(*a*) Formulate a performance function Y that can be used to describe the failure condition of the beam. Consider bending only. [Hint: Consider Y = (capacity) − (demand). Failure corresponds to the case when Y < 0.]

(*b*) Formulate an alternative performance function Y involving the product and quotient of random variables. Consider bending only. [Hint: See Example 3.3.]

(*c*) If P and F_y are both normal random variables, calculate the probability of failure using the results of (*a*).

(*d*) If P and F_y are both lognormal random variables, calculate the probability of failure using the results of (*b*).

3.6. Consider a three-span continuous beam. All supports are pinned supports. Each span has a length of L. The beam has a modulus of elasticity E and a moment of inertia I. All spans are subjected to a uniformly distributed load w. The maximum deflection of the beam occurs in the outer spans and is equal to

$$0.0069\frac{wL^4}{EI}$$

Your job is to evaluate the probability that the deflection will exceed the code-specified limit of L/360 given the following information:

L = 5 m (deterministic).

w is lognormal with a mean value of 10 kN/m and a coefficient of variation of 0.4.

E is lognormal with a mean value of 2×10^7 kN/m^2 and a coefficient of variation of 0.25.

I is lognormal with a mean value of 8×10^{-4} m^4 and a standard deviation of 1.5×10^{-4} m^4.

Calculate the probability of failing the deflection criterion.

3.7. Consider the performance function

$$Y = 3X_1 - 2X_2$$

where X_1 and X_2 are both normally distributed random variables with

$$\mu_{X_1} = 16.6 \qquad \mu_{X_2} = 18.8$$
$$\sigma_{X_1} = 2.45 \qquad \sigma_{X_2} = 2.83$$

The two variables are correlated, and the covariance is equal to 2.0. Determine the probability of failure if failure is defined as the state when Y < 0.

3.8. The resistance (or capacity) R of a member is to be modeled using

$$R = R_n MPF$$

where R_n is the nominal value of the capacity determined using code procedures and M, P, and F are random variables that account for various uncertainties in the capacity. If M, P, and F are all lognormal random variables, determine the mean and variance of R in terms of the means and variances of M, P, and F.

4

SIMULATION TECHNIQUES

SEVERAL TECHNIQUES MAY be used to solve structural reliability problems. In this chapter, simulation techniques are presented as one possible way to solve such problems. The basic idea behind simulation is, as the name implies, to numerically simulate some phenomenon and then observe the number of times some event of interest occurs. The basic concept behind simulation is relatively straightforward, but the procedure can become computationally intensive.

4.1 MONTE CARLO METHODS

4.1.1 Basic Concept

Assume that we have information from N tests, and assume that we put all N test results in a bag as shown in Figure 4.1. Now suppose that we need a sample of n test results. Instead of doing n additional tests, we could randomly select n of the N test results from the bag. In Figure 4.1, this sampling technique is referred to as the "special technique."

The Monte Carlo method is a special technique that we can use to generate some results numerically without actually doing any physical testing. We can use results of previous tests (or other information) to establish the probability distributions of the important parameters in our problem. Then we use this distribution information to generate samples of numerical data. To illustrate the basic idea, consider the following example.

EXAMPLE 4.1. Consider a series of actual tests of concrete cylinders to determine the compressive strength f'_c. Assume that a relative frequency histogram has been drawn using the data, and a lognormal probability distribution as shown in Figure 4.2 appears to fit the data reasonably well.

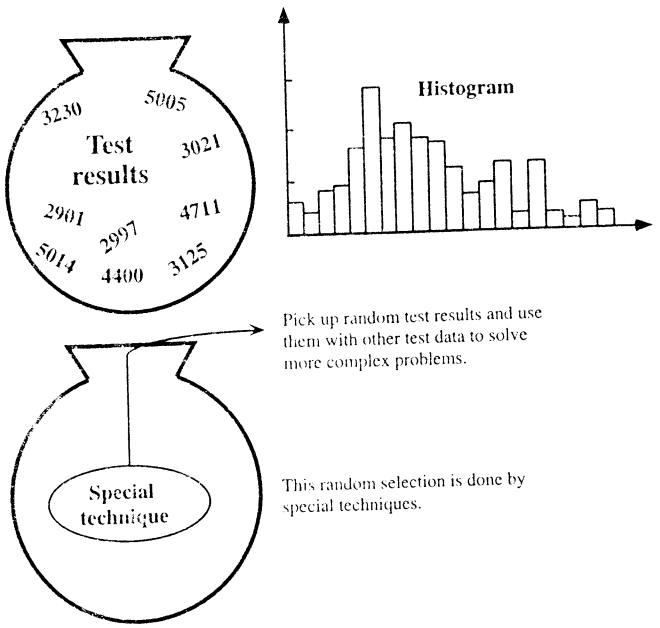

FIGURE 4.1 Schematic of the Monte Carlo method.

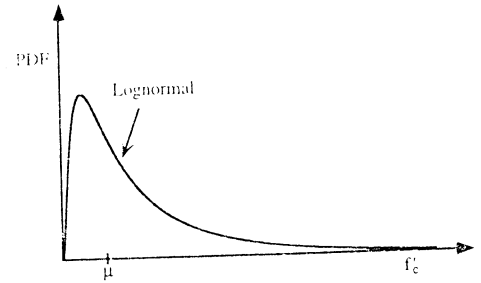

FIGURE 4.2 Lognormal PDF for Example 4.1.

Now consider a concrete column. The compressive load-carrying capacity is $0.85 f'_c A_c$ where A_c is the cross-section area of the column and is assumed to be deterministic. Assume that the applied load, Q, is normally distributed with a mean value μ_Q and a coefficient of variation V_Q. What is probability of failure, p_F?

Solution. The performance function for this case could be written as

$$Y = R - Q$$

where $R = 0.85 f'_c A_c$. The probability of failure is the probability that $R < Q$ or

$$p_F = P(Y < 0) = P(R - Q < 0)$$

Unfortunately, in this case, the distribution of R is not normal (it is lognormal) even though Q is normal. Therefore, although we can calculate a mean and variance using the methods presented in Chapter 3, we do not have a closed-form solution for the probability distribution of Y. However, the problem can be solved using Monte Carlo simulation. The basic procedure is as follows:

1. Randomly generate a value of f'_c (using the probability distribution information given above) and calculate $R = 0.85 f'_c A_c$.
2. Randomly generate a value of Q using its probability distribution.
3. Calculate $Y = R - Q$.
4. Store the calculated value of Y.
5. Repeat Steps 1–4 until a sufficient number of Y values have been generated.
6. Plot the simulated data on normal probability paper (see Chapter 2) and use the graph to estimate the probability of failure. Alternatively, if a sufficient number of simulated values are available, estimate the probability of failure as

$$\tilde{P} = \frac{\text{number of times that } Y < 0}{\text{total number of simulated Y values}}$$

We have skipped some details, but the basic procedure is just like this. The details are discussed in subsequent sections.

The Monte Carlo method is often applied in three situations:

1. It is used to solve complex problems for which closed-form solutions are either not possible or extremely difficult. For example, probabilistic problems involving complicated nonlinear finite element models can be solved by Monte Carlo simulation provided that the necessary computing power is available and the required input information is known.
2. It is used to solve complex problems that can be solved (at least approximately) in closed form if many simplifying assumptions are made. By using Monte Carlo simulation, the "original" problem can be studied without these assumptions, and more realistic results can be obtained.
3. It is used to check the results of other solution techniques.

This chapter provides only a brief introduction to the topic of simulation. For additional details and applications, see Marek, Gustar, and Anagnos (1996), Rubinstein (1981), Ross (1997), Ayyub and McCuen (1997), and Ang and Tang (1984).

4.1.2 Generation of Uniformly Distributed Random Numbers

The basis of all Monte Carlo simulation procedures is the generation of random numbers that are uniformly distributed between 0 and 1. Tables of randomly generated numbers are available (e.g., Rand Corporation, 1955), as are computer subroutines, and many popular mathematical programs have such subroutines built in. Table 4.1 is an example of a table of uniform random variables that was generated using a standard spreadsheet program.

Once we have some realizations u of a uniformly distributed random number U between 0 and 1, we can generate realizations x of a uniformly distributed random number X between any two values a and b ($a \leq x \leq b$) using the following formula:

$$x = a + (b - a)u \tag{4.1}$$

We can also generate sample values i for a uniformly distributed random *integer* I between two integer values a and b (including the values a and b) using the following formula:

$$i = a + \text{TRUNC}[(b - a + 1)u] \tag{4.2}$$

where TRUNC[ ] denotes a function which truncates its argument (i.e., removes the fractional part and returns the integer part of a real number).

TABLE 4.1 Simulated values of a uniformly distributed random variable (values between 0 and 1)

0.050203	0.269082	0.442000	0.390912	0.084078	0.597430	0.249519	0.892361
0.619129	0.472640	0.833705	0.876064	0.821741	0.149907	0.653035	0.908841
0.872402	0.422864	0.412275	0.462844	0.444990	0.774895	0.345225	0.834681
0.376568	0.467299	0.145451	0.926969	0.337626	0.648000	0.323649	0.656117
0.139927	0.415784	0.849178	0.307840	0.059633	0.498886	0.433912	0.320231
0.318491	0.523667	0.598193	0.005036	0.132786	0.892575	0.835353	0.666829
0.987671	0.243629	0.561388	0.414869	0.378796	0.301706	0.328349	0.937925
0.033265	0.741569	0.169408	0.118229	0.195227	0.515915	0.575213	0.709037
0.234626	0.408673	0.967040	0.398450	0.730552	0.411115	0.703421	0.591021
0.623157	0.021790	0.864834	0.319895	0.726890	0.023835	0.073214	0.012818
0.957884	0.547472	0.332286	0.970153	0.762535	0.055788	0.118198	0.082675
0.518906	0.749779	0.849239	0.173711	0.586932	0.554094	0.767510	0.751549
0.442305	0.681600	0.834803	0.406201	0.803613	0.162908	0.224067	0.956908
0.445845	0.140538	0.885769	0.760491	0.132450	0.866573	0.869015	0.663656
0.834284	0.888607	0.359783	0.297708	0.564196	0.141392	0.706259	0.199316
0.811213	0.105136	0.607227	0.119755	0.449202	0.731407	0.930631	0.250893
0.935728	0.067202	0.705435	0.150792	0.448347	0.502579	0.392346	0.740989
0.450423	0.864772	0.841975	0.671010	0.403912	0.711509	0.072970	0.974548
0.579058	0.363628	0.781091	0.455458	0.431471	0.579272	0.923032	0.748009
0.662648	0.863948	0.491226	0.011902	0.905759	0.806421	0.929014	0.749077
0.039918	0.858242	0.557054	0.278726	0.593005	0.191778	0.047029	0.050478
0.414075	0.432234	0.934263	0.173040	0.358684	0.660817	0.372265	0.925138
0.103214	0.412091	0.087985	0.395398	0.606067	0.401349	0.596393	0.385754
0.112308	0.749199	0.242988	0.543260	0.649922	0.168065	0.111393	0.986175
0.821833	0.260933	0.012574	0.562700	0.402142	0.109256	0.144383	0.030641

Before leaving the topic of generating uniform random variables, two comments are in order. First, most (if not all) random number generators require the user to input a "seed" value. This number is an integer that is used by the routine to start the simulation algorithm. By choosing a different seed, you can generate a different set of uniformly distributed numbers. In general, if you use the same seed over and over again, you will generate the same set of uniformly distributed random numbers over and over again. Second, the built-in generators found in many software packages should be used with caution. Some random number generation algorithms work better than others. A discussion of this issue is beyond the scope of this text. For a concise summary of possible problems and a listing of various random number generation algorithms, see Press, Teukolsky, Vetterling, and Flannery (1992, Chapter 7).

4.1.3 Generation of Standard Normal Random Numbers

Since the normal probability distribution plays such an important role in structural reliability analysis, the capability to simulate normally distributed random variables is important. To begin, consider a standard normal distribution. To generate a set of standard normal random numbers $z_1, z_2, \ldots, z_n$, we first need to generate a corresponding set of uniformly distributed random variables $u_1, u_2, \ldots, u_n$, between 0 and 1. Then, for each u_i, we can generate a value z_i using

$$z_i = \Phi^{-1}(u_i) \tag{4.3}$$

where Φ^{-1} is the inverse of the standard normal cumulative distribution function. Figure 4.3 shows this relationship graphically.

Equation 2.43 in Chapter 2 can be used to estimate the inverse function in Eq. 4.3. Many computer programs have a standard normal random number generator built in.

4.1.4 Generation of Normal Random Numbers

In the previous section, we discussed how to generate sample values of a normally distributed random variable following a *standard* normal distribution. What if we need to generate sample values from an arbitrary normal distribution? To do this, we use the relationship discussed in Chapter 2. Assume we have a normally distributed random variable X with mean μ_X and standard deviation σ_X. The basic relationship between X and the standard normal variate Z is

$$X = \mu_X + Z\sigma_X \tag{4.4}$$

So, given a sample value z_i generated using the approach discussed in Section 4.1.3, the corresponding x_i value can be calculated using

$$x_i = \mu_X + z_i\sigma_X \tag{4.5}$$

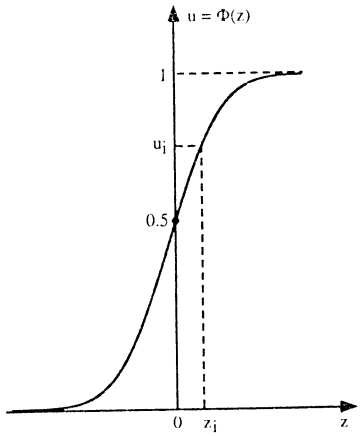

FIGURE 4.3 Generation of standard normal random variables.

EXAMPLE 4.2. Assume that the dead load D on a structure is a normally distributed random variable. The mean μ_D is 2.0 k/ft, and coefficient of variation V_D is 10 percent. The PDF for dead load is shown in Figure 4.4. Generate 10 values of the random variable D.

Solution. To begin, we need to generate some uniformly distributed random variables. For this example, the first 10 values in column 1 of Table 4.1 are used. Then Eq. 4.3 is used to generate corresponding z_i values. Finally, Eq. 4.5 is used to convert the z_i values to D_i values. Note that $\sigma_D = V_D\mu_D = 0.2$ k/ft. The numerical results are tabulated in Table 4.2.

TABLE 4.2 Summary of calculations for Example 4.2.

u_i	z_i	D_i
0.050203	−1.64289	1.67
0.619129	0.303194	2.06
0.872402	1.13782	2.23
0.376568	−0.314507	1.94
0.139927	−1.08065	1.78
0.318491	−0.471923	1.91
0.987671	2.24672	2.45
0.033265	−1.83483	1.63
0.234626	−0.723697	1.86
0.623157	0.313783	2.06

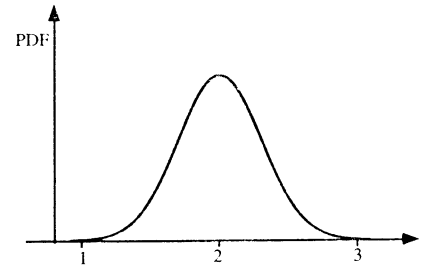

FIGURE 4.4 Probability density function used in Example 4.2.

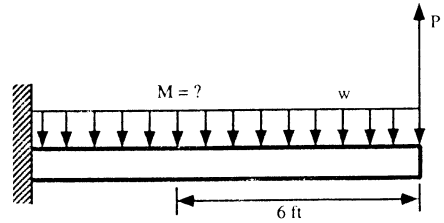

FIGURE 4.5 Example of a wood cantilever beam.

EXAMPLE 4.3. Consider a wood cantilever beam such as the one shown in Figure 4.5. Using the Monte Carlo technique, calculate the mean and standard deviation of the moment M at a distance of 6 ft from the free end. The loads P and w are independent normal random variables with the following parameters:

$$\mu_P = 4000 \text{ lb} \quad \mu_W = 50 \text{ lb/ft}$$
$$\sigma_P = 400 \text{ lb} \quad \sigma_W = 5 \text{ lb/ft}$$

Solution. Using statics, we know the bending moment M at the location of interest is

$$M = 6P - 18w$$

Since P and w are both independent normal random variables and M is a linear function of P and w, we can say that M is also a normal random variable. The mean and standard deviation of M can be calculated *exactly* using Eqs. 3.2 and 3.4 as follows:

$$\mu_M = 6\mu_P - 18\mu_W = 23{,}100 \text{ lb-ft} = 23.1 \text{ k-ft}$$

$$\sigma_M = \sqrt{(6\sigma_P)^2 + (18\sigma_W)^2} = 2400 \text{ lb-ft} = 2.40 \text{ k-ft}$$

As we will demonstrate, we can also use Monte Carlo simulation. After the simulation, we will be able to see how the simulation results compare with the theoretically correct results.

For this example, we will calculate five values of the bending moment M using Monte Carlo simulation. Therefore, we will need to simulate five values of the load P and five values of the load w. This will require us to determine 10 values (two simulated variables times five values of each variable) of a uniformly distributed random variable between 0 and 1. The uniform random numbers for this example are taken from Table 4.1. We will take the first 10 values in column 1. The first five will be used to calculate five values of P, and the last five will be used to calculate five values of w. The simulated values of P and w are calculated using Eq. 4.5 and are summarized in Table 4.3 for convenience.

Next, five data sets $\{P_i, w_i\}$ are formed and used to generate five values for the moment M using

$$M_i = 6P_i - 18w_i$$

The following results are generated for M_i (in lb-ft):

19,200
23,620
25,990
22,410
29,480

Hence the sample mean and sample standard deviation can be calculated using Eqs. 2.25 and 2.26 from Chapter 2 as follows:

$$\bar{M} = \frac{1}{5}\sum_{i=1}^{5} M_i = 22{,}340 \text{ lb-ft} = 22.34 \text{ k-ft}$$

$$s_M = \sqrt{\frac{\left(\sum_{i=1}^{5} M_i^2\right) - 5(\bar{M})^2}{5-1}} = 2659 \text{ lb-ft} = 2.659 \text{ k-ft}$$

TABLE 4.3 Tabulation of calculations for Example 4.3

u_i	z_i	$P_i = 4000 + z_i\,(400)$
0.050203	−1.64289	3343
0.619129	0.303194	4121
0.872402	1.13782	4455
0.376568	−0.314507	3874
0.139927	−1.08065	3568
u_i	z_i	$w_i = 50 + z_i\,(5)$
0.318491	−0.471923	47.64
0.987671	2.24672	61.23
0.033265	−1.83483	40.83
0.234626	0.723697	46.38
0.623157	0.313783	51.57

Comparing these values with the theoretically exact values obtained earlier, we see that the values are comparable but not as accurate as we might like. The accuracy of the Monte Carlo simulation approach increases as the number of sample values increases. If more simulated values of M were generated, we would expect our estimates of mean and standard deviation to improve. To illustrate this, the preceding calculation was repeated with the following input:

- The first 50 values of uniform random variables found in the first two columns of Table 4.1 were used to simulate values of P.
- The next 50 values of uniform random variables found in columns 3 and 4 of Table 4.1 were used to simulate values of w.
- Using the simulated values of P and w, 50 values of M were generated, and the mean and standard deviation were calculated.

Using these 50 simulated values of M, the sample mean and sample standard deviation were found to be 22.99 k-ft and 2.332 k-ft, respectively. These values are much closer to the theoretically correct values of 23.1 k-ft and 2.40 k-ft.

4.1.5 Generation of Lognormal Random Numbers

Let X be a lognormal random variable with mean μ_X and standard deviation σ_X. To generate a sample value x_i, we begin as before by generating a sample value u_i of a uniformly distributed random number such that $0 \le u_i \le 1$. Then, a sample value z_i from a standard normal distribution is calculated using Eq. 4.3. Finally, using the relationship between normal and lognormal variables, we obtain x_i using

$$x_i = \exp\left[\mu_{\ln X} + z_i \sigma_{\ln X}\right] \tag{4.6}$$

where

$$\sigma^2_{\ln X} = \ln\left(V_X^2 + 1\right) \tag{4.7}$$

$$\approx V_X^2 \qquad (\text{for } V_X < 0.20)$$

$$\mu_{\ln X} = \ln(\mu_X) - \frac{1}{2}\sigma^2_{\ln X} \tag{4.8}$$

$$\approx \ln(\mu_X) \qquad (\text{for } V_X < 0.20)$$

If the approximate relationships in Eqs. 4.7 and 4.8 are used, Eq. 4.6 can be rewritten as

$$x_i = \mu_X \exp\left[z_i V_X\right] \tag{4.9}$$

4.1.6 General Procedure for Generating Random Numbers from an Arbitrary Distribution

In the preceding sections, we considered generation of sample values of random variables for some of the most common distributions used in structural reliability analysis. A general procedure can be formulated which, theoretically, is applicable to any type of distribution function.

Consider a random variable X with a cumulative distribution function $F_X(x)$. To generate sample values x_i for the random variable, the following steps can be taken:

1. Generate a sample value u_i for a uniformly distributed random variable between 0 and 1.
2. Calculate a sample value x_i from the following formula:

$$x_i = F_X^{-1}(u_i) \tag{4.10}$$

where F_X^{-1} is the inverse of F_X.

This procedure is completely general. However, in some cases it is difficult to determine a closed-form solution for the inverse CDF.

4.1.7 Accuracy of Probability Estimates

In Example 4.1, we discussed how simulation results can be used to estimate a probability of failure. It is important to recognize that this estimate of probability is indeed only an estimate. However, the estimate improves as the number of simulations increases. In this section, we look at the relationship between the uncertainty in the estimated probability and the number of simulations.

The estimated probability $\bar{P}$ is calculated as the ratio

$$\bar{P} = \frac{n}{N} \tag{4.11}$$

where N is the total number of simulations and n is the number of times (out of N simulations) that a particular criterion was achieved. For example, suppose 100 simulations are conducted to see how many times $Y < 0$ for Example 4.1. Of those 100 simulations, five simulations result in $Y < 0$. Thus $n = 5$ and $N = 100$, so our estimate of the probability $P(Y < 0)$ is $5/100 = 0.05$. Now suppose a completely new simulation is done with $N = 100$. This time we get $n = 7$ and our estimate of $P(Y < 0)$ is $7/100 = 0.07$. The point is that the estimated probability $\bar{P}$ is a "sample estimate." By this we mean that the calculated value will vary from sample to sample. Therefore, the estimated probability itself can be treated as a random variable with its own mean, standard deviation, and coefficient of variation.

Let P_{true} be the theoretically correct probability we are trying to estimate by calculating $\bar{P}$. It can be shown (e.g., Soong and Grigoriu, 1993) that the expected value, variance, and coefficient of variation of the estimated probability $\bar{P}$ are as follows:

$$E[\bar{P}] = P_{true}; \qquad \sigma_{\bar{P}}^2 = \frac{1}{N}[P_{true}(1 - P_{true})]; \qquad V_{\bar{P}} = \sqrt{\frac{(1 - P_{true})}{N(P_{true})}} \tag{4.12}$$

Observe that the "uncertainty" in the estimate of the probability decreases as the total number of simulations, N, increases. These relationships provide a way to determine how many simulations are required to estimate a probability and limit the uncertainty in the estimate.

EXAMPLE 4.4. Suppose we want to be able to estimate probabilities as low as 10^{-2} and keep the coefficient of variation of our estimate at or below 10 percent. How many simulations are required?

Solution. By rearranging the expression for coefficient of variation in Eqs. 4.12, we can solve for the number of simulations N as

$$N = \frac{1 - P_{true}}{V_{\bar{P}}^2(P_{true})} = \frac{1 - 10^{-2}}{(0.10)^2(10^{-2})} = 9900$$

To estimate relatively small probabilities using Monte Carlo simulation while controlling the uncertainty in the estimate, a large number of simulations is needed. The required sample size depends on the desired coefficient of variation and the relative magnitude of the probability to be estimated.

EXAMPLE 4.5. Consider the function

$$Y = R - Q$$

where R and Q are both random variables. R is lognormally distributed with a mean value of 200 and a standard deviation of 20. Q follows an extreme Type I distribution with a mean value of 100 and a standard deviation of 12. The goal is to determine the probability of $Y < 0$ using simulation.

Solution

1. Determine the number (N) of desired simulated values. To keep the example simple, we will choose $N = 25$.
2. Generate $N = 25$ uniform random variables for simulating R and another $N = 25$ uniform random variables for simulating Q. For this example, we will use column 1 of Table 4.1 for simulating R and column 2 of the same table for simulating Q.
3. Generate 25 values of R and 25 values of Q. These will be referred to as r_i and q_i ($i = 1, 2, \ldots, 25$). To generate the r_i values, we can use Eq. 4.6. Note that this requires us to determine $\mu_{\ln R}$ and $\sigma_{\ln R}$. Using Eqs. 4.7 and 4.8, $\mu_{\ln R} = 5.29$ and $\sigma_{\ln R} = 0.0998$. To simulate values of Q, we need to use Eqs. 2.58, 2.62, 2.63, and 4.10. Equation 4.10 requires the inverse of the Type I distribution, which can be obtained in closed form. The result is

$$q_i = F_Q^{-1}(u_i) = 94.6 - \frac{\ln[-\ln(u_i)]}{0.107}$$

4. Calculate N values of y_i using $y_i = r_i - q_i$. The simulated values of r_i, q_i, and y_i are listed in Table 4.4.
5. Estimate the probability of $Y < 0$ using the simulation data. By observing the values in the far right column of Table 4.4, we see that there are no values of y_i below zero. Thus Eq. 4.11 would lead to $P(Y < 0) = 0$ which is not reasonable. As will be shown in Example 5.9 of Chapter 5, it turns out that the probability of $Y < 0$ for this problem is on the order of 10^{-4} to 10^{-5}. A calculation similar to that done in Example 4.4 would quickly tell us that $N = 25$ is simply too small to make a reliable estimate

TABLE 4.4 Simulated values of R, Q, and Y = R − Q for Example 4.5

u_i	z_i	r_i	u_i	q_i	$y_i = r_i - q_i$
0.050203	−1.643	168.3	0.269082	92.1	76.3
0.619129	0.303	204.4	0.472640	97.3	107.1
0.872402	1.138	222.2	0.422864	96.0	126.2
0.376568	−0.315	192.2	0.467299	97.2	95.1
0.139927	−1.081	178.1	0.415784	95.8	82.2
0.318491	−0.472	189.2	0.523667	98.7	90.5
0.987671	2.247	248.2	0.243629	91.4	156.8
0.033265	−1.835	165.2	0.741569	105.9	59.3
0.234626	−0.724	184.5	0.408673	95.6	88.9
0.623157	0.314	204.7	0.021790	82.1	122.6
0.957884	1.727	235.6	0.547472	99.3	136.3
0.518906	0.047	199.3	0.749779	106.2	93.0
0.442305	−0.145	195.5	0.681600	103.6	91.9
0.445845	−0.136	195.7	0.140538	88.3	107.4
0.834284	0.971	218.5	0.888607	114.6	104.0
0.811213	0.882	216.6	0.105136	87.0	129.6
0.935728	1.520	230.8	0.067202	85.3	145.5
0.450423	−0.125	195.9	0.864772	112.6	83.3
0.579058	0.199	202.3	0.363628	94.5	107.8
0.662648	0.420	206.8	0.863948	112.6	94.3
0.039918	−1.752	166.5	0.858242	112.2	54.4
0.414075	−0.217	194.1	0.432234	96.2	97.8
0.103214	−1.263	174.8	0.412091	95.7	79.1
0.112308	−1.214	175.7	0.749199	106.2	69.5
0.821833	0.922	217.5	0.260933	91.8	125.6

of such a small probability using Eq. 4.11. As an alternative, we can plot the data on normal probability paper as shown in Figure 4.6. A *very rough* estimate of the failure probability can be obtained by extrapolating the curve to the left until it reaches the y axis. This extrapolation is shown by the dashed line in Figure 4.6. The intersection point corresponds to a value of the standard normal variable in the neighborhood of −4. Therefore, the estimate of the failure probability is $\Phi(-4) = 3.17 \times 10^{-5}$.

4.1.8 Simulation of Correlated Normal Random Variables

In the preceding sections, we covered simulation of random variables, and it was implicitly assumed that the variables were uncorrelated. In practice, the variables may be correlated, so our simulation procedure must be able to simulate the correlation. A transformation technique for simulating correlated normal random variables is covered in this section. Although it is strictly valid for normal random variables only, it can be used for other types of random variables *as an approximation*.

Let $X_1, X_2, \ldots, X_n$ be *correlated normal* random variables. The mean values and covariance matrix are given by

$$\{\mu_X\} = \{\mu_{X_1}, \mu_{X_2}, \ldots, \mu_{X_n}\} \tag{4.13}$$

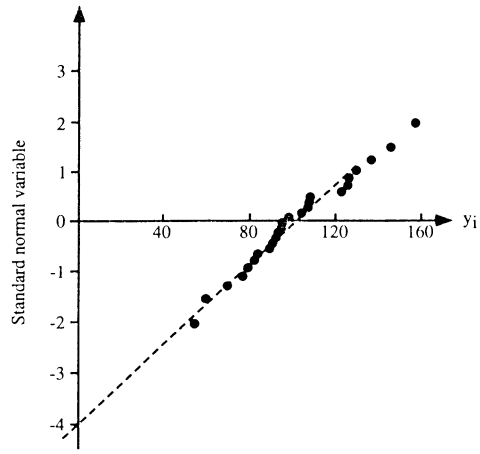

FIGURE 4.6 Simulated data from Example 4.5 plotted on normal probability paper.

$$[C_X] = \begin{bmatrix} \mathrm{CoV}(X_1, X_1) & \mathrm{CoV}(X_1, X_2) & \cdots & \cdots & \mathrm{CoV}(X_1, X_n) \\ \mathrm{CoV}(X_2, X_1) & \mathrm{CoV}(X_2, X_2) & & & \mathrm{CoV}(X_2, X_n) \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \mathrm{CoV}(X_n, X_1) & \mathrm{CoV}(X_n, X_2) & \cdots & \cdots & \mathrm{CoV}(X_n, X_n) \end{bmatrix} \tag{4.14}$$

To generate correlated random numbers for $X_1, X_2, \ldots, X_n$, it is necessary to first generate a set of *uncorrelated* random numbers $Y_1, Y_2, \ldots, Y_n$ using the techniques discussed earlier. Then $X_1, X_2, \ldots, X_n$ are calculated using the variable transformation

$$\{X\} = [T]\{Y\} \tag{4.15}$$

where [T] is a transformation matrix. To apply this approach, we need to determine the matrix [T] as well as the mean and variance values for the uncorrelated Y_i variables. To do this, we must use some concepts from linear algebra. The following discussion assumes that the reader has some familiarity with the concepts of eigenvalues and eigenvectors. For information on these concepts, see the reference by Kaplan (1991) or any other text covering linear algebra.

Let [A] be an n × n matrix (i.e., a square matrix with n rows and n columns) which is symmetric. A *diagonal* matrix [D] and a square matrix [T] can be found such that the following relationships hold:

$$[D] = [T]^T[A][T] \tag{4.16}$$

$$[A] = [T][D][T]^T \tag{4.17}$$

The superscript T denotes transpose. The matrix [T] contains the *orthonormal eigenvectors* corresponding to the eigenvalues of the matrix [A]. The diagonal matrix [D] contains the *eigenvalues* of [A].

In the present context of simulating random variables, the matrix [A] is the covariance matrix $[C_X]$ of the original, correlated variables {X}. The matrix [T] is made up of the orthonormal eigenvectors corresponding to the eigenvalues of the matrix $[C_X]$. Thus the first column of [T] contains the orthonormal eigenvector corresponding to the first eigenvalue, the second column contains the eigenvector corresponding to the second eigenvalue, and so on. The matrix [T] is an *orthogonal matrix*, meaning that its inverse is equal to its transpose. The diagonal matrix [D] corresponds to the covariance matrix $[C_Y]$ of the uncorrelated variables {Y}. With these changes in notation, Eqs. 4.16 and 4.17 become

$$[C_Y] = [T]^T [C_X] [T] = \begin{bmatrix} \sigma^2_{Y_1} & 0 & \cdots & \cdots & 0 \\ 0 & \sigma^2_{Y_2} & & & \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & \sigma^2_{Y_n} \end{bmatrix} \tag{4.18}$$

$$[C_X] = [T][C_Y][T]^T \tag{4.19}$$

The diagonal elements of $[C_Y]$ contain the variances of the uncorrelated Y variables needed to do the simulation. The mean values of the Y_i variables can be obtained using

$$\{\mu_Y\} = [T]^T \{\mu_X\} \tag{4.20}$$

Once simulated values of {Y} are obtained, Eq. 4.15 can be used to obtain simulated values of {X}.

EXAMPLE 4.6. Suppose we have two correlated normal random variables X_1 and X_2. The variable X_1 has a mean value of 10 and a coefficient of variation of 10 percent. The variable X_2 has a mean value of 15 and a coefficient of variation of 20 percent. The correlation coefficient between the two variables is $\rho_{12} = 0.5$. Determine the mean values and variances of the transformed coordinates Y_1 and Y_2 that must be considered to simulate X_1 and X_2.

Solution. Based on the given information, the standard deviations of X_1 and X_2 are

$$\sigma_{X_1} = V_{X_1}\mu_{X_1} = (0.10)(10) = 1; \qquad \sigma_{X_2} = V_{X_2}\mu_{X_2} = (0.20)(15) = 3$$

The vector of mean values and the covariance matrix for the original variables X_1 and X_2 are

$$\begin{Bmatrix} \mu_{X_1} \\ \mu_{X_2} \end{Bmatrix} = \begin{Bmatrix} 10 \\ 15 \end{Bmatrix}$$

$$[C_X] = \begin{bmatrix} \sigma^2_{X_1} & \rho_{12}\sigma_{X_1}\sigma_{X_2} \\ \rho_{12}\sigma_{X_1}\sigma_{X_2} & \sigma^2_{X_2} \end{bmatrix} = \begin{bmatrix} 1 & 1.5 \\ 1.5 & 9 \end{bmatrix}$$

By doing an eigenvalue and eigenvector analysis of $[C_X]$, we find the transformation matrix [T] to be

$$[T] = \begin{bmatrix} 0.984 & 0.178 \\ -0.178 & 0.984 \end{bmatrix}$$

(Note that the eigenvectors must be suitably normalized to qualify as orthonormal eigenvectors.) With this information, the mean values and covariance matrix for the Y variables can be found using Eqs. 4.18 and 4.20 as follows:

$$\begin{Bmatrix} \mu_{Y_1} \\ \mu_{Y_2} \end{Bmatrix} = [T]^T \begin{Bmatrix} \mu_{X_1} \\ \mu_{X_2} \end{Bmatrix} = \begin{bmatrix} 0.984 & -0.178 \\ 0.178 & 0.984 \end{bmatrix} \begin{Bmatrix} 10 \\ 15 \end{Bmatrix} = \begin{Bmatrix} 7.16 \\ 16.5 \end{Bmatrix}$$

$$[C_Y] = [T]^T [C_X] [T] = \begin{bmatrix} 0.984 & -0.178 \\ 0.178 & 0.984 \end{bmatrix} \begin{bmatrix} 1 & 1.5 \\ 1.5 & 9 \end{bmatrix} \begin{bmatrix} 0.984 & 0.178 \\ -0.178 & 0.984 \end{bmatrix}$$

$$= \begin{bmatrix} 0.728 & 0 \\ 0 & 9.27 \end{bmatrix}$$

4.2 LATIN HYPERCUBE SAMPLING

The techniques of random sampling and Monte Carlo simulation discussed in Section 4.1 are very powerful and useful techniques for performing probabilistic analyses. However, in some instances, the problem being analyzed is extremely complex, and the time needed to evaluate the problem for a single trial (N = 1) may be very long. As a result, the time needed to perform hundreds or thousands of simulations may be unfeasible.

The Latin hypercube method is one technique for reducing the number of simulations needed to obtain reasonable results. In this method, the range of possible values of each random input variable is partitioned into "strata," and a value from each stratum is randomly selected as a representative value. The representative values for each random variable are then combined so that each representative value is considered once and only once in the simulation process. In this way, all possible values of the random variables are represented in the simulation.

To be specific, let's assume that we need to simulate values of some function Y described by

$$Y = f(X_1, X_2, \ldots, X_K) \tag{4.21}$$

where f( ) is some deterministic function (but possibly not known in closed form) and the $X_i (i = 1, 2, \ldots, K)$ are the random input variables. The basic steps in Latin hypercube sampling are as follows:

1. Partition the range of each X_i into N intervals. The partitioning should be done so that the probability of a value of X_i occurring in each interval is 1/N. [This restriction simplifies the presentation but is not strictly necessary. For additional details, see Iman and Conover (1980).]
2. For each X_i variable and each of its N intervals, randomly select a representative value for the interval. In practical applications, if the number of intervals is large, the center point (i.e., the middle value) of each interval can be used instead of doing random sampling.
3. After Steps 1 and 2, there will be N representative values for each of the K random variables. There are N^K possible combinations of these representative values. The objective of Latin hypercube sampling is to select N combinations such that each representative value appears once and only once in the N combinations.

4. To obtain the first combination, randomly select one of the representative values for each of the K input random variables. To obtain the second combination, randomly select one of the N − 1 remaining representative values of each random variable. To obtain the third combination, randomly select one of the N − 2 remaining representative values of each random variable. Continue this selection process until you have N combinations of values of the input random variables.
5. Evaluate Eq. 4.21 for each of the N combinations of input variables generated above. This will lead to N values of the function. These values will be referred to as $y_i (i = 1, 2, \ldots, N)$.

This procedure provides the simulated data. Now we must determine how to use the data to estimate probabilities or statistical parameters for Y. The most commonly used formulas include the following:

$$\text{Estimated mean value of } Y = \bar{Y} = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{4.22}$$

$$\text{Estimated } m^{\text{th}} \text{ moment of } Y = \frac{1}{N}\sum_{i=1}^{N} (y_i)^m \tag{4.23}$$

$$\text{Estimated CDF } F_Y(y) = \frac{\text{number of times } y_i \le y}{N} \tag{4.24}$$

These formulas have the same format as those presented in Chapter 2 (Section 2.3.2) and those presented in Section 4.1.1.

EXAMPLE 4.7. To illustrate how Latin hypercube sampling is implemented, we consider a simple example involving two random variables. We want to estimate the mean value of the following function of two random variables:

$$Y = 3X_1 - 7X_2$$

The variable X_1 is uniformly distributed between 2 and 4, and X_2 is normally distributed with mean $\mu = 0.6$ and standard deviation $\sigma = 0.1$. Since there are two random variables, K = 2.

Solution. For this simple illustration, assume that it is sufficient to define four intervals (N = 4) for each random variable. Each interval must have an equal probability (1/N = 0.25) of a value in the interval. The corresponding intervals are as follows:

X_1 (uniform):	(2, 2.5)	(2.5, 3)	(3, 3.5)	(3.5, 4)
X_2 (normal):	$(-\infty, 0.533)$	(0.533, 0.6)	(0.6, 0.667)	$(0.667, +\infty)$

Now we must randomly select a value from each interval as a representative value of the interval. Suppose that we do this and obtain the following representative values:

X_1 (uniform):	(2.23)	(2.90)	(3.04)	(3.81)
X_2 (normal):	(0.51)	(0.55)	(0.61)	(1.5)

There are a total of $N^K = (4)^2 = 16$ possible combinations of these representative values. We must now randomly create four pairs such that each representative value appears once and only once. Suppose the following pairs are created:

$$(x_1, x_2) = (2.23, 0.55); \quad (2.90, 1.5); \quad (3.04, 0.61); \quad (3.81, 0.51)$$

For each of these pairs, we must compute the corresponding value of Y:

$$y_1(2.23, 0.55) = 2.84$$
$$y_2(2.90, 1.5) = -1.8$$
$$y_3(3.04, 0.61) = 4.85$$
$$y_4(3.81, 0.51) = 7.86$$

These represent the four simulated values (N = 4) of Y. The sample mean is obtained using Eq. 4.22:

$$\bar{Y} = \frac{1}{N}\sum_{i=1}^{4} y_i = \frac{1}{4}(2.84 - 1.8 + 4.85 + 7.86) = 3.44$$

For comparison, since the function is linear, Eq. 3.2 can be used to calculate the exact mean value of the function:

$$\mu_Y = 3(\mu_{X_1}) - 7(\mu_{X_2}) = 4.8$$

The agreement between the exact and the estimated mean value is not very good because we used only four intervals for each random variable. By choosing more intervals, we could improve our estimate.

For additional details on the Latin hypercube sampling method and its advantages and disadvantages, see Dunn and Clark (1974), McKay, Beckman, and Conover (1979), Iman and Conover (1980), and Press, Teukolsky, Vetterling, and Flannery (1992). For applications of this method to structural reliability analysis, see Han and Wen (1994) and Zhou and Nowak (1988).

4.3 ROSENBLUETH'S 2K+1 POINT ESTIMATE METHOD

Complex problems may also be solved by using a point estimate method. Although there are many such methods, the 2K+1 method proposed by Rosenblueth (1975) is one of the easiest to implement. Loosely speaking, this method may be thought of as a simulation technique in which the number of simulations is N = 2K + 1, where K is the number of input random variables. The basic idea is to evaluate a function

of random variables at 2K+1 key points and then to use this information to estimate the mean and variance (or coefficient of variation) of the function. However, the CDF of the function cannot be obtained by this method.

Consider a function Y described by

$$Y = f(X_1, X_2, \ldots, X_K) \tag{4.25}$$

where $f(\ )$ is some deterministic function (but possibly not known in closed form) and the $X_i (i = 1, 2, \ldots, K)$ are the random input variables. The steps in Rosenblueth's 2K+1 method are as follows:

1. Determine the mean value (μ_{X_i}) and standard deviation (σ_{X_i}) for each of the K input random variables.
2. Define y_0 as the value of Eq. 4.25 when all input variables are equal to their mean values, that is,
$$y_0 = f(\mu_{X_1}, \mu_{X_2}, \ldots, \mu_{X_K}) \tag{4.26}$$
3. Evaluate the function Y at 2K additional points as follows. For each random variable X_i, evaluate the function at two values of X_i which are shifted from the mean value μ_{X_i} by $\pm\sigma_{X_i}$ while all other variables are assumed to be equal to their mean values. These values of the function will be referred to as y_i^+ and y_i^-. The subscript denotes the variable which is shifted, and the superscript indicates the direction of the shift. In mathematical notation,
$$y_i^+ = f(\mu_{X_1}, \mu_{X_2}, \ldots, \mu_{X_i} + \sigma_{X_i}, \ldots, \mu_{X_K}) \tag{4.27a}$$
$$y_i^- = f(\mu_{X_1}, \mu_{X_2}, \ldots, \mu_{X_i} - \sigma_{X_i}, \ldots, \mu_{X_K}) \tag{4.27b}$$
4. For each random variable, calculate the following two quantities based on y_i^+ and y_i^-:
$$\bar{y}_i = \frac{y_i^+ + y_i^-}{2} \tag{4.28a}$$
$$V_{y_i} = \frac{y_i^+ - y_i^-}{y_i^+ + y_i^-} \tag{4.28b}$$
5. Calculate the estimated mean and coefficient of variation of Y as follows:
$$\bar{Y} = y_0 \prod_{i=1}^{K} \left(\frac{\bar{y}_i}{y_0}\right) \tag{4.29a}$$
$$V_Y = \sqrt{\left\{\prod_{i=1}^{K}\left(1 + V_{y_i}^2\right)\right\} - 1} \tag{4.29b}$$

There are two distinct advantages to this method. First, it is not necessary to know the distributions of the input random variables; only the first two moments are needed. Second, the number of function evaluations (i.e., "simulations") is relatively small compared to Latin hypercube sampling or general Monte Carlo simulation. For a discussion of the assumptions and theoretical basis for this method, see the original paper by Rosenblueth (1975). For sample applications of this method in structural reliability calculations, see Tantawi (1986).

EXAMPLE 4.8. For the function considered in Example 4.7, estimate the mean and coefficient of variation using Rosenblueth's 2K+1 method.

Solution. The function is

$$Y = 3X_1 - 7X_2$$

where X_1 is uniformly distributed between 2 and 4, and X_2 is normally distributed with mean $\mu = 0.6$ and standard deviation $\sigma = 0.1$. Since there are two random variables, $K = 2$. For X_1, based on the given distribution information, the mean value is 3 and the variance is 1/3. The following procedure is used:

1. Determine mean values and standard deviations. This has been done.
2. Calculate y_0. Using Eq. 4.26,
$$y_0 = 3\mu_{X_1} - 7\mu_{X_2} = 3(3) - 7(0.6) = 4.8$$
3. Evaluate the function at $2K = 2(2) = 4$ additional points using Eqs. 4.27a and b.
$$y_1^+ = f(\mu_{X_1} + \sigma_{X_1}, \mu_{X_2}) = 3(3 + \sqrt{1/3}) - 7(0.6) = 6.53$$
$$y_1^- = f(\mu_{X_1} - \sigma_{X_1}, \mu_{X_2}) = 3(3 - \sqrt{1/3}) - 7(0.6) = 3.07$$
$$y_2^+ = f(\mu_{X_1}, \mu_{X_2} + \sigma_{X_2}) = 3(3) - 7(0.6 + 0.1) = 4.10$$
$$y_2^- = f(\mu_{X_1}, \mu_{X_2} - \sigma_{X_2}) = 3(3) - 7(0.6 - 0.1) = 5.50$$
4. Use Eqs. 4.28a and 4.28b to calculate some intermediate quantities:
$$\bar{y}_1 = \frac{y_1^+ + y_1^-}{2} = \frac{6.53 + 3.07}{2} = 4.80$$
$$\bar{y}_2 = \frac{y_2^+ + y_2^-}{2} = \frac{4.10 + 5.50}{2} = 4.80$$
(By coincidence, these two values are the same as y_0. In general, they are different.)
$$V_{y_1} = \frac{y_1^+ - y_1^-}{y_1^+ + y_1^-} = \frac{6.53 - 3.07}{6.53 + 3.07} = 0.360$$
$$V_{y_2} = \frac{y_2^+ - y_2^-}{y_2^+ + y_2^-} = \frac{4.10 - 5.50}{4.10 + 5.50} = -0.146$$
5. Use Eqs. 4.29a and 4.29b to estimate the mean and coefficient of variation of Y:
$$\bar{Y} = y_0 \prod_{i=1}^{2}\left(\frac{\bar{y}_i}{y_0}\right) = 4.80\left(\frac{4.80}{4.80}\right)\left(\frac{4.80}{4.80}\right) = 4.80$$
$$V_Y = \sqrt{\left\{\prod_{i=1}^{2}\left(1 + V_{y_i}^2\right)\right\} - 1} = \sqrt{\left(1 + (0.360)^2\right)\left(1 + (-0.146)^2\right) - 1} = 0.392$$

For comparison purposes, consider the exact mean and coefficient of variation. In Example 4.7, the exact mean value of this function was found to be 4.80. From Eq. 3.4, the exact variance (assuming no correlation) is

$$\sigma_Y^2 = \sum_{i=1}^{2} a_i^2 \sigma_{X_i}^2 = (3)^2(\sqrt{1/3})^2 + (-7)^2(0.1)^2 = 3.49$$

Therefore, the exact coefficient of variation is

$$V_Y = \frac{\sigma_Y}{\mu_Y} = \frac{\sqrt{3.49}}{4.80} = 0.389$$

For this simple problem, the estimated and exact mean value are identical. The estimated and exact values of the coefficient of variation are very close.

PROBLEMS

4.1. Consider the following limit state function:

$$g(R, D, L) = R - D - L$$

where R is the load-carrying capacity, D is the dead load effect, and L is the live load effect. The load-carrying capacity is lognormally distributed with a nominal value of 245 k-ft, a bias factor of 1.12, and a coefficient of variation of 13 percent. (The bias factor is defined as the ratio of the mean value to the nominal value.) Both the dead load and the live load are normally distributed variables. The statistics of the dead load are: nominal value = 60 k-ft, bias factor = 1.03, and CoV = 8 percent. The statistics of the live load are: nominal value = 110 k-ft, bias factor = 0.85, and CoV = 13.5 percent. All three variables are uncorrelated. Calculate the probability of failure using the Monte Carlo simulation method. (To simplify the computations, limit the number of simulations to 25.)

4.2. Consider the following limit state equation for a reinforced concrete beam:

$$g = A_S F_Y \left(d - 0.59 \frac{A_S F_y}{f'_c b} \right) - [D + L]$$

The parameters of the random variables in this equation are listed in Table P4.1. The other parameters can be treated as deterministic variables. Their values are b = 15 in, d = 24 in, and $A_S = 8$ in^2. Use Monte Carlo simulation to predict the probability of failure.

TABLE P4.1

Parameter	Mean	CoV, %	Distribution
f'_c	3750 psi	12.5%	Normal
F_y	42.5 ksi	11.5	Lognormal
D	100 k-ft	10.5	Normal
L	200 k-ft	17.5	Extreme Type I

4.3. Consider Example 4.3. In that example, we used different sets of five uniformly distributed random numbers to generate five values of P and five values of w. Suppose we combined the two sets of five uniformly distributed random numbers to make a set of 10 numbers, and use this one set to generate 10 values of P and 10 values of w. In other words, each uniformly distributed number would be used twice. Do you see any problems with doing this? Explain your answer.

4.4. Consider Example 4.2. Using the computer program of your choice:

(*a*) Generate a set of 100 values of dead load.
(*b*) Calculate the sample mean and standard deviation of your simulated data and compare them with the values given in the problem.
(*c*) Plot the relative frequency histogram for the data. Does it resemble the PDF of a normal distribution? Should it?
(*d*) Plot the cumulative frequency histogram for the data. Does it resemble the CDF of a normal distribution? Should it?
(*e*) Plot the data on normal probability paper.
(*f*) Choose a new seed for your random number generator, and repeat (*a*) to (*e*). Compare the results of the two simulations.
(*g*) Repeat (*a*) and (*b*) for 1000 simulated data points.

4.5. Using the computer program of your choice:

(*a*) Simulate 100 values of uniformly distributed random integers between −3 and 4 inclusive.
(*b*) Plot a relative frequency histogram of the simulated data. Comment on what the ideal histogram should look like in this case, and comment on how your histogram compares with the ideal histogram.
(*c*) Repeat (*a*) and (*b*) for 1000 simulated values.

4.6. Use Monte Carlo simulation to solve Problem 3.6 at the end of Chapter 3.

4.7. Using the software program of your choice, solve Example 4.3 using 200 simulated data points.

4.8. Use Rosenblueth's method to solve Problem 3.4 (parts *a* and *b*) at the end of Chapter 3.

4.9. Use Rosenblueth's method to find the mean and coefficient of variation of the performance function discussed in Example 3.2.

4.10. Use Rosenblueth's method to find the mean and coefficient of variation of the function Y discussed in Example 3.3.

4.11. Using the software program of your choice:

(*a*) Solve Example 4.5 by Monte Carlo simulation. Estimate the required number of simulations needed to calculate the failure probability (estimated to be on the order