4

Monte Carlo Simulation

Sankaran Mahadevan

Vanderbilt University
Nashville, Tennessee

1 INTRODUCTION

Monte Carlo simulation is a widely used technique in the probabilistic analysis of engineering systems. It is a numerical experimentation technique to obtain the statistics of the output variables of a system computational model, given the statistics of the input variables. In each experiment, the values of the input random variables are sampled based on their distributions, and the output variables are calculated using the computational model. A number of experiments are carried out in this manner, and the results are used to compute the statistics of the output variables.

In the context of physics-based reliability analysis, the Monte Carlo simulation method has been used for two purposes: (i) validation of the analytical methods described in Chapter 6, and (ii) solution of large, complex systems when analytical approximations are not easy to make. The second case occurs when the performance function* $g(X)$ cannot be written

* The performance function used here and in Chapter 6 is called the limit-state function in Chapters 2 and 4. Reference is also made to first-order reliability methods (FORM) and second-order reliability methods (SORM). The FORM algorithm is introduced in Chapter 2, while FORM and SORM are discussed in fundamental detail in Chapter 6.

as a closed-form expression in terms of the input random variables X. Instead, $g(X)$ can only be evaluated numerically through an analysis procedure such as the finite element method for particular sets of values of the input variables. Monte Carlo simulation performs multiple deterministic analyses of the system for different sets of values of the input variables, and calculates the required statistics.

In order to evaluate the failure probability corresponding to a particular condition $g(X) < 0$, the direct Monte Carlo simulation method would consist of the following steps.

1. Generate the values of the input random variables according to their probability distribution functions.
2. Perform the deterministic analysis and check if the system has failed (i.e., $g(X) < 0$).
3. Repeat steps 1 and 2 a number of times (N) and count the number of failures (N_f).
4. Obtain the estimate of the mean probability of failure as

$$\bar{p}_f = \frac{N_f}{N} \tag{1}$$

The expected value of this sample mean gives the population mean of the failure probability. The accuracy of the estimate is dependent upon the number of simulation runs N. As N approaches infinity, the estimated probability of failure should approach the true value. Hence a large number of such simulations is required for accuracy in the estimate.

In order to successfully conduct this type of simulation, the first task is to learn how to choose the values of the input variables for each experiment. Most of the commercially available statistical analysis computer programs have built-in random-number generators for many of the commonly used probability distributions. The basic concepts of these generators are presented in the next section.

2 GENERATION OF RANDOM NUMBERS

If a statistical estimate is to be based upon repeated experiments, then the values of an input variable should be chosen such that the group of values for all the runs is representative of the probability characteristics of this input variable. In other words, the values should be sampled based on the probability distribution of the random variable. Also, the values (numbers) should appear to be random.

The basic building block of such sampling is the ability to generate random numbers from a uniform distribution between 0 and 1. That is, any

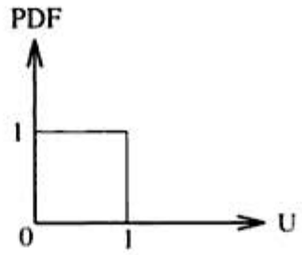

Figure 1 Uniform distribution $U(0, 1)$.

value of the variable between 0 and 1 (both values included) is as likely as any other value in this range. Such a variable may be denoted as $U(0, 1)$, where the symbol U indicates the uniform distribution, and the numbers within the parenthesis indicate the range. Such a variable may also be shown pictorially as in Figure 1.

The schemes to sample random numbers (values) from $U(0, 1)$ are actually deterministic, but the numbers thus generated have the appearance of arbitrary values of independent $U(0, 1)$ random variables, and are sufficient for the purpose of simulation. Therefore such random numbers are called pseudo-random numbers.

Now, every computer comes with a random-number generator. A common method of generating random numbers starts with a seed value (x_0), and then computes successive values of x by using a recursive formula. A commonly used recursive formula is

$$x_{i+1} = (ax_i + c) \text{ modulo } m \tag{2}$$

where a, c, and m are non-negative integers. The above equation means that $ax_i + c$ is divided by m and the remainder is taken as x_{i+1}. The random number between 0 and 1 is then obtained as

$$u_{i+1} + \frac{x_{i+1}}{m} \tag{3}$$

The values of x in Eq. (2) can go from 0 to a maximum of $m - 1$. To avoid repetition of the same numbers in short cycles, m is chosen to be the largest prime number possible in the computer. A satisfactory value of m for a 32-bit computer is $2^{31} - 1$. Similarly, the value of a is $2^7 + 1$, and $c = 1$. Another recursive formula is more commonly used by taking $c = 0$ in Eq. (2).

The reader is encouraged to consider the discussion and example source codes that are contained in Ref. 1. The random-number generator algorithm discussed above is suitable for relatively small sample sizes suitable for generating the mid-range of the cumulative distribution function. How-

ever, for large simulations that are necessary for the tails of the distributions, or on computers with limited floating point word size, the cited reference should be consulted. The need for a suitable random-number generator for large problems was discussed briefly in Chapter 2. An example Monte Carlo code based on the inverse transform method is listed in Chapter 2.

3 GENERATION OF RANDOM VARIABLE VALUES

3.1 The Inverse Transform Method

Once a $U(0, 1)$ random number has been generated, this can be used to generate a value of the desired random variable with the given distribution. The most common method is the inverse transform method. If the CDF of the random variable X is $F_X(x)$, and u is a random number generated from a $U(0, 1)$ distribution, then the corresponding value of X is simply

$$x = F_X^{-1}(u) \tag{4}$$

The above relationship can be easily explained with the help of Figure 2. The figure shows the CDF of X on the right-hand side and the CDF of U on the left-hand side. Since u is a value of U, a uniformly distributed random variable between 0 and 1, the CDF value at u is

$$F_U(u) = u \tag{5}$$

from the left-hand side of Figure 2. That is, $P(U \leq u) = u$. We use the same probability value on the right-hand side, and find the value of X that corresponds to this probability. Thus we obtain a value of X that has the same cumulative probability as u.

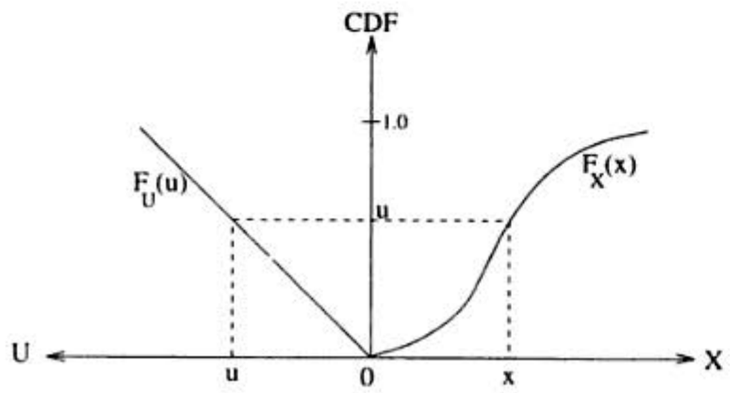

Figure 2 The inverse transform method.

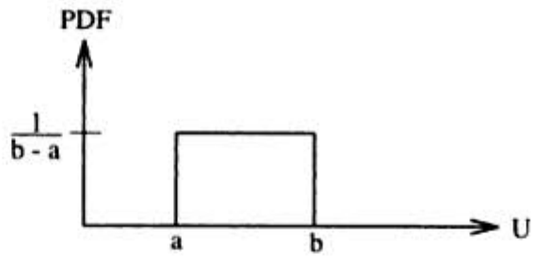

Figure 3 Uniformly distributed random variable X $U(a, b)$.

Example 1

If X is a uniform random variable and its range is (a, b), then its PDF and CDF are (see Figure 3)

$$f_X(x) = \frac{1}{b - a} \tag{6}$$

$$F_X(x) = \frac{x - a}{b - a} \tag{7}$$

Therefore, using the inverse transform method, a value of X corresponding to a random number u generated from $U(0, 1)$ may be obtained as

$$\frac{x - a}{b - a} = u \tag{8}$$

or

$$x = a + u(b - a)$$

Example 2

Consider generating a value for an exponential random variable X with the CDF

$$F_X(x) = 1 - e^{-\lambda x} \tag{9}$$

Using the inverse transform method, a value of X can be generated as

$$x = F_X^{-1}(u)$$
$$= -\frac{1}{\lambda} \ln(1 - u) \tag{10}$$

Notice that since u is a random value of $U(0, 1)$, $1 - u$ is also a random value of $U(0, 1)$. Therefore the above formula can be simplified as

$$x = -\frac{1}{\lambda} \ln u \tag{11}$$

The inverse transform method is easy to apply if the inverse of the CDF of X can be expressed analytically. This is not possible, for example, in the case of normal and log-normal variables. Consider a normal random variable. The PDF is

$$f_X(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{12}$$

The CDF is the integral of the above function and can only be computed numerically. Therefore we cannot write an analytical expression for the inverse of the CDF. In such cases, several other methods such as the acceptance–rejection method, or the composition method, etc., may be useful. A numerical inversion formula can also be developed by curve-fitting of the numerical normal distribution. The simple Monte Carlo code in the Appendix to Chapter 2 uses such a curve-fit for the inversion problem.

3.2 The Acceptance–Rejection Method

Let $f(x)$ be the PDF of a desired variable X. In the acceptance–rejection method, we first generate a random number x from another PDF $g(x)$. Then we generate a $U(0, 1)$ random number u, and check the following criterion

$$u \leq \frac{f(x)}{cg(x)} \tag{13}$$

where c is pre-calculated constant as the maximum value of the ratio $f(x)/g(x)$. If the criterion is satisfied, then x is accepted as a value of the random variable X with the probability $f(x)/cg(x)$. If the criterion is not satisfied, then x is rejected and the above process is repeated. See Ross [2] for a proof of the validity of this method.

Example 3

The generation of normal and log-normal variable values using the acceptance–rejection method will be illustrated below.

Consider generating a value of a standard normal random variable S, i.e., $N(0, 1)$. Its PDF is

$$f(s) = \frac{1}{\sqrt{2\pi}} e^{-s^2/2} \qquad -\infty < s < \infty \tag{14}$$

We will first try to generate Z, the absolute value of s. The PDF of Z is written as

$$f(z) = \frac{2}{\sqrt{2\pi}} e^{-z^2/2} \qquad 0 < z < \infty \tag{15}$$

Let us choose the other density function, $g(z)$, to be an exponential density function

$$g(z) = e^{-z} \qquad 0 < z < \infty \tag{16}$$

Then

$$\frac{f(z)}{g(z)} = \frac{1}{\sqrt{2\pi}} e^{z - z^2/2} \tag{17}$$

The maximum value of this ratio occurs when $z - z^2/2$ is maximum. This happens when $z = 1$. Therefore

$$c = \max \frac{f(z)}{g(z)} = \frac{f(1)}{g(1)} = \sqrt{\frac{e}{2\pi}} \tag{18}$$

The value of a standard normal variable S is generated in the following steps.

1. Generate z from the exponential density function $g(z)$ of Eq. (16).
2. Generate a random number u_1 from $U(0, 1)$.
3. If $u_1 \leq f(z)/cg(z)$, accept z. (Note: $f(z)$ is calculated using Eq. (15).) Otherwise return to Step 1.

Now that a value of the variable Z has been generated, we can generate a value of S by letting S be equally likely to be either z or $-z$. To do this, simply generate another uniform random number u_2 within the range $(-0.5, 0.5)$. If u_2 is negative, set $s = -z$; if u_2 is positive, set $s = z$.

Having generated a value of the standard normal variable S, how do we generate a corresponding value of any other normal variable X? To do this, observe that

$$S = \frac{X - \mu}{\sigma} \tag{19}$$

where μ is the mean value and σ is the standard deviation of X. Therefore, the corresponding value of X is obtained as

$$x = \mu + \sigma s \tag{20}$$

Note also that s can be used to generate a log-normal variable X as well, because

$$S = \frac{\ln X - \lambda}{\xi} \tag{21}$$

where λ is the mean value of $\ln(X)$ and ξ is the standard deviation of $\ln(X)$. Therefore, the value of X corresponding to s is obtained as

$$x = \exp(\lambda + \xi s) \tag{22}$$

- Note 1: The method of generating a value of X from a value of S is referred to as the function of random variables method, since X can be expressed as a function of S.
- Note 2: A polar method by Box and Muller [3] is also available to generate normal/log-normal variables. See, for example, Ross [2] for an explanation of this method. This method is also an example of the function of random variables method.

3.3 The Composition Method

In this method, the PDF of the required random variable is expressed as a weighted sum of several other density functions (with which it is easy to generate random values). This is written as

$$f_X(x) = \sum_i f_{X_i}(x) p_i \tag{23}$$

where $f_{X_i}(x)$, $i = 1, 2, \ldots, n$ is a set of component density functions, and p_i is the probability or relative weight associated with $f_{X_i}(x)$ [4]. The steps for the generation of x are

1. Generate a $U(0, 1)$ random number for the probability p_i and select the corresponding density function $f_{X_i}(x)$.
2. Generate another random number according the PDF $f_{X_i}(x)$. This is the required value of X.

Example 4 (from Ang and Tang [4])

Consider a random variable X with a PDF

$$f_X(x) = \tfrac{2}{3} + x^2 \qquad 0 \le x \le 1 \tag{24}$$

as shown in Figure 4. The PDF may be decomposed as

$$f_X(x) = \tfrac{2}{3} f_1(x) + \tfrac{1}{3} f_2(x) \qquad 0 \le x \le 1 \tag{25}$$

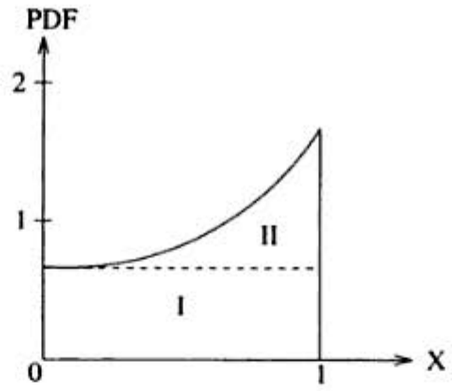

Figure 4 The composition method.

where

$$f_1(x) = 1$$
$$f_2(x) = 3x^2 \tag{26}$$

The relative weights p_1 and p_2 are $\frac{2}{3}$ and $\frac{1}{3}$ respectively, corresponding to the areas of the two density functions in Figure 4.

Two $U(0, 1)$ random numbers u_1 and u_2 are generated. If $u_1 < \frac{2}{3}$, then u_2 is used to determine a value x from $f_1(x)$, by the inverse transform method. If $u_1 > \frac{2}{3}$, then u_2 is used to determine a value x from $f_2(x)$.

In this example problem, the corresponding CDFs are

$$F_1(x) = x$$
$$F_2(x) = x^3 \tag{27}$$

Therefore, the inverse transform formulas for the two components are

$$x = F_1^{-1}(u_2) = u_2$$
$$x = F_2^{-1}(u_2) = (u_2)^{1/3} \tag{28}$$

Thus, the random-number generation scheme is

$$x = u_2 \qquad \text{if } u_1 < \tfrac{2}{3}$$
$$x = (u_2)^{1/3} \qquad \text{if } u_1 > \tfrac{2}{3} \tag{29}$$

4 CORRELATED RANDOM VARIABLES

A typical mechanical reliability problem may have a number of random variables, as discussed in Chapter 2. The methods described in the previous section are for generating individual random variables. If some of the random

variables are correlated, then it is not enough to generate the random variable values independently. In general, it is not easy to generate correlated variables, except when the variables are normal.

The problem of correlated random variables in mechanical system reliability is non-trivial and can only be treated generally by some approximations. For a more complete discussion of the subject in the context of reliability modeling using matrix structural analysis methods, the reader is directed to Appendix B in Melchers, book [5]. Linear systems can be described in terms of the expected or mean value for each of the variables, and the covariance of the system. Covariance is the second moment of the joint probability distribution functions of variables. Typically, in mechanical vibration or structures problems, the covariance is given as a matrix of the following form

$$L_{ij} = \mathrm{COV}(X_i, X_j) = \sigma_{ij} \tag{30}$$

In the case of independent random variables, COV_{ij} is a diagonal matrix of the individual variable variances.

As described by Melchers, there are two algorithms for generating independent random variables. The first is the Rosenblatt transformation [6] which applies to non-normally distributed variables. The second applies to linear systems of normally distributed variables. As discussed in Chapter 6, we can approximate mechanical system reliability problems using a linearization of the performance function and apply the Rosenblatt transformation to obtain equivalent normal distributions for non-normally distributed random variables. The general, albeit approximate mechanical reliability problem is modeled as a linear system with a correlation matrix which can be transformed to independent random variables as follows.

The method of converting the covariance matrix into a diagonal form is the same as that of generating eigenvalues of the covariance matrix. The transformation produces a diagonal matrix of variances in a new variable space where the new variables are statistically independent of each other. The Monte Carlo simulation algorithm generates elements of the distributions of the individual independent random variables. The inverse of the transformation should be applied now to obtain the values of correlated variables from individually generated values. A full discussion of the decomposition process is given in Chapter 6.

Therefore, using the Cholesky decomposition method, we know that the correlated variables X can be transformed to uncorrelated variables Y as

$$Y = L^{-1}X \tag{31}$$

where L^{-1} is the inverse of the lower triangular matrix obtained by Cholesky

factorization of the *covariance matrix* (not the correlation matrix!) of X. However, note that when Eq. (31) is applied, the resulting Y variables are linear combinations of the original X variables; therefore, the density functions are different in the Y space. The density functions of the Y variables are used to generate the Y values. The Y values are then transformed to X values using

$$X = LY \tag{32}$$

which is simply the inverse relationship of Eq. (31).

A possible way to generate correlated non-normal variables is to first generate values in the uncorrelated standard normal space, and then use the equivalent normal transformation to find the values in the original space. However, this transformation varies from point to point. The transformation at the most probable point (MPP) (estimated by FORM as discussed in Chapter 6) may serve as an approximate relationship, for all the samples.

5 A SIMULATION: EXAMPLE 5

Consider the performance or limit-state function $g(X) = f_y A - P$, where f_y is the yield strength, A is the area of cross-section, and P is the applied bending moment at a beam cross-section. This is the same numerical example as in Chapter 6, which is solved using various analytical methods. The same problem will be solved with different simulation techniques in this chapter. The data is reproduced in Table 1.

The variables are assumed to be uncorrelated, since it is difficult to generate correlated non-normal variables. The probability of failure is expressed as

$$p_f = P(g < 0) = P(\sigma_Y Z - M < 0) \tag{33}$$

Table 1 Statistical Description of the Random Variables

Random variable definitions			
Variable	Mean	COV	Distribution
f_y	40 ksi	0.125	Log-normal
A	1 in^2	0.05	Log-normal
P	20 kips	0.20	Type I asymptotic extreme

For the sake of convenience, let the variables be re-named as before

$$X_1 = f_Y$$
$$X_2 = A \tag{34}$$
$$X_3 = P$$

The steps of the direct Monte Carlo simulation method are:

1. Generate values of f_y, A, and P according to their respective distributions.
2. Compute the value of g.
3. Repeat Steps 1 and 2 N times, and count the number of failures N_f.
4. The probability of failure is estimated as $p_f = N_f/N$.

The first step is easier if the variables are uncorrelated. In that case, each variable is separately generated according to its own distribution. Values of the log-normal variables X_1 and X_2 are generated using Eq. (22). For the extreme value variable X_3, the CDF is given as

$$F_{X_3}(x_3) = \exp[-e^{-\alpha(x_3-u)}] \tag{35}$$

where α and u are the parameters of the distribution. These can be expressed in terms of the mean μ_{X_3} and standard deviation σ_{X_3} as

$$\alpha = \frac{\pi}{\sqrt{6}}\frac{1}{\sigma_{X_3}} \tag{36}$$

$$u = \mu_{X_3} - \frac{0.577}{\alpha} \tag{37}$$

A $U(0, 1)$ random number is first generated (denoted as u'). Then the corresponding value of X_3 is computed using the inverse transform method as

$$\begin{aligned} u' &= \exp[-e^{-\alpha(x_3-u)}] \\ -\ln(u') &= e^{-\alpha(x_3-u)} \\ -\ln(-\ln(u')) &= \alpha(x_3 - u) \end{aligned} \tag{38}$$

Therefore

$$x_3 = u - \frac{\ln(-\ln(u'))}{\alpha}$$

The number of samples was chosen to be 100,000. The probability of failure was estimated to be 0.0031.

6 ACCURACY OF MONTE CARLO SIMULATION

In the Monte Carlo method, a deterministic simulation of the system is performed for each set of random variable values generated according to the methods of the above sections. An important question at this stage is: how many such simulations are required? The answer obviously depends on the required accuracy. Shooman [7] derived the following formula relating the number of simulations n and the percentage error:

$$\% \text{ error} = 200\sqrt{\frac{1 - p_f}{np_f}} \tag{39}$$

where p_f is the estimated failure probability. There is a 95% chance that the error in the estimated probability will be less than that given in the above equation. For example, if 10,000 simulations were performed (i.e., $n = 10{,}000$), and the estimated probability $p_f = 0.01$, then Eq. (39) will yield a 20% error. That is, it is 95% likely that the actual probability will be within 0.01 ± 0.002 [4].

Failure probabilities of interest in mechanical reliability problems are usually in the range 10^{-3} to 10^{-6}. Assume that the failure probability corresponding to a limit state is on the order of 10^{-3}, and that this is to be estimated within an accuracy of 10%. What is the number of simulations required to achieve this accuracy? Using Eq. (39), the number of simulations required is $n = 399{,}600$ with a 95% likelihood that the error will be within 10%. It is easy to appreciate the large number of simulations required in the Monte Carlo method, since 10^{-3} is a relatively high failure probability for mechanical systems. The method becomes prohibitively expensive and time-consuming for large systems where each simulation may require the execution of a complicated matrix or other numerical analysis code. Several techniques have been proposed to alleviate this problem. These are discussed in the following section.

7 IMPROVING THE EFFICIENCY OF SIMULATION

As seen in the previous section, direct Monte Carlo simulation is economically prohibitive, especially for complicated structures in which the deterministic analysis itself is expensive. In order for Monte Carlo simulation to be practical, its efficiency has to be improved, i.e., accurate estimates have to be obtained using a small number of simulations. The efficiency of the simulation is characterized by the mean value and the variance of the estimated output variable. Hence the efficiency of the simulation can be increased by reducing the variance of the estimated output variable using