Random Numbers

Introduction

The building block of a simulation study is the ability to generate random numbers, where a random number represents the value of a random variable uniformly distributed on (0, 1). In this chapter we explain how such numbers are computer generated and also begin to illustrate their uses.

3.1 Pseudorandom Number Generation

Whereas random numbers were originally either manually or mechanically generated, by using such techniques as spinning wheels, or dice rolling, or card shuffling, the modern approach is to use a computer to successively generate pseudorandom numbers. These pseudorandom numbers constitute a sequence of values, which, although they are deterministically generated, have all the appearances of being independent uniform (0, 1) random variables.

One of the most common approaches to generating pseudorandom numbers starts with an initial value x_0, called the seed, and then recursively computes successive values x_n, $n \geqslant 1$, by letting

$$x_n = ax_{n-1} \quad \text{modulo } m \tag{3.1}$$

where a and m are given positive integers, and where the above means that ax_{n-1} is divided by m and the remainder is taken as the value of x_n. Thus, each x_n is either $0, 1, \ldots, m-1$ and the quantity x_n/m—called a pseudorandom number—is taken as an approximation to the value of a uniform (0, 1) random variable.

Simulation. DOI: http://dx.doi.org/10.1016/B978-0-12-415825-2.00003-6

The approach specified by Equation (3.1) to generate random numbers is called the multiplicative congruential method. Since each of the numbers x_n assumes one of the values $0, 1, \ldots, m-1$, it follows that after some finite number (of at most m) of generated values a value must repeat itself; and once this happens the whole sequence will begin to repeat. Thus, we want to choose the constants a and m so that, for any initial seed x_0, the number of variables that can be generated before this repetition occurs is large.

In general the constants a and m should be chosen to satisfy three criteria:

1. For any initial seed, the resultant sequence has the "appearance" of being a sequence of independent uniform (0, 1) random variables.
2. For any initial seed, the number of variables that can be generated before repetition begins is large.
3. The values can be computed efficiently on a digital computer.

A guideline that appears to be of help in satisfying the above three conditions is that m should be chosen to be a large prime number that can be fitted to the computer word size. For a 32-bit word machine (where the first bit is a sign bit) it has been shown that the choices of $m = 2^{31} - 1$ and $a = 7^5 = 16,807$ result in desirable properties. (For a 36-bit word machine the choices of $m = 2^{35} - 31$ and $a = 5^5$ appear to work well.)

Another generator of pseudorandom numbers uses recursions of the type

$$x_n = (ax_{n-1} + c) \quad \text{modulo } m$$

Such generators are called mixed congruential generators (as they involve both an additive and a multiplicative term). When using generators of this type, one often chooses m to equal the computer's word length, since this makes the computation of $(ax_{n-1} + c)$ modulo m—that is, the division of $ax_{n-1} + c$ by m—quite efficient.

As our starting point in the computer simulation of systems we suppose that we can generate a sequence of pseudorandom numbers which can be taken as an approximation to the values of a sequence of independent uniform (0, 1) random variables. That is, we do not explore the interesting theoretical questions, which involve material outside the scope of this text, relating to the construction of "good" pseudorandom number generators. Rather, we assume that we have a "black box" that gives a random number on request.

3.2 Using Random Numbers to Evaluate Integrals

One of the earliest applications of random numbers was in the computation of integrals. Let $g(x)$ be a function and suppose we wanted to compute θ where

$$\theta = \int_0^1 g(x)\,dx$$

To compute the value of θ, note that if U is uniformly distributed over (0, 1), then we can express θ as

$$\theta = E\,[g(U)]$$

If $U_1, \ldots, U_k$ are independent uniform (0, 1) random variables, it thus follows that the random variables $g(U_1), \ldots, g(U_k)$ are independent and identically distributed random variables having mean θ. Therefore, by the strong law of large numbers, it follows that, with probability 1,

$$\sum_{i=1}^{k} \frac{g(U_i)}{k} \rightarrow E\,[g(U)] = \theta \quad \text{as } k \rightarrow \infty$$

Hence we can approximate θ by generating a large number of random numbers u_i and taking as our approximation the average value of $g(u_i)$. This approach to approximating integrals is called the *Monte Carlo* approach.

If we wanted to compute

$$\theta = \int_a^b g(x)\,dx$$

then, by making the substitution $y = (x-a)/(b-a),\ dy = dx/(b-a)$, we see that

$$\begin{aligned}\theta &= \int_0^1 g(a+[b-a]\,y)(b-a)\,dy \\ &= \int_0^1 h(y)\,dy\end{aligned}$$

where $h(y) = (b-a)g(a+[b-a]\,y)$. Thus, we can approximate θ by continually generating random numbers and then taking the average value of h evaluated at these random numbers.

Similarly, if we wanted

$$\theta = \int_0^\infty g(x)\,dx$$

we could apply the substitution $y = 1/(x+1), dy = -dx/(x+1)^2 = -y^2\,dx$, to obtain the identity

$$\theta = \int_0^1 h(y)\,dy$$

where

$$h(y) = \frac{g\left(\frac{1}{y}-1\right)}{y^2}$$

The utility of using random numbers to approximate integrals becomes more apparent in the case of multidimensional integrals. Suppose that g is a function with an n-dimensional argument and that we are interested in computing

$$\theta = \int_0^1 \int_0^1 \cdots \int_0^1 g(x_1, \ldots, x_n)\, dx_1\, dx_2 \cdots dx_n$$

The key to the Monte Carlo approach to estimate θ lies in the fact that θ can be expressed as the following expectation:

$$\theta = E\left[g(U_1, \ldots, U_n)\right]$$

where $U_1, \ldots, U_n$ are independent uniform (0, 1) random variables. Hence, if we generate k independent sets, each consisting of n independent uniform (0, 1) random variables

$$\begin{gathered} U_1^1, \ldots, U_n^1 \\ U_1^2, \ldots, U_n^2 \\ \vdots \\ U_1^k, \ldots, U_n^k \end{gathered}$$

then, since the random variables $g(U_1^i, \ldots, U_n^i)$, $i = 1, \ldots, k$, are all independent and identically distributed random variables with mean θ, we can estimate θ by $\sum_{i=1}^k g(U_1^i, \ldots, U_n^i)/k$.

For an application of the above, consider the following approach to estimating π.

Example 3a The Estimation of π Suppose that the random vector (X, Y) is uniformly distributed in the square of area 4 centered at the origin. That is, it is a random point in the region specified in Figure 3.1. Let us consider now the probability that this random point in the square is contained within the inscribed circle of radius 1 (see Figure 3.2). Note that since (X, Y) is uniformly distributed in the square it follows that

$$\begin{aligned} P\{(X, Y) \text{ is in the circle}\} &= P\{X^2 + Y^2 \leqslant 1\} \\ &= \frac{\text{Area of the circle}}{\text{Area of the square}} = \frac{\pi}{4} \end{aligned}$$

Hence, if we generate a large number of random points in the square, the proportion of points that fall within the circle will be approximately $\pi/4$. Now if X and Y were independent and both were uniformly distributed over $(-1, 1)$, their joint density would be

$$\begin{aligned} f(x, y) &= f(x)f(y) \\ &= \frac{1}{2} \cdot \frac{1}{2} \\ &= \frac{1}{4}, \qquad -1 \leqslant x \leqslant 1, \quad -1 \leqslant y \leqslant 1 \end{aligned}$$

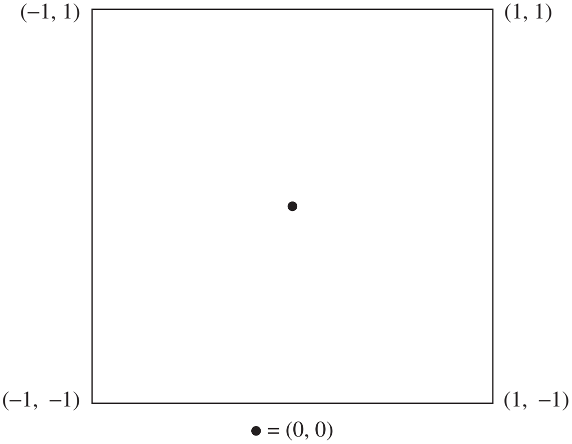

Figure 3.1. Square.

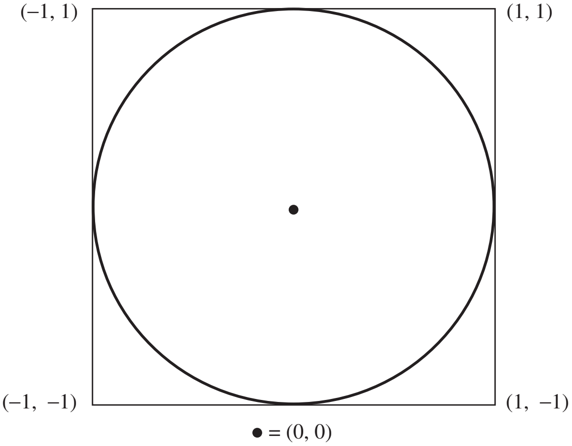

Figure 3.2. Circle within Square.

Since the density function of (X, Y) is constant in the square, it thus follows (by definition) that (X, Y) is uniformly distributed in the square. Now if U is uniform on (0, 1) then $2U$ is uniform on (0, 2), and so $2U - 1$ is uniform on $(-1, 1)$. Therefore, if we generate random numbers U_1 and U_2, set $X = 2U_1 - 1$ and $Y = 2U_2 - 1$, and define

$$I = \begin{cases} 1 & \text{if } X^2 + Y^2 \leqslant 1 \\ 0 & \text{otherwise} \end{cases}$$

then

$$E[I] = P\{X^2 + Y^2 \leqslant 1\} = \frac{\pi}{4}$$

Hence we can estimate $\pi/4$ by generating a large number of pairs of random numbers u_1, u_2 and estimating $\pi/4$ by the fraction of pairs for which $(2u_1 - 1)^2 + (2u_2 - 1)^2 \leqslant 1$. □

Thus, random number generators can be used to generate the values of uniform (0, 1) random variables. Starting with these random numbers we show in Chapters 4 and 5 how we can generate the values of random variables from arbitrary distributions. With this ability to generate arbitrary random variables we will be able to simulate a probability system—that is, we will be able to generate, according to the specified probability laws of the system, all the random quantities of this system as it evolves over time.

Exercises

1. If $x_0 = 5$ and

$$x_n = 3x_{n-1} \bmod 150$$

find $x_1, \ldots, x_{10}$.

2. If $x_0 = 3$ and

$$x_n = (5x_{n-1} + 7) \bmod 200$$

find $x_1, \ldots, x_{10}$.

In Exercises 3–9 use simulation to approximate the following integrals. Compare your estimate with the exact answer if known.

3. $\int_0^1 \exp\{e^x\}\,dx$

4. $\int_0^1 (1 - x^2)^{3/2}\,dx$

5. $\int_{-2}^2 e^{x+x^2}\,dx$

6. $\int_0^\infty x(1 + x^2)^{-2}\,dx$

7. $\int_{-\infty}^\infty e^{-x^2}\,dx$

8. $\int_0^1 \int_0^1 e^{(x+y)^2}\,dy\,dx$

9. $\int_0^\infty \int_0^x e^{-(x+y)}\,dy\,dx$

[*Hint:* Let $I_y(x) = \begin{cases} 1 \text{ if } y < x \\ 0 \text{ if } y \geqslant x \end{cases}$ and use this function to equate the integral to one in which both terms go from 0 to ∞.]

10. Use simulation to approximate $\mathrm{Cov}(U, e^U)$, where U is uniform on (0, 1). Compare your approximation with the exact answer.

11. Let U be uniform on (0, 1). Use simulation to approximate the following:

(a) $\text{Corr}\left(U, \sqrt{1-U^2}\right)$.
(b) $\text{Corr}\left(U^2, \sqrt{1-U^2}\right)$.

12. For uniform (0, 1) random variables $U_1, U_2, \ldots$ define

$$N = \text{Minimum}\left\{n: \sum_{i=1}^{n} U_i > 1\right\}$$

That is, N is equal to the number of random numbers that must be summed to exceed 1.

(a) Estimate $E[N]$ by generating 100 values of N.
(b) Estimate $E[N]$ by generating 1000 values of N.
(c) Estimate $E[N]$ by generating 10,000 values of N.
(d) What do you think is the value of $E[N]$?

13. Let $U_i, i \geqslant 1$, be random numbers. Define N by

$$N = \text{Maximum}\left\{n: \prod_{i=1}^{n} U_i \geqslant e^{-3}\right\}$$

where $\prod_{i=1}^{0} U_i \equiv 1$.

(a) Find $E[N]$ by simulation.
(b) Find $P\{N = i\}$, for $i = 0, 1, 2, 3, 4, 5, 6$, by simulation.

14. With $x_1 = 23$, $x_2 = 66$, and

$$x_n = 3x_{n-1} + 5x_{n-2} \quad \text{mod}(100), \quad n \geqslant 3$$

we will call the sequence $u_n = x_n/100, n \geqslant 1$, the *text's random number sequence*. Find its first 14 values.

Bibliography

Knuth, D., *The Art of Computer Programming*, Vol. 2, 2nd ed., *Seminumerical Algorithms*. Addison-Wesley, Reading, MA, 2000.

L'Ecuyer, P., "Random Numbers for Simulation," *Commun. Assoc. Comput. Mach.* **33**, 1990.

Marsaglia, G., “Random Numbers Fall Mainly in the Planes,” *Proc. Natl. Acad. Sci. USA* **61**, 25–28, 1962.

Marsaglia, G., “*The Structure of Linear Congruential Sequences*,” in *Applications of Number Theory to Numerical Analysis*, S. K. Zaremba, ed., Academic Press, London, 1972, pp. 249–255.

Naylor, T., *Computer Simulation Techniques*. Wiley, New York, 1966.

Ripley, B., *Stochastic Simulation*. Wiley, New York, 1986.

von Neumann, J., “Various Techniques Used in Connection with Random Digits, ‘Monte Carlo Method,’” *U.S. National Bureau of Standards Applied Mathematics Series*, No. 12, 36–38, 1951.