18. MONTE CARLO TECHNIQUES

Revised June 1994 with the help of T. Adye, O. Dahl, and H.-J. Trost.

Monte Carlo techniques are used to simulate on a computer random behavior which is too complex to be derived analytically. Most calculations are based upon pseudorandom numbers, a reproducible sequence of numbers generated on the open interval (0,1) in such a way that they satisfy various statistical tests for a uniform distribution, with independent numbers. (Caution: some commercial random number generators fill the closed interval [0,1]. The occurrence of 0 or 1 can sometimes cause problems for the algorithms below). No such numbers are truly uniform and independent. Many commercial random number generators sacrifice randomness in favor of speed. It is not rare that unforeseen correlations will introduce non-negligible errors in the results. A useful test for this is to recompute the same results with a different algorithm for the pseudorandom numbers. To improve the performance of an existing generator one may use the Bays-Durham algorithm [see Ref. 1 for discussion]: (a) Initialize by generating and storing N (*e.g.*, $N = 97$) random numbers in an array v, using the available generator. Generate a new random number u and save it. (b) On the next call, use this u as an address $j = 1+$ (integer part of Nu) to select v_j as the random number to be returned. Also save this v_j as u for the next call. Replace v_j in the array with a new random number using the available generator. On the next call, go to (b).

A second problem sometimes encountered in computations requiring long sequences of random numbers is that all pseudorandom number generators will eventually begin over and repeat the same sequence. One may choose algorithms which minimize the number used. One may also use two or three different generators in different parts of the program.

Monte Carlo simulations of complex processes break them down into a sequence of steps. At each step a particular outcome is chosen from a set of possibilities according to a certain p.d.f. To do this we must transform our uniform random numbers into random numbers sampled from different distributions on different ranges.

Two techniques are in wide use to do this. We will discuss only single variable cases; multiple variable cases use straightforward extensions of these techniques. We assume we are in possession of a random number u chosen from a uniform distribution on (0,1).

18.1. Inverse transform method

If the desired probability density function is $f(x)$ on the range $-\infty < x < \infty$, its cumulative distribution function (expressing the probability that $x \leq a$) is given by Eq. (16.1). If a is chosen with probability density $f(a)$, then the integrated probability up to point a, $F(a)$, is itself a random variable which will occur with uniform probability density on $[0, 1]$. If x can take on any value, and ignoring the endpoints, we can then find a unique x chosen from the p.d.f. $f(s)$ for a given u if we set

$$u = F(x) \ , \tag{18.1}$$

provided we can find an inverse of F, defined by

$$x = F^{-1}(u) \ . \tag{18.2}$$

This method is shown in Fig. 18.1a.

For a discrete distribution, $F(x)$ will have a discontinuous jump of size $f(x_k)$ at each allowed $x_k, k = 1, 2, \cdots$. Choose u from a uniform distribution on (0,1) as before. Find x_k such that

$$F(x_{k-1}) < u \leq F(x_k) \equiv \text{Prob}\ (x \leq x_k) = \sum_{i=1}^{k} f(x_i) \ ; \tag{18.3}$$

then x_k is the value we seek (note: $F(x_0) \equiv 0$). This algorithm is illustrated in Fig. 18.1b.

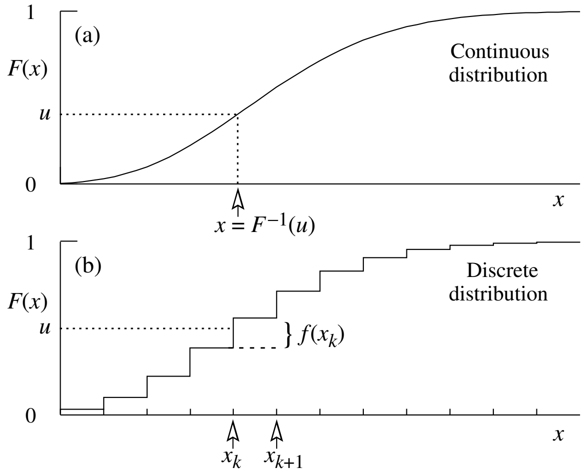

Figure 18.1: Use of a random number u chosen from a uniform distribution (0,1) to find a random number x from a distribution with cumulative distribution function $F(x)$.

18.2. Acceptance-rejection method (Von Neumann)

Very commonly an analytic form for $F(x)$ is unknown or too complex to work with, so that obtaining an inverse as in Eq. (18.2) is impractical. We suppose that for any given value of x the probability density function $f(x)$ can be computed and further that enough is known about $f(x)$ that we can enclose it entirely inside a shape which is C times an easily generated distribution $h(x)$ as illustrated in Fig. 18.2.

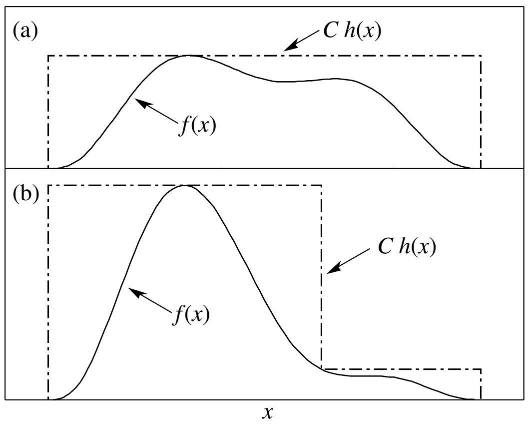

Figure 18.2: Illustration of the acceptance-rejection method. Random points are chosen inside the upper bounding figure, and rejected if the ordinate exceeds $f(x)$. Lower figure illustrates importance sampling.

Frequently $h(x)$ is uniform or is a normalized sum of uniform distributions. Note that both $f(x)$ and $h(x)$ must be normalized to unit area and therefore the proportionality constant $C > 1$. To generate $f(x)$, first generate a candidate x according to $h(x)$. Calculate $f(x)$ and the height of the envelope $C\,h(x)$; generate u and test if $uC\,h(x) \leq f(x)$. If so, accept x; if not reject x and try again. If we regard x and $uC\,h(x)$ as the abscissa and ordinate of a point in a two-dimensional plot, these points will populate the entire area $C\,h(x)$ in a smooth manner; then we accept those which fall under $f(x)$. The efficiency is the ratio of areas, which must equal $1/C$; therefore we must keep C as close as possible to 1.0. Therefore we try to choose $C\,h(x)$ to be as close to $f(x)$ as convenience dictates, as in the lower part of Fig. 18.2. This practice is called importance sampling, because