

Chapter 2

Uncertainty Analysis and Sampling Techniques

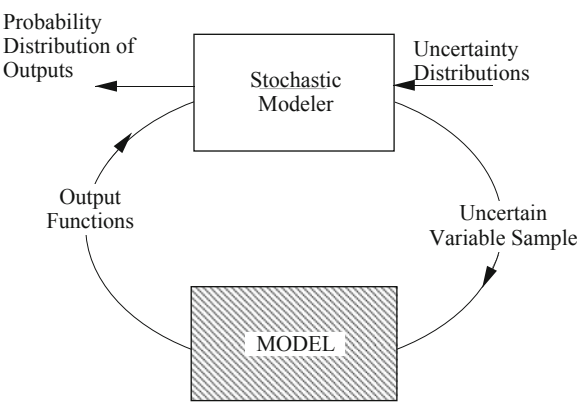
The probabilistic or stochastic modeling (Fig. 2.1) iterative loop in the stochastic optimization procedure (Fig. 1.4 in Chap. 1) involves:

1. Specifying the uncertainties in key input parameters in terms of probability distributions
2. Sampling the distribution of the specified parameter in an iterative fashion
3. Propagating the effects of uncertainties through the model and applying statistical techniques to analyze the results

2.1 Specifying Uncertainty Using Probability Distributions

To accommodate the diverse nature of uncertainty, different distributions can be used. Some of the representative distributions are shown in Fig. 2.2. The type of distribution chosen for an uncertain variable reflects the amount of information that is available. For example, the uniform and loguniform distributions represent an equal likelihood of a value lying anywhere within a specified range, on either a linear or logarithmic scale, respectively. Furthermore, a normal (Gaussian) distribution reflects a symmetric but varying probability of a parameter value being above or below the mean value. In contrast, lognormal and some triangular distributions are skewed such that there is a higher probability of values lying on one side of the median than the other. A beta distribution provides a wide range of shapes and is a very flexible means of representing variability over a fixed range. Modified forms of these distributions, uniform* and loguniform*, allow several intervals of the range to be distinguished. Finally, in some special cases, user-specified distributions can be used to represent any arbitrary characterization of uncertainty, including chance distribution (i.e., fixed probabilities of discrete values).

Fig. 2.1 The stochastic modeling framework



2.2 Sampling Techniques

Sampling is a statistical procedure which involves selecting a limited number of observations, states, or individuals from a population of interest. A sample is assumed to be representative of the whole population to which it belongs. Instead of evaluating all the members of the population, which would be time-consuming and costly, sampling techniques are used to infer some knowledge about the population. Sampling techniques can be divided into two groups: probability sampling and nonprobability

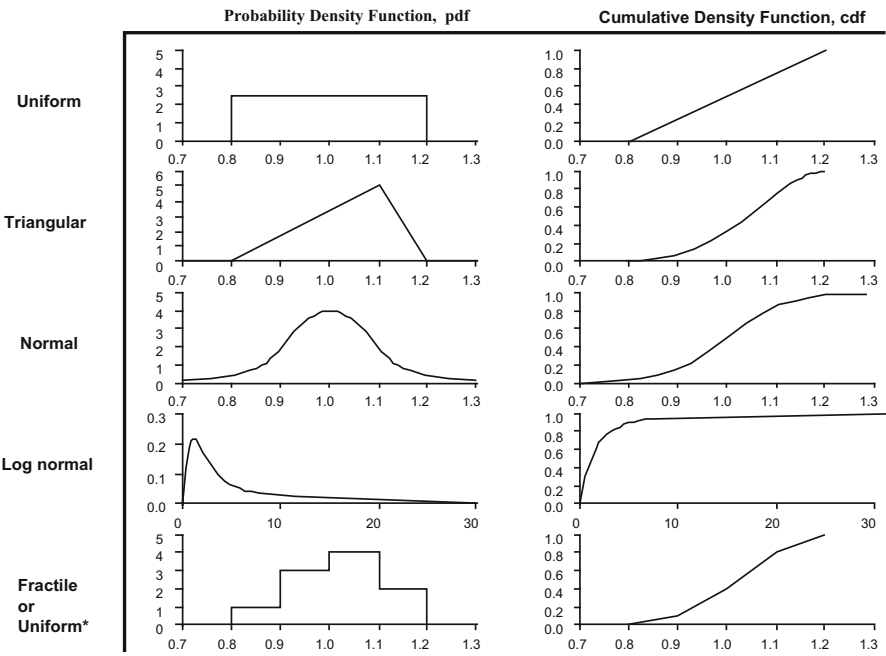


Fig. 2.2 Examples of probabilistic distribution functions for stochastic modeling

sampling. Probabilistic sampling techniques are based on Monte Carlo methods and are most relevant to this chapter. They are described in three subsections below. The description of the sampling techniques below is derived from the sampling chapter by Diwekar and Ulas [10].

2.2.1 Monte Carlo Sampling

One of the simplest and most widely used methods for sampling is the Monte Carlo method. Monte Carlo methods are numerical methods which provide approximate solutions to a variety of physical and mathematical problems by random sampling. The name Monte Carlo, which was suggested by Nicholas Metropolis, takes its name from a city in the Monaco principality which is famous for its casinos, because of the similarity between statistical experiments and the random nature of the games of chance such as roulette.

Monte Carlo methods were originally developed for the Manhattan Project during World War II, to simulate probabilistic problems related to random neutron diffusion in fissile material. Although they were limited by the computational tools of that time, they became widely used in many branches of science after the first electronic computers were built in 1945. The first publication which presents the Monte Carlo algorithm is probably by Metropolis and Ulam [33].

The basic idea behind Monte Carlo simulation has been that input samples should be randomly generated in order to describe a random output. In a crude Monte Carlo approach, a value is drawn at random from the probability distribution for each input, and the corresponding output value is computed. The entire process is repeated n times producing n corresponding output values. These output values constitute a random sample from the probability distribution over the output induced by the probability distributions over the inputs. The simplest distribution that is approximated by the Monte Carlo method is a uniform distribution $U(0, 1)$ with n samples on a k -dimensional unit hypercube. One advantage of this approach is that the precision of the output distribution may be estimated using standard statistical techniques. On average the error of approximation is of the order $O(N^{-1/2})$. One remarkable feature of this sampling technique is that the error bound is not dependent on the dimension k . However, this bound is probabilistic, which means that there is never any guarantee that the expected accuracy will be achieved in a concrete calculation.

The success of a Monte Carlo calculation depends on the choice of an appropriate random sample. The required random numbers and vectors are generated by the computer in a deterministic algorithm. Therefore, these numbers are called pseudorandom numbers or pseudorandom vectors. One of the oldest and best known methods for generating pseudorandom numbers for Monte Carlo sampling is the linear congruential generator (LCG) first introduced by D. H. Lehmer [30]. The general

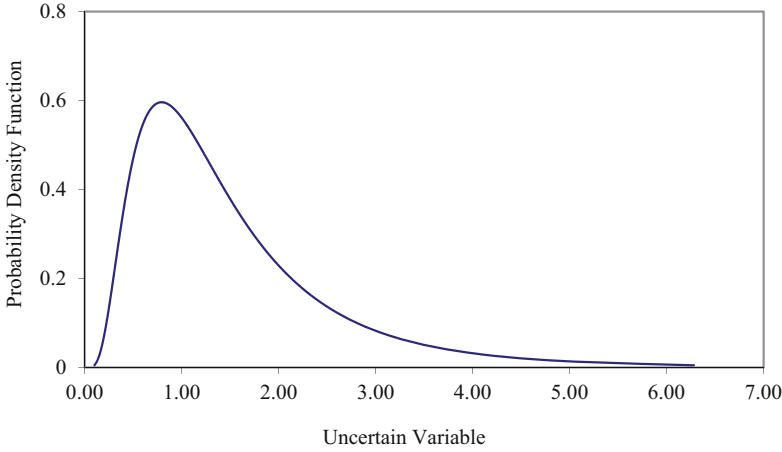


Fig. 2.3 PDF for a lognormal distribution. *PDF probability density function*

formula for a linear congruential generator is the following:

$$I_n = (aI_{n-1} + c) \bmod m \quad (2.1)$$

In this formula, a is the multiplier, c is the increment which is typically set to zero, and m is the modulus. These are preselected constants. The proper choice of these constants is very important for obtaining a sample which performs well in statistical tests. One other preselected constant is the seed, I_0 which is the first number in the output of a linear congruential generator. The random number generator used for Monte Carlo sampling provides a uniform distribution $U(0, 1)$. The specific values of each variable are selected by inverse transformation over the cumulative probability distribution. The following example shows how to generate a sample from pseudorandom numbers.

Example 2.1 We generated four pseudorandom numbers for sampling. These random numbers are $I_n = 0.6, 0.25, 0.925, 0.850$. Find the Monte Carlo samples for the lognormal distribution shown in Fig. 2.3.

Solution From the PDF shown in Fig. 2.3, we created the CDF (Fig. 2.4). We use the y-axis of Fig. 2.4 to place the random numbers on the figure and selected the corresponding x-axis numbers as samples in Table 2.1.

Pseudorandom numbers of different sample sizes on a unit square generated using the linear congruential generator are given in Fig. 2.5. From this figure it can be seen that the pseudorandom number generator produces samples that may be clustered in certain regions of the unit square and does not produce uniform samples. Therefore, in order to reach high accuracy, larger sample sizes are needed, which adversely affects the efficiency of this method. Variance reduction techniques address this problem of increasing efficiency of Monte Carlo methods and are described in the following section.

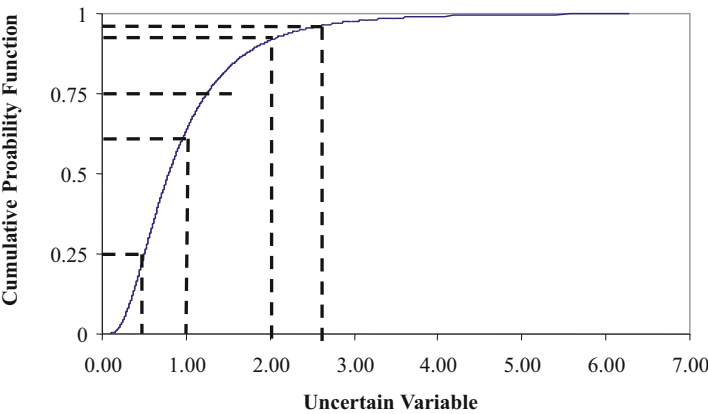


Fig. 2.4 Sample placement on the CDF. *CDF cumulative density function*

Table 2.1 Sample generation

Sample no.	Random number	Sample
1	0.6	1.0
2	0.25	0.5
3	0.925	2.6
4	0.850	2.0

2.3 Variance Reduction Techniques

To increase the efficiency of Monte Carlo simulations and overcome disadvantages such as probabilistic error bounds, variance reduction techniques have been developed [23].

The sampling approaches for variance reduction that are used most frequently in optimization under uncertainty are: importance sampling, Latin Hypercube Sampling (LHS) [22, 32], descriptive sampling, and Hammersley sequence sampling (HSS) [24]. The latter technique belongs to the group of quasi-Monte Carlo methods which were introduced in order to improve the efficiency of Monte Carlo methods by using quasi-random sequences that show better statistical properties and deterministic error bounds. These commonly used sampling techniques are described below with examples.

2.3.1 Importance Sampling

Importance sampling, which may also be called biased sampling, is a variance reduction technique for increasing the efficiency of Monte Carlo algorithms. Monte

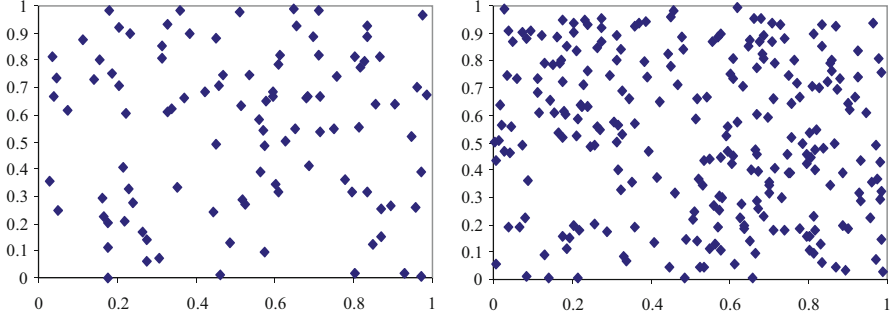


Fig. 2.5 (*Left hand side*) 100 pseudorandom numbers on a unit square, (*right hand side*) 250 pseudorandom numbers on a unit square obtained by the linear congruential generator developed by Wichmann and Hill [62]

Carlo methods are commonly used to integrate a function F over the domain D :

$$I = \int_D F(x) dx \quad (2.2)$$

The Monte Carlo integration for this function can be written as:

$$I_{mcs} = \frac{1}{N} \sum_{i=1}^N F(x_i) \quad (2.3)$$

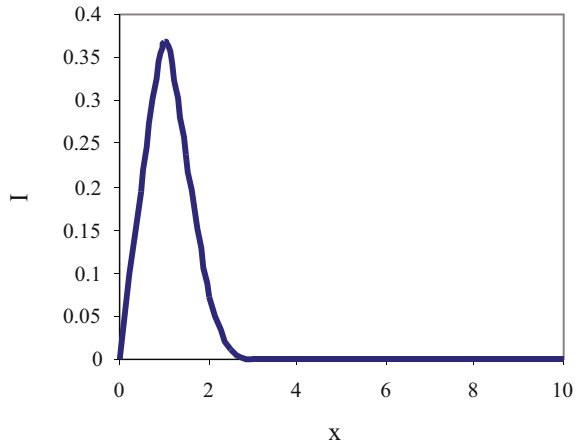
where x_i are random numbers generated from a uniform distribution and N corresponds to number of samples.

If random numbers are drawn from a uniform distribution, information is spread over the interval we are sampling over. However, if a nonuniform (biased) distribution $G(x)$ (which draws more samples from the areas which make a substantial contribution to the integral) is used, the approximation of the integral will be more accurate and the process will be more efficient. This is the basic idea behind importance sampling, where a weighting function is used to approximate the integral as follows.

$$I_{imp} = \frac{1}{n} \sum_{i=1}^n \frac{F(x_i)}{G(x_i)} \quad (2.4)$$

Importance sampling is crucial for sampling low-probability events. We will revisit importance sampling when we consider the reweighting scheme in the BONUS algorithm in Chap. 5. The most critical issue for the implementation of importance sampling is the choice of the biased distribution which emphasizes the important regions of the input variables. A simple example for the application of importance sampling for estimation of a simple integral is given below.

Fig. 2.6 The function behavior



Example 2.2 Integrate the following function using the Monte Carlo method and the method of importance sampling.

$$I = \int_0^{\infty} x^2 \exp(-x^2) dx \quad (2.5)$$

Solution This function is not possible to integrate analytically but its value is known to be $\sqrt{\pi}/4 = 0.44311328 \dots$. As can be observed from Fig. 2.6, the value of this function decreases rapidly when x is greater than about 3.5. Therefore, there are only a small number of input arguments x where the integral has an appreciable value. If we apply a Monte Carlo integration to estimate this integral, we can uniformly sample the domain of this integral by using a uniform distribution between 0 and 1000 (a large value) and evaluate the integral.

However, we know that this integral only has an appreciable value at a specific interval. Because of that, if we use a uniform sample, most of the points will be from areas that correspond to values where the integral has a very small value. Therefore, we can use a nonuniform distribution function instead, for sampling. If we choose a distribution like the lognormal distribution, the number of samples required to obtain an accurate estimation will be less. For example, let us consider a lognormal distribution with mean $\mu = 1$ and a standard deviation of $\sigma = 1.7$. This is shown in Fig. 2.7. We can see that if we use a lognormal distribution, we will be sampling more from the areas of importance that make a significant contribution to the integral. The estimation of this integral using a uniform sample and a lognormal sample is compared in Table 2.2. As we can see, the integral is accurately estimated using importance sampling after only 100 samples. However, it requires 10,000 samples with the crude Monte Carlo method where a uniform distribution is used.

Fig. 2.7 Lognormal distribution with a mean $\mu = 1$ and a standard deviation of $\sigma = 1.7$

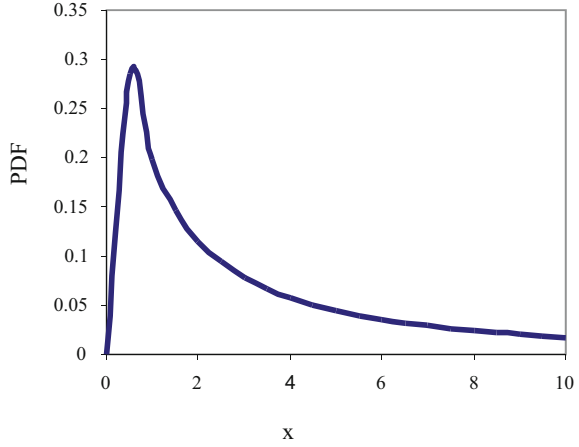


Table 2.2 The estimation of the integral by using uniform random sampling and importance sampling

N	Uniform random sampling	Importance sampling
10	0	0.11054
100	0.00095	0.44363
1000	0.07585	0.44312
10000	0.44131	0.44311

2.3.2 Stratified Sampling

Stratification is the grouping of the members of a population into equal or unequal probability areas (strata) before sampling. The strata must be mutually exclusive, which means that every element in the population must be assigned to only one stratum. Also, no population element is excluded. It is required that the proportion of each stratum in the sample should be the same as in the population.

Latin Hypercube Sampling (LHS) is one form of stratified sampling that can yield more precise estimates of the distribution function [32] and therefore reduce the number of samples required to improve computational efficiency. It is a full stratification of the sampled distribution with a random selection inside each stratum. In LHS, the range of each uncertain parameter X_i is subdivided into nonoverlapping intervals of equal probability. One value from each interval is selected at random with respect to the probability distribution in the interval. The n values thus obtained for X_1 are paired in a random manner (i.e., equally likely combinations) with n values of X_2 . These n values are then combined with n values of X_3 to form n -triplets, and so on, until n k -tuplets are formed. To clarify how intervals are formed, consider the simple example given below.

Example 2.3 Consider two uncertain variables X_1 and X_2 . X_1 has a normal distribution with a mean value of $\mu = 8$ and a standard deviation of $\sigma = 1$. X_2 has a uniform distribution between 5 and 10. Generate an LHS sample for $n = 5$.

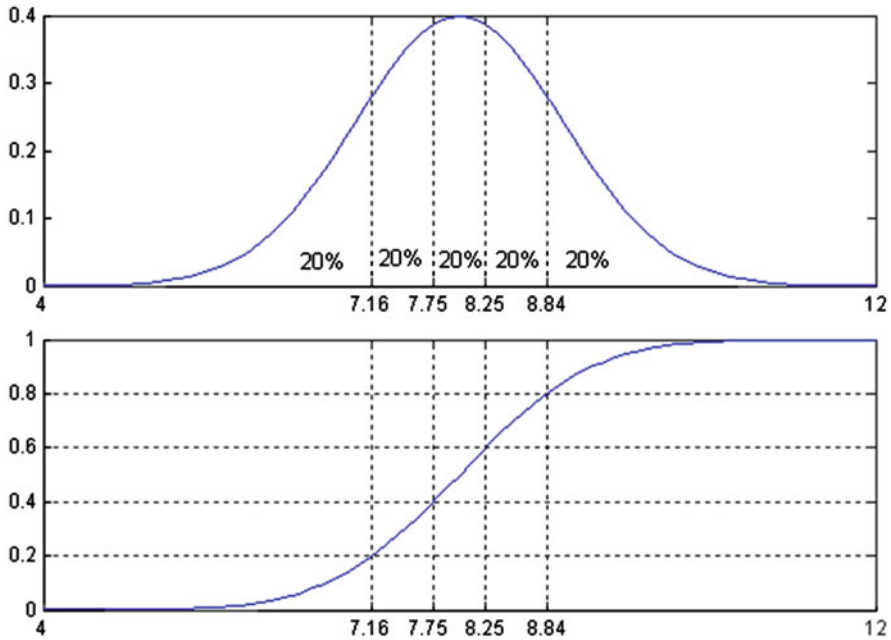


Fig. 2.8 Distribution and stratification for variable X_1

Solution Figure 2.8 shows the normal distribution PDF and CDF generated using the mean and standard deviation for X_1 and Fig. 2.9 shows the uniform distribution. For LHS, we divide each distribution into equal probability strata. Therefore, we have divided each distribution with five intervals with a 20 % probability each.

The next step to obtain a Latin hypercube sample is to choose specific values of X_1 and X_2 in each of their five respective intervals. This selection is done in a random manner with respect to density in each interval. Next the selected values of X_1 and X_2 are paired randomly to form the 2-dimensional input vectors of size 5. This pairing is done by a random permutation of the first 5 integers with each input variable. For example, we can consider two random permutations of the integers (1, 2, 3, 4, 5):

Permutation 1: (2, 5, 3, 1, 4) Permutation 2: (4, 3, 2, 5, 1)

We can use these as interval numbers for X_1 (Permutation 1) and X_2 (Permutation 2). In order to get the specific values of X_1 and X_2 , $n = 5$ random numbers are randomly selected from the standard uniform distribution. If we denote these values by U_m , where $m = 1, 2, 3, 4, 5$. Each random number U_m is scaled to obtain a cumulative probability P_m , so that each P_m lies within m -th interval:

$$P_m = \frac{U_m}{5} + \frac{m-1}{5} \quad (2.6)$$

In Tables 2.3 and 2.4, possible selections of Latin hypercube sample of size 5 for random variables X_1 and X_2 are presented respectively. Therefore if we apply the

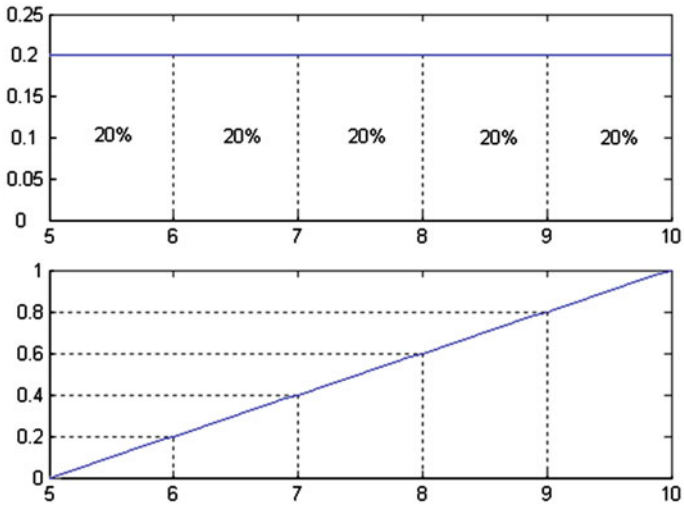


Fig. 2.9 Distribution and stratification for variable X_2

Table 2.3 Possible selection of values for a Latin hypercube sample of size 5 for the random variable X_1

Interval number (m)	Uniform (0,1) (U_m)	Scaled probabilities (P_m)	Corresponding sample
1	0.5832	0.1166	6.808
2	0.8125	0.3625	7.648
3	0.2980	0.4596	7.899
4	0.8470	0.7694	8.737
5	0.4369	0.8874	9.213

Table 2.4 Possible selection of values for a Latin hypercube sample of size 5 for the random variable X_2

Interval number (m)	Uniform (0,1) (U_m)	Scaled probabilities (P_m)	Corresponding sample
1	0.3370	0.0674	5.337
2	0.1678	0.2336	6.168
3	0.8419	0.5684	7.842
4	0.4372	0.6874	8.437
5	0.8127	0.9625	9.813

two permutations (Permutation 1 and 2) to choose the corresponding intervals for X_1 and X_2 , as given in Table 2.5, we can perform the pairing operation. In Fig. 2.10, this pairing process is illustrated.

LHS was designed to improve the uniformity properties of Monte Carlo methods, since it was shown that the error of approximating a distribution by finite samples

BONUS Algorithm for Large Scale Stochastic Nonlinear
Programming Problems

Diwekar, U.; David, A.

2015, XVIII, 146 p. 57 illus., 19 illus. in color., Softcover

ISBN: 978-1-4939-2281-9