. Note that, these are the estimates of the complete population from the simulated sample, so we use sample statistics. For more information on unbiased estimators of population parameters from samples, refer to (Casella and Berger 2001).

Mean $(\bar{x})$

$$\bar{x} = \frac{1}{n} \sum_i x_i$$

Median 50th percentile
Standard Deviation (s)

$$s = \sqrt{\frac{1}{N-1} \sum_i (x_i - \bar{x})^2}$$

Variance (s^2)

$$s^2 = \frac{1}{N-1} \sum_i (x_i - \bar{x})^2$$

Skewness

$$\text{Skewness} = \frac{\sum_i (x_i - \bar{x})^3}{(N-1)s^3}$$

Kurtosis

$$\text{Kurtosis} = \frac{\sum_i (x_i - \bar{x})^4}{(N-1)s^4} - 3$$

Coeff. of Variability

$$\text{Coeff. of Variability} = \frac{s}{\bar{x}}$$

Minimum (x_{min})

$$x_{min} = \min_i \; x_i$$

Maximum (x_{max})

$$x_{max} = \max_i \; x_i$$

Range Width

$$\text{Range Width} = x_{max} - x_{min}$$

Mean Std. Error

$$\text{Mean Std. Error} = \frac{s}{\sqrt{n}}$$

Other than calculating the basic statistics, one can also calculate the capability statistics in case of a six-sigma-based simulation (Pyzdek 2003) (for some more details, refer to section 7.2), or perform sensitivity analysis to find out the input variables which cause the predominance of variation in the values of the output parameter of interest.

6.2 Example of MC Simulation Output

Table 3 shows an example of the calculated basic statistics which result after a Monte Carlo simulation has been performed. The table shows the output analysis from 1000 trials. For each method of simulation (bootstrapped MC and general MC), the table shows the basic statistics involved with the values of the output parameter, like mean, median, mode, variance, standard deviation, skewness, kurtosis, coefficient of variability, and so on. The table also shows the average standard error in the calculation.

7 APPLICATION AREAS FOR MONTE CARLO SIMULATION

In this section, we discuss some example problems where Monte Carlo simulation can be used. Each of these problems is representative of a broad class of similar problems, which can be solved using MC simulation. For a detailed study, refer to (Glasserman 2003).

7.1 Monte Carlo Simulation in Finance

Financial analysts use Monte Carlo simulation quite often to model various scenarios. Following are a few scenarios in which typically MC simulation gets used.

7.1.1 Real Options Analysis

In real options analysis (used in corporate finance or project finance), stochastic models use MC simulation to characterize a project's *net present value* (NPV). The traditional static and deterministic models produce single value of NPV for each project. Stochastic models can capture the input variables that are impacted by uncertainty, run MC simulation, and the average NPV of the potential investment, its volatility and other sensitivities are observed from the analysis of the output.

7.1.2 Portfolio Analysis

Monte Carlo Methods are used for portfolio evaluation (Wikipedia 2008d). Here, for each simulation, the (correlated) behavior of the factors impacting the component instruments is simulated over time, the value of the instruments is calculated, and the portfolio value is then observed. The various portfolio values are then combined in a histogram (i.e. the portfolio's probability distribution), and the statistical characteristics of the portfolio are then observed. A similar approach is used in calculating *value at risk*.

7.1.3 Option Analysis

Like real option analysis, MC simulation can be used for analyzing other types of financial instruments, like options. A MC simulation can generate various alternative price paths for the underlying share for options on equity. The payoffs in each path can be subjected to statistical analysis for making decisions. Similarly, in bond and bond options, the annualized interest rate is a uncertain variable, which can be simulated using MC analysis.

7.1.4 Personal Financial Planning

MC methods are used for personal financial planning (Wikipedia 2008d), for example, simulating the overall market to find the probability of attaining a particular target balance for the retirement savings account (known as 401(k) in United States).

7.2 Monte Carlo Simulation in Reliability Analysis and Six Sigma

In reliability engineering, we deal with the ability of a system or component to perform its required functions under stated

conditions for a specified period of time. One generally starts with evaluating the failure distribution and repair distribution of a component or a system. Then random numbers are generated for these distributions and the output result is statistically analyzed to provide with the probability of various failure events. This general method can be used for evaluating life cycle costs (for example, for *fix when broken* or *planned replacement* models), cost-effectiveness of product testing and various other reliability problems.

Six sigma is a business management strategy, which seeks to identify and remove the causes of defects and errors in manufacturing and business processes (Antony 2008). It uses various statistical methods accompanied by quality management procedures, follows a defined sequence of steps, and has quantified financial targets (cost reduction or profit increase). MC simulations can be used in six-sigma efforts for enabling six-sigma leaders to identify optimal strategy in selecting projects, providing probabilistic estimates for project cost benefits, creating virtual testing grounds in later phases for proposed process and product changes, predicting quality of business processes, identifying defect-producing process steps driving unwanted variation etc. Six-sigma principles can be applied to various industries, including manufacturing, financial and software. For more details, refer to (Pyzdek 2003).

7.3 Monte Carlo Simulation in Mathematics and Statistical Physics

Monte Carlo simulation is used to numerically solve complex multi-dimensional partial differentiation and integration problems. Is is also used to solve optimization problems in Operations Research (these optimization methods are called simulation optimization). In the context of solving integration problems, MC method is used for simulating quantum systems, which allows a direct representation of many-body effects in the quantum domain, at the cost of statistical uncertainty that can be reduced with more simulation time. One of the most famous early uses of MC simulation was by Enrico Fermi in 1930, when he used a random method to calculate the properties of the newly-discovered neutron (Wikipedia 2008c).

7.4 Monte Carlo Simulation in Engineering

Monte Carlo simulation is used in various engineering disciplines for multitude of reasons. One of the most common use is to estimate reliability of mechanical components in mechanical engineering. Effective life of pressure vessels in reactors are often analyzed using MC simulatio, which falls under chemical engineering. In electronics engineering and circuit design, circuits in computer chips are simulated using MC methods for estimating the probability of fetching instructions in memory buffers. In computer science and software engineering, various algorithms use MC methods, for example, to detect the reachable states of a software model and so on.

8   MONTE CARLO SIMULATION SOFTWARE

Various options are available to use Monte Carlo simulations in computers. One can use any high-level programming language like C, C++, Java, or one of the .NET programming languages introduced by Microsoft®, to develop a computer program for generating uniform random numbers, generating random numbers for specific distributions and output analysis. This program will possibly be tailor-made for specific situations. Various software libraries are also available in most of these high level programming languages, to facilitate the development of MC simulation code. Then, there are stand-alone software packages which can be used for MC simulations. These are general purpose simulation software packages, which can be used to model an industry-specific problem, generate random numbers, and perform output analysis. **Examples needed**. Finally, MC simulations can also be performed using add-ins to popular spreadsheet software like Microsoft® Excel. Using these software, one typically starts by developing a deterministic model for the problem, and then defines distributions for the input variables which contain uncertainty. Finally, these add-ins are capable of generating charts and graphs of the output parameters for further analysis. Crystal Ball from Oracle® (Gentry, Blankinship, and Wainwright 2008), @RISK from Palisade, and the Solver add-in from Frontline Systems are a few examples of this type of software.

9   CONCLUSION

Monte Carlo simulation is a very useful mathematical technique for analyzing uncertain scenarios and providing probabilistic analysis of different situations. The basic principle for applying MC analysis is simple and easy to grasp. Various software have accelerated the adoption of MC simulation in different domains including mathematics, engineering, finance etc. In this tutorial article, we have discussed the methodology, theoretical basis, and application domains for Monte Carlo simulation. Readers interested in further exploring this field are adviced to go through the list of references, or contact the author.

ACKNOWLEDGMENTS

REFERENCES

Antony, J. 2008. Pros and cons of six sigma: an academic perspective. Available via <http://www.onesixsigma.com/node/7630>.

Casella, G., and R. L. Berger. 2001. *Statistical inference*. 2nd ed. Duxbury Press.

Cohen, A. C., and B. J. Whitten. 1988. *Parameter estimation in reliability and life span models*. N.Y., USA: Marcel Dekker, Inc.

D'agostino, R. B., and M. A. Stephens. 1986. *Goodness-of-fit techniques*. N.Y., USA: Marcel Dekker, Inc.

Devroye, L. 1986. *Non-uniform random variate generation*. N.Y., USA: Springer-Verlag.

Efron, B., and R. J. Tibshirani. 1993. *An introduction to the bootstrap*. N.Y., USA: Chapman and Hall.

Fishman, G. S. 1995. *Monte carlo: Concepts, algorithms, and applications*. N.Y., USA: Springer-Verlag.

Gentry, B., D. Blankinship, and E. Wainwright. 2008. *Oracle crystal ball user manual*. 11.1.1 ed. Denver, USA: Orcale, Inc.

Glasserman, P. 2003. *Monte carlo methods in financial engineering*. N.Y., USA: Springer.

Law, A. M., and W. D. Kelton. 2000. *Simulation modeling & analysis*. 3rd ed. N.Y., USA: McGraw-Hill, Inc.

Pyzdek, T. 2003. *The six sigma handbook: The complete guide for greenbelts, blackbelts, and managers at all levels*. 2nd ed. N.Y., USA: McGraw-Hill.

Schuyler, J. R. 1996. *Decision analysis in projects*. P.A., USA: Project Management Institute.

Wikipedia 2008a. Bootstrapping (statistics) — wikipedia, the free encyclopedia. Available via <http://en.wikipedia.org/w/index.php?title=Bootstrapping_(statistics)&oldid=239201200>. [accessed September 19, 2008].

Wikipedia 2008b. Maximum likelihood — wikipedia, the free encyclopedia. Available via <http://en.wikipedia.org/w/index.php?title=Maximum_likelihood&oldid=237429266>. [accessed September 19, 2008].

Wikipedia 2008c. Monte carlo method — wikipedia, the free encyclopedia. Available via <http://en.wikipedia.org/w/index.php?title=Monte_Carlo_method&oldid=237702035>. accessed September 19, 2008].

Wikipedia 2008d. Monte carlo methods in finance — wikipedia, the free encyclopedia. Available via <http://en.wikipedia.org/w/index.php?title=Monte_Carlo_methods_in_finance&oldid=236249285>. [accessed September 19, 2008].

AUTHOR BIOGRAPHY

SAMIK RAYCHAUDHURI, Ph.D. is a senior member of the technical staff of the Oracle Crystal Ball Global Business Unit and an active member of INFORMS (Institute of Operations Research and Management Science). He has a bachelors degree in Industrial Engineering from Indian Institute of Technology, Kharagpur, India, and a masters and Ph.D. degree in Industrial Engineering from University of Wisconsin at Madison, USA. His research interests include Monte Carlo simulation, gaming mode simulation models (simulation games) and development of nonlinear optimization algorithms. His email address is <samikr@gmail.com>.