

# Chapter 1

## Monte Carlo simulation. Basic concepts

The name “Monte Carlo” was coined in the 1940s by scientists working on the nuclear weapon project in Los Alamos to designate a class of numerical methods based on the use of random numbers. Nowadays, Monte Carlo methods are widely used to solve complex physical and mathematical problems (James, 1980; Rubinstein, 1981; Kalos and Whitlock, 1986), particularly those involving multiple independent variables where more conventional numerical methods would demand formidable amounts of memory and computer time. The book by Kalos and Whitlock (1986) gives a readable survey of Monte Carlo techniques, including simple applications in radiation transport, statistical physics, and many-body quantum theory.

In Monte Carlo simulation of radiation transport, the history (track) of a particle is viewed as a random sequence of free flights that end with an interaction event where the particle changes its direction of movement, loses energy and, occasionally, produces secondary particles. The Monte Carlo simulation of a given experimental arrangement (e.g. an electron beam, coming from an accelerator and impinging on a water phantom) consists of the numerical generation of random histories. To simulate these histories we need an “interaction model”, i.e. a set of differential cross sections (DCS) for the relevant interaction mechanisms. The DCSs determine the probability distribution functions (PDF) of the random variables that characterize a track; 1) free path between successive interaction events, 2) kind of interaction taking place and 3) energy loss and angular deflection in a particular event (and initial state of emitted secondary particles, if any). Once these PDFs are known, random histories can be generated by using appropriate sampling methods. If the number of generated histories is large enough, quantitative information on the transport process may be obtained by simply averaging over the simulated histories.

The Monte Carlo method yields the same information as the solution of the Boltzmann transport equation, with the same interaction model, but is easier to implement (Berger, 1963). In particular, the simulation of radiation transport in finite samples is

straightforward, while even the simplest finite geometries (e.g. thin foils) are very difficult to be dealt with by the transport equation. The main drawback of the Monte Carlo method lies in its random nature, all the results are affected by statistical uncertainties, which can be reduced at the expense of increasing the sampled population and, hence, the computation time. Under special circumstances, the statistical uncertainties may be lowered by using variance-reduction techniques (Rubinstein, 1981; Bielajew and Rogers, 1988).

## 1.1 Elements of probability theory

The essential characteristic of Monte Carlo simulation is the use of random numbers and random variables. A random variable is a quantity that results from a repeatable process and whose actual values (realizations) cannot be predicted with certainty. In the real world, randomness originates either from uncontrolled factors (as occurs e.g. in games of chance) or from the quantum nature of microscopic systems and processes (e.g. nuclear disintegration and radiation interactions). As a familiar example, assume that we throw two dice in a box; the sum of points in their upper faces is a discrete random variable, which can take the values 2 to 12, while the distance  $x$  between the dice is a continuous random variable, which varies between zero (dice in contact) and a maximum value determined by the dimensions of the box. In the computer, random variables are generated by means of numerical transformations of random numbers (see below).

Let  $x$  be a continuous random variable that takes values in the interval  $x_{\min} \leq x \leq x_{\max}$ . To measure the likelihood of obtaining  $x$  in an interval  $(a, b)$  we use the probability  $P\{x|a < x < b\}$ , defined as the ratio  $n/N$  of the number  $n$  of values of  $x$  that fall within that interval and the total number  $N$  of generated  $x$ -values, in the limit  $N \rightarrow \infty$ . The probability of obtaining  $x$  in a differential interval of length  $dx$  about  $x_1$  can be expressed as

$$P\{x|x_1 < x < x_1 + dx\} = p(x_1) dx, \quad (1.1)$$

where  $p(x)$  is the probability distribution function (PDF) of  $x$ . Since 1) negative probabilities have no meaning and 2) the obtained value of  $x$  must be somewhere in  $(x_{\min}, x_{\max})$ , the PDF must be definite positive and normalized to unity

$$p(x) \geq 0 \quad \text{and} \quad \int_{x_{\min}}^{x_{\max}} p(x) dx = 1. \quad (1.2)$$

Any “function” that satisfies these two conditions can be interpreted as a PDF. In Monte Carlo simulation we shall frequently use the uniform distribution,

$$U_{x_{\min}, x_{\max}}(x) \equiv \begin{cases} 1/(x_{\max} - x_{\min}) & \text{if } x_{\min} < x < x_{\max}, \\ 0 & \text{otherwise,} \end{cases} \quad (1.3)$$

which is discontinuous. The definition (1.2) also includes singular distributions such as the Dirac delta,  $\delta(x - x_0)$ , which is defined by the property

$$\int_a^b f(x) \delta(x - x_0) dx = \begin{cases} f(x_0) & \text{if } a < x_0 < b, \\ 0 & \text{if } x_0 < a \text{ or } x_0 > b \end{cases} \quad (1.4)$$

for any function  $f(x)$  that is continuous at  $x_0$ . An equivalent, more intuitive definition is the following,

$$\delta(x - x_0) \equiv \lim_{\Delta \rightarrow 0} U_{x_0 - \Delta, x_0 + \Delta}(x), \quad (1.4')$$

which represents the delta distribution as the zero-width limit of a sequence of uniform distributions centred at the point  $x_0$ . Hence, the Dirac distribution describes a single-valued discrete random variable (i.e. a constant). The PDF of a random variable  $x$  that takes the discrete values  $x = x_1, x_2, \dots$  with point probabilities  $p_1, p_2, \dots$  can be expressed as a mixture of delta distributions,

$$p(x) = \sum_i p_i \delta(x - x_i). \quad (1.5)$$

Discrete distributions can thus be regarded as particular forms of continuous distributions.

Given a continuous random variable  $x$ , the cumulative distribution function of  $x$  is defined by

$$\mathcal{P}(x) \equiv \int_{x_{\min}}^x p(x') dx'. \quad (1.6)$$

This is a non-decreasing function of  $x$  that varies from  $\mathcal{P}(x_{\min}) = 0$  to  $\mathcal{P}(x_{\max}) = 1$ . In the case of a discrete PDF of the form (1.5),  $\mathcal{P}(x)$  is a step function. Notice that the probability  $P\{x|a < x < b\}$  of having  $x$  in the interval  $(a, b)$  is

$$P\{x|a < x < b\} = \int_a^b p(x) dx = \mathcal{P}(b) - \mathcal{P}(a), \quad (1.7)$$

and that  $p(x) = d\mathcal{P}(x)/dx$ .

The  $n$ -th moment of  $p(x)$  is defined as

$$\langle x^n \rangle = \int_{x_{\min}}^{x_{\max}} x^n p(x) dx. \quad (1.8)$$

The moment  $\langle x^0 \rangle$  is simply the integral of  $p(x)$ , which is equal to unity, by definition. However, higher order moments may or may not exist. An example of a PDF that has no even-order moments is the Lorentz or Cauchy distribution,

$$p_L(x) \equiv \frac{1}{\pi} \frac{\gamma}{\gamma^2 + x^2}, \quad -\infty < x < \infty. \quad (1.9)$$

Its first moment, and other odd-order moments, can be assigned a finite value if they are defined as the “principal value” of the integrals, e.g.

$$\langle x \rangle_L = \lim_{a \rightarrow \infty} \int_{-a}^{+a} x \frac{1}{\pi} \frac{\gamma}{\gamma^2 + x^2} dx = 0, \quad (1.10)$$

but the second and higher even-order moments are infinite, irrespective of the way they are defined.

The first moment, when it exists, is called the mean or expected value of the random variable  $x$ ,

$$\langle x \rangle = \int x p(x) dx. \quad (1.11)$$

The expected value of a function  $f(x)$  is defined in a similar way,

$$\langle f(x) \rangle \equiv \int f(x) p(x) dx. \quad (1.12)$$

Since  $f(x)$  is a random variable, it has its own PDF,  $\pi(f)$ , which is such that the probability of having  $f$  in a certain interval of length  $df$  is equal to the probability of having  $x$  in the corresponding interval or intervals<sup>1</sup>. Thus, if  $f(x)$  is a monotonously increasing function of  $x$  (so that there is a one-to-one correspondence between the values of  $x$  and  $f$ ),  $p(x) dx = \pi(f) df$  and

$$\pi(f) = p(x) (df/dx)^{-1}. \quad (1.13)$$

It can be shown that the definitions (1.11) and (1.12) are equivalent. If  $f(x)$  increases monotonously with  $x$ , the proof is trivial: we can start from the definition (1.11) and write

$$\langle f \rangle = \int f \pi(f) df = \int f(x) p(x) (dx/df) df = \int f(x) p(x) dx,$$

which agrees with (1.12). Notice that the expectation value is linear, i.e.

$$\langle a_1 f_1(x) + a_2 f_2(x) \rangle = a_1 \langle f_1(x) \rangle + a_2 \langle f_2(x) \rangle, \quad (1.14)$$

where  $a_1$  and  $a_2$  are arbitrary real constants.

If the first and second moments of the PDF  $p(x)$  exist, we define the variance of  $x$  [or of  $p(x)$ ] by

$$\text{var}(x) \equiv \langle (x - \langle x \rangle)^2 \rangle = \int (x - \langle x \rangle)^2 p(x) dx = \langle x^2 \rangle - \langle x \rangle^2. \quad (1.15)$$

The square root of the variance,  $\sigma \equiv [\text{var}(x)]^{1/2}$ , is called the “standard deviation” (and sometimes the “standard uncertainty”); it gives a measure of the dispersion of the random variable (i.e. of the width of the PDF). The Dirac delta is the only PDF that has zero variance. Similarly, the variance of a function  $f(x)$  is defined as

$$\text{var}\{f(x)\} = \langle f^2(x) \rangle - \langle f(x) \rangle^2. \quad (1.16)$$

Thus, for a constant  $f(x) = a$ ,  $\langle f \rangle = a$  and  $\text{var}\{f\} = 0$ .

---

<sup>1</sup>When  $f(x)$  does not increase or decrease monotonously with  $x$ , there may be multiple values of  $x$  corresponding to a given value of  $f$ .

### 1.1.1 Two-dimensional random variables

Let us now consider the case of a two-dimensional random variable,  $(x, y)$ . The corresponding (joint) PDF  $p(x, y)$  satisfies the conditions

$$p(x, y) \geq 0 \quad \text{and} \quad \int dx \int dy p(x, y) = 1. \quad (1.17)$$

The *marginal* PDFs of  $x$  and  $y$  are defined as

$$q(x) \equiv \int p(x, y) dy \quad \text{and} \quad q(y) \equiv \int p(x, y) dx, \quad (1.18)$$

i.e.  $q(x)$  is the probability of obtaining the value  $x$  and *any* value of  $y$ . The joint PDF can be expressed as

$$p(x, y) = q(x) p(y|x) = q(y) p(x|y), \quad (1.19)$$

where

$$p(x|y) = \frac{p(x, y)}{q(y)} \quad \text{and} \quad p(y|x) = \frac{p(x, y)}{q(x)} \quad (1.20)$$

are the *conditional* PDFs of  $x$  and  $y$ , respectively. Notice that  $p(x|y)$  is the normalized PDF of  $x$  for a fixed value of  $y$ .

The expectation value of a function  $f(x, y)$  is

$$\langle f(x, y) \rangle = \int dx \int dy f(x, y) p(x, y). \quad (1.21)$$

The moments of the PDF are defined by

$$\langle x^n y^m \rangle = \int dx \int dy x^n y^m p(x, y). \quad (1.22)$$

In particular,

$$\langle x^n \rangle = \int dx \int dy x^n p(x, y) = \int x^n q(x) dx. \quad (1.23)$$

Again, the only moment that is necessarily defined is  $\langle x^0 y^0 \rangle = 1$ . When the corresponding moments exist, the variances of  $x$  and  $y$  are given by

$$\text{var}(x) = \langle x^2 \rangle - \langle x \rangle^2 \quad \text{and} \quad \text{var}(y) = \langle y^2 \rangle - \langle y \rangle^2. \quad (1.24)$$

The variance of  $x + y$  is

$$\text{var}(x + y) = \langle (x + y)^2 \rangle - \langle x + y \rangle^2 = \text{var}(x) + \text{var}(y) + 2 \text{cov}(x, y), \quad (1.25)$$

where

$$\text{cov}(x, y) = \langle xy \rangle - \langle x \rangle \langle y \rangle \quad (1.26)$$

is the *covariance* of  $x$  and  $y$ , which can be positive or negative. A related quantity is the *correlation coefficient*,

$$\rho(x, y) = \frac{\text{cov}(x, y)}{\sqrt{\text{var}(x) \text{var}(y)}}, \quad (1.27)$$

which takes values from  $-1$  to  $1$ . Notice that  $\text{cov}(x, x) = \text{var}(x)$ . When the variables  $x$  and  $y$  are independent, i.e. when  $p(x, y) = p_x(x)p_y(y)$ , we have

$$\text{cov}(x, y) = 0 \quad \text{and} \quad \text{var}(x + y) = \text{var}(x) + \text{var}(y). \quad (1.28)$$

Moreover, for independent variables,

$$\text{var}\{a_1x + a_2y\} = a_1^2 \text{var}(x) + a_2^2 \text{var}(y). \quad (1.29)$$

## 1.2 Random sampling methods

The first component of a Monte Carlo calculation is the numerical sampling of random variables with specified PDFs. In this section we describe different techniques to generate random values of a variable  $x$  distributed in the interval  $(x_{\min}, x_{\max})$  according to a given PDF  $p(x)$ . We concentrate on the simple case of single-variable distributions, since random sampling from multivariate distributions can always be reduced to single-variable sampling (see below). A more detailed description of sampling methods can be found in the textbooks of Rubinstein (1981) and Kalos and Whitlock (1986).

### 1.2.1 Random number generator

In general, random sampling algorithms are based on the use of random numbers  $\xi$  uniformly distributed in the interval  $(0, 1)$ . These random numbers can be easily generated on the computer (see e.g. Kalos and Whitlock, 1986; James, 1990). Among the “good” random number generators currently available, the simplest ones are the so-called multiplicative congruential generators (Press and Teukolsky, 1992). A popular example of this kind of generator is the following,

$$R_n = 7^5 R_{n-1} \pmod{2^{31} - 1}, \quad \xi_n = R_n / (2^{31} - 1), \quad (1.30)$$

which produces a sequence of random numbers  $\xi_n$  uniformly distributed in  $(0, 1)$  from a given “seed”  $R_0$  ( $< 2^{31} - 1$ ). Actually, the generated sequence is not truly random, since it is obtained from a deterministic algorithm (the term “pseudo-random” would be more appropriate), but it is very unlikely that the subtle correlations between the values in the sequence have an appreciable effect on the simulation results. The generator (1.30) is known to have good random properties (Press and Teukolsky, 1992). However, the sequence is periodic, with a period of the order of  $10^9$ . With present-day computational facilities, this value is not large enough to prevent re-initiation in a single simulation run. An excellent critical review of random number generators has been published by James (1990), where he recommends using algorithms that are more sophisticated than simple congruential ones. The generator implemented in the FORTRAN77 function `RAND` (table 1.1) is due to L’Ecuyer (1988); it produces 32-bit floating point numbers uniformly distributed in the *open* interval between zero and one. Its period is of the order of  $10^{18}$ , which is virtually infinite for practical simulations.

**Table 1.1:** FORTRAN77 random number generator.

```

C *****
C                               FUNCTION RAND
C *****
C                               FUNCTION RAND(DUMMY)
C
C This is an adapted version of subroutine RANECU written by F. James
C (Comput. Phys. Commun. 60 (1990) 329-344), which has been modified to
C give a single random number at each call.
C
C The 'seeds' ISEED1 and ISEED2 must be initialized in the main program
C and transferred through the named common block /RSEED/.
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER*4 (I)
C      PARAMETER (USCALE=1.0D0/2.0D0**31)
C      COMMON/RSEED/ISEED1,ISEED2
C
C      I1=ISEED1/53668
C      ISEED1=40014*(ISEED1-I1*53668)-I1*12211
C      IF(ISEED1.LT.0) ISEED1=ISEED1+2147483563
C
C      I2=ISEED2/52774
C      ISEED2=40692*(ISEED2-I2*52774)-I2*3791
C      IF(ISEED2.LT.0) ISEED2=ISEED2+2147483399
C
C      IZ=ISEED1-ISEED2
C      IF(IZ.LT.1) IZ=IZ+2147483562
C      RAND=IZ*USCALE
C
C      RETURN
C      END

```

### 1.2.2 Inverse transform method

The cumulative distribution function of  $p(x)$ , eq. (1.6), is a non-decreasing function of  $x$  and, therefore, it has an inverse function  $\mathcal{P}^{-1}(\xi)$ . The transformation  $\xi = \mathcal{P}(x)$  defines a new random variable that takes values in the interval (0,1), see fig. 1.1. Owing to the correspondence between  $x$  and  $\xi$  values, the PDF of  $\xi$ ,  $p_\xi(\xi)$ , and that of  $x$ ,  $p(x)$ , are related by  $p_\xi(\xi) d\xi = p(x) dx$ . Hence,

$$p_\xi(\xi) = p(x) \left( \frac{d\xi}{dx} \right)^{-1} = p(x) \left( \frac{d\mathcal{P}(x)}{dx} \right)^{-1} = 1, \quad (1.31)$$

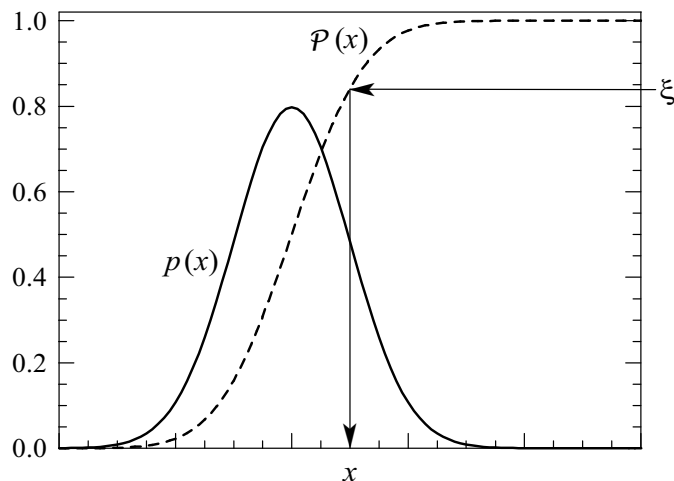
that is,  $\xi$  is distributed uniformly in the interval (0,1).

Now it is clear that if  $\xi$  is a random number, the variable  $x$  defined by  $x = \mathcal{P}^{-1}(\xi)$  is randomly distributed in the interval  $(x_{\min}, x_{\max})$  with PDF  $p(x)$  (see fig. 1.1). This provides a practical method of generating random values of  $x$  using a generator of random numbers uniformly distributed in (0,1). The randomness of  $x$  is guaranteed by

that of  $\xi$ . Notice that  $x$  is the (unique) root of the equation

$$\xi = \int_{x_{\min}}^x p(x') dx', \quad (1.32)$$

which will be referred to as the *sampling equation* of the variable  $x$ . This procedure for random sampling is known as the *inverse transform method*; it is particularly adequate for PDFs  $p(x)$  given by simple analytical expressions such that the sampling equation (1.32) can be solved analytically.



**Figure 1.1:** Random sampling from a distribution  $p(x)$  using the inverse transform method.

Consider, for instance, the uniform distribution in the interval  $(a, b)$ ,

$$p(x) \equiv U_{a,b}(x) = \frac{1}{b-a}.$$

The sampling equation (1.32) then reads

$$\xi = \frac{x-a}{b-a}, \quad (1.33)$$

which leads to the well-known sampling formula

$$x = a + \xi(b-a). \quad (1.34)$$

As another familiar example, consider the exponential distribution

$$p(s) = \frac{1}{\lambda} \exp(-s/\lambda), \quad s > 0, \quad (1.35)$$

of the free path  $s$  of a particle between interaction events (see section 1.4.1). The parameter  $\lambda$  represents the mean free path. In this case, the sampling equation (1.32) is easily solved to give the sampling formula

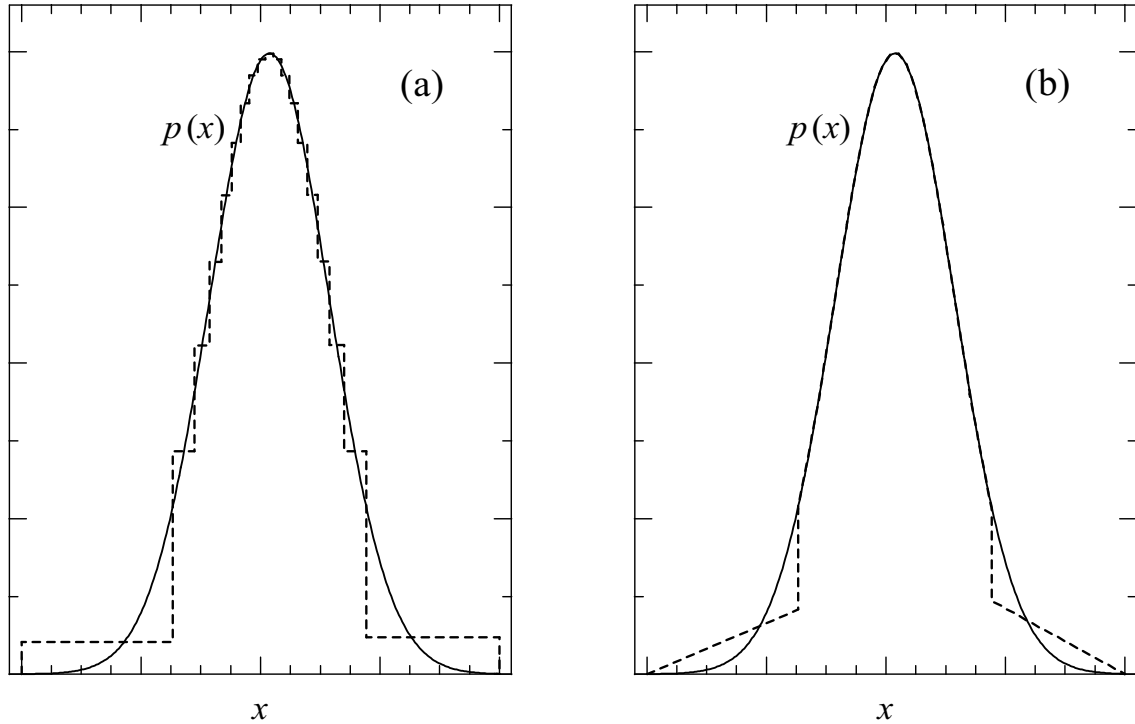
$$s = -\lambda \ln(1 - \xi) = -\lambda \ln \xi. \quad (1.36)$$

The last equality follows from the fact that  $1 - \xi$  is also a random number distributed in  $(0,1)$ .



### Numerical inverse transform

The inverse transform method can also be efficiently used for random sampling from continuous distributions  $p(x)$  that are given in numerical form, or that are too complicated to be sampled analytically. To apply this method, the cumulative distribution function  $\mathcal{P}(x)$  has to be evaluated at the points  $x_i$  of a certain grid. The sampling equation  $\mathcal{P}(x) = \xi$  can then be solved by inverse interpolation, i.e. by interpolating in the table  $(\xi_i, x_i)$ , where  $\xi_i \equiv \mathcal{P}(x_i)$  ( $\xi$  is regarded as the independent variable). Care must be exercised to make sure that the numerical integration and interpolation do not introduce significant errors.



**Figure 1.2:** Random sampling from a continuous distribution  $p(x)$  using the numerical inverse transform method with  $N = 20$  grid points. a) Piecewise constant approximation. b) Piecewise linear approximation.

A simple, general, approximate method for numerical sampling from continuous distributions is the following. The values  $x_n$  ( $n = 0, 1, \dots, N$ ) of  $x$  for which the cumulative distribution function has the values  $n/N$ ,

$$\mathcal{P}(x_n) = \int_{x_{\min}}^{x_n} p(x) dx = \frac{n}{N}, \quad (1.37)$$

are previously computed and stored in memory. Notice that the *exact* probability of having  $x$  in the interval  $(x_n, x_{n+1})$  is  $1/N$ . We can now sample  $x$  by linear interpolation:

we generate a random number  $\xi$  and consider the quantity  $y \equiv \xi N$ , which takes values in the interval  $(0, N)$ . We set  $n = [y]$ , where the symbol  $[y]$  denotes the integer part of  $y$  (i.e. the largest integer that is less than  $y$ ). The value of  $x$  is obtained as

$$x = x_n + (x_{n+1} - x_n)u, \quad u \equiv y - n \in (0, 1). \quad (1.38)$$

This is equivalent to approximating the PDF by a piecewise constant function (see fig. 1.2a). Since the spacing between the points  $x_n$  (at which the cumulative distribution function is specified) is roughly proportional to  $1/p(x_n)$ , the approximation is more accurate in regions where  $p(x)$  is large.

The algorithm can be improved by storing the values  $p(x_n)$  of the PDF at the points  $x_n$  in memory and approximating the PDF in the interval  $(x_n, x_{n+1})$  linearly,

$$p_{\text{la}}(x) \simeq C_n \left[ p(x_n) + \frac{p(x_{n+1}) - p(x_n)}{x_{n+1} - x_n} (x - x_n) \right], \quad (1.39)$$

with a normalization constant  $C_n$  such that the integral of  $p_{\text{la}}(x)$  over the interval  $(x_n, x_{n+1})$  equals  $1/N$ . In general, this piecewise linear approximation is not continuous. Of course,  $p_{\text{la}}(x)$  will differ from the exact PDF  $p(x)$  when the latter is not linear in the interval, but the differences are smaller than for the piecewise constant approximation with the same number  $N$  of grid points (see fig. 1.2). Again, the approximation is better where  $p(x)$  is larger. An exact algorithm for random sampling from the piecewise linear approximation (1.39) is the following,

- (i) Generate a random number  $\xi$  and set  $y = \xi N$ ,  $n = [y]$  and  $u = y - n$ .
- (ii) If  $p(x_n) \neq 0$ , set  $r = p(x_{n+1})/p(x_n)$  and

$$t = \begin{cases} \frac{(1 - u + r^2 u)^{1/2} - 1}{r - 1} & \text{if } r \neq 1, \\ u & \text{if } r = 1. \end{cases} \quad (1.40)$$

- (iii) If  $p(x_n) = 0$ , set  $t = u^{1/2}$ .
- (iv) Deliver  $x = x_n + (x_{n+1} - x_n)t$ .

### 1.2.3 Discrete distributions

The inverse transform method can also be applied to discrete distributions. Consider that the random variable  $x$  can take the discrete values  $x = 1, \dots, N$  with point probabilities  $p_1, \dots, p_N$ , respectively. The corresponding PDF can be expressed as

$$p(x) = \sum_{i=1}^N p_i \delta(x - i), \quad (1.41)$$

where  $\delta(x)$  is the Dirac distribution. Here  $p(x)$  is assumed to be defined in an interval  $(a, b)$  with  $a < 1$  and  $b > N$ . The corresponding cumulative distribution function is

$$\mathcal{P}(x) = \sum_{i=1}^{[x]} p_i, \quad (1.42)$$

where  $[x]$  stands for the integer part of  $x$ . Notice that  $\mathcal{P}(x) = 0$  when  $x < 1$ . Then, eq. (1.32) leads to the sampling formula

$$\begin{aligned} x &= 1 && \text{if } \xi \leq p_1 \\ &= 2 && \text{if } p_1 < \xi \leq p_1 + p_2 \\ &\vdots \\ &= j && \text{if } \sum_{i=1}^{j-1} p_i < \xi \leq \sum_{i=1}^j p_i \\ &\vdots \end{aligned} \tag{1.43}$$

We can define the quantities

$$P_1 = 0, \quad P_2 = p_1, \quad P_3 = p_1 + p_2, \quad \dots, \quad P_{N+1} = \sum_{i=1}^N p_i = 1. \tag{1.44}$$

To sample  $x$  we generate a random number  $\xi$  and set  $x$  equal to the index  $i$  such that

$$P_i < \xi \leq P_{i+1}. \tag{1.45}$$

If the number  $N$  of  $x$ -values is large, this sampling algorithm may be quite slow because of the large number of comparisons needed to determine the sampled value. The easiest method to reduce the number of comparisons is to use binary search instead of sequential search. The algorithm for binary search, for a given value of  $\xi$ , proceeds as follows:

- (i) Set  $i = 1$  and  $j = N + 1$ .
- (ii) Set  $k = [(i + j)/2]$ .
- (iii) If  $P_k < \xi$ , set  $i = k$ ; otherwise set  $j = k$ .
- (iv) If  $j - i > 1$ , go to step (ii).
- (v) Deliver  $i$ .

When  $2^n < N \leq 2^{n+1}$ ,  $i$  is obtained after  $n+1$  comparisons. This number of comparisons is evidently much less than the number required when using purely sequential search.

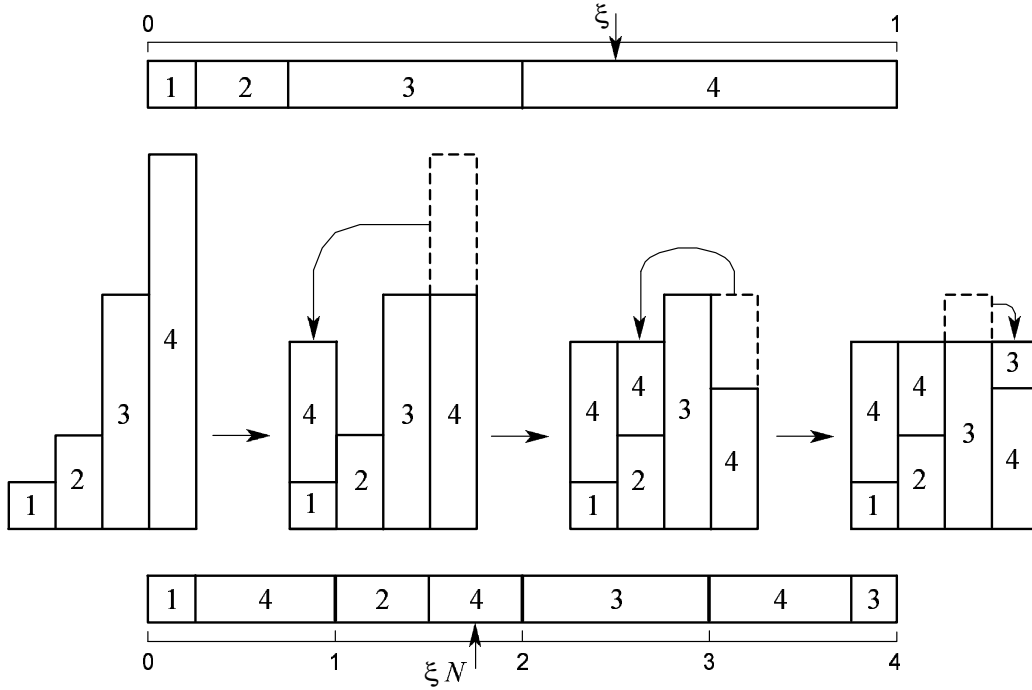
### Walker's aliasing method

Walker (1977) described an optimal sampling method for discrete distributions, which yields the sampled value with only one comparison. The idea underlying Walker's method can be easily understood by resorting to graphical arguments (Salvat, 1987). For this purpose, let us represent the PDF (1.41) as a histogram constructed with  $N$  bars of width  $1/N$  and heights  $Np_i$  (see fig. 1.3). Now, the histogram bars can be cut off at convenient heights and the resulting pieces can be arranged to fill up the square of unit side in such a way that each vertical line crosses, at most, two different pieces. This arrangement can be performed systematically by selecting the lowest and the highest bars in the histogram, say the  $\ell$ th and the  $j$ th, respectively, and by cutting the highest bar off to complete the lowest one, which subsequently is kept unaltered. In order to

keep track of the performed transformation, we label the added piece with the “alias” value  $K_\ell = j$ , giving its original position in the histogram, and introduce the “cutoff” value  $F_\ell$  defined as the height of the lower piece in the  $\ell$ th bar of the resulting square. This lower piece keeps the label  $\ell$ . Evidently, iteration of this process eventually leads to the complete square (after  $N - 1$  steps). Notice that the point probabilities  $p_i$  can be reconstructed from the alias and cutoff values. We have

$$Np_i = F_i + \sum_{j \neq i} (1 - F_j) \delta(i, K_j), \quad (1.46)$$

where  $\delta(i, j)$  denotes the Kronecker delta ( $= 1$  if  $i = j$  and  $= 0$  otherwise). Walker’s method for random sampling of  $x$  proceeds as follows: We sample two independent random numbers, say  $\xi_1$  and  $\xi_2$ , and define the random point  $(\xi_1, \xi_2)$ , which is uniformly distributed in the square. If  $(\xi_1, \xi_2)$  lies over a piece labelled with the index  $i$ , we take  $x = i$  as the selected value. Obviously, the probability of obtaining  $i$  as a result of the sampling equals the fractional area of the pieces labelled with  $i$ , which coincides with  $p_i$ .



**Figure 1.3:** Graphical representation of the inverse transform method (top) and Walker’s aliasing method (bottom) for random sampling from a discrete distribution. In this example, the random variable can take the values  $i = 1, 2, 3$  and  $4$  with relative probabilities  $1, 2, 5$  and  $8$ , respectively.

As formulated above, Walker’s algorithm requires the generation of two random numbers for each sampled value of  $x$ . With the aid of the following trick, the  $x$ -value

can be generated from a single random number. Continuing with our graphical picture, assume that the  $N$  bars in the square are aligned consecutively to form a segment of length  $N$  (bottom of fig. 1.3). To sample  $x$ , we can generate a single random value  $\xi N$ , which is uniformly distributed in  $(0, N)$  and determines one of the segment pieces. The result of the sampling is the label of the selected piece. Explicitly, the sampling algorithm proceeds as follows:

- (i) Generate a random number  $\xi$  and set  $R = \xi N + 1$ .
- (ii) Set  $i = [R]$  and  $r = R - i$ .
- (iii) If  $r > F_i$ , deliver  $x = K_i$ .
- (iv) Deliver  $x = i$ .

We see that the sampling of  $x$  involves only the generation of a random number and one comparison (irrespective of the number  $N$  of possible outcomes). The price we pay for this simplification reduces to doubling the number of memory locations that are needed: the two arrays  $K_i$  and  $F_i$  are used instead of the single array  $p_i$  (or  $P_i$ ). Unfortunately, the calculation of alias and cutoff values is fairly involved and this limits the applicability of Walker's algorithm to distributions that remain constant during the course of the simulation.

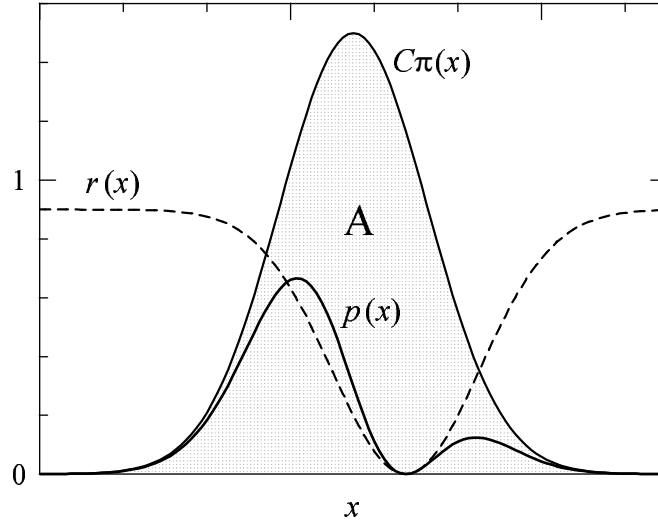
### 1.2.4 Rejection methods

The inverse transform method for random sampling is based on a one-to-one correspondence between  $x$  and  $\xi$  values, which is expressed in terms of a single-valued function. There is another kind of sampling method, due to von Neumann, that consists of sampling a random variable from a certain distribution [different from  $p(x)$ ] and subjecting it to a random test to determine whether it will be accepted for use or rejected. These rejection methods lead to very general techniques for sampling from any PDF.

The rejection algorithms can be understood in terms of simple graphical arguments (fig. 1.4). Consider that, by means of the inverse transform method or any other available sampling method, random values of  $x$  are generated from a PDF  $\pi(x)$ . For each sampled value of  $x$  we sample a random value  $y$  uniformly distributed in the interval  $(0, C\pi(x))$ , where  $C$  is a positive constant. Evidently, the points  $(x, y)$ , generated in this way, are uniformly distributed in the region A of the plane limited by the  $x$ -axis ( $y = 0$ ) and the curve  $y = C\pi(x)$ . Conversely, if (by some means) we generate random points  $(x, y)$  uniformly distributed in A, their  $x$ -coordinate is a random variable distributed according to  $\pi(x)$  (irrespective of the value of  $C$ ). Now, consider that the distribution  $\pi(x)$  is such that  $C\pi(x) \geq p(x)$  for some  $C > 0$  and that we generate random points  $(x, y)$  uniformly distributed in the region A as described above. If we reject the points with  $y > p(x)$ , the accepted ones (with  $y \leq p(x)$ ) are uniformly distributed in the region between the  $x$ -axis and the curve  $y = p(x)$  and hence, their  $x$ -coordinate is distributed according to  $p(x)$ .

A rejection method is thus completely specified by representing the PDF  $p(x)$  as

$$p(x) = C\pi(x)r(x), \quad (1.47)$$



**Figure 1.4:** Random sampling from a distribution  $p(x)$  using a rejection method.

where  $\pi(x)$  is a PDF that can be easily sampled e.g. by the inverse transform method,  $C$  is a positive constant and the function  $r(x)$  satisfies the conditions  $0 < r(x) \leq 1$ . The rejection algorithm for sampling from  $p(x)$  proceeds as follows:

- (i) Generate a random value  $x$  from  $\pi(x)$ .
- (ii) Generate a random number  $\xi$ .
- (iii) If  $\xi > r(x)$ , go to step (i).
- (iv) Deliver  $x$ .

From the geometrical arguments given above, it is clear that the algorithm does yield  $x$  values distributed according to  $p(x)$ . The following is a more formal proof: Step (i) produces  $x$ -values in the interval  $(x, x + dx)$  with probability  $\pi(x)dx$ , these values are accepted with probability  $r(x) = p(x)/[C\pi(x)]$  and, therefore, (apart from a normalization constant) the probability of delivering a value in  $(x, x + dx)$  is equal to  $p(x)dx$  as required. It is important to realize that, as regards Monte Carlo, the normalization of the simulated PDF is guaranteed by the mere fact that the algorithm delivers some value of  $x$ .

The efficiency of the algorithm, i.e. the probability of accepting a generated  $x$ -value, is

$$\epsilon = \int_a^b r(x)\pi(x) dx = \frac{1}{C}. \quad (1.48)$$

Graphically, the efficiency equals the ratio of the areas under the curves  $y = p(x)$  and  $y = C\pi(x)$ , which are 1 and  $C$ , respectively. For a given  $\pi(x)$ , since  $r(x) \leq 1$ , the constant  $C$  must satisfy the condition  $C\pi(x) \geq p(x)$  for all  $x$ . The minimum value of  $C$ , with the requirement that  $C\pi(x) = p(x)$  for some  $x$ , gives the optimum efficiency.

The PDF  $\pi(x)$  in eq. (1.47) should be selected in such a way that the resulting sampling algorithm is as fast as possible. In particular, random sampling from  $\pi(x)$

must be performed rapidly, by the inverse transform method or by the composition method (see below). High efficiency is also desirable, but not decisive. One hundred percent efficiency is obtained only with  $\pi(x) = p(x)$  (but random sampling from this PDF is just the problem we want to solve); any other PDF gives a lower efficiency. The usefulness of the rejection method lies in the fact that a certain loss of efficiency can be largely compensated with the ease of sampling  $x$  from  $\pi(x)$  instead of  $p(x)$ . A disadvantage of this method is that it requires the generation of several random numbers  $\xi$  to sample each  $x$ -value.

### 1.2.5 Two-dimensional variables. Composition methods

Let us consider a two-dimensional random variable  $(x, y)$  with joint probability distribution  $p(x, y)$ . Introducing the marginal PDF  $q(y)$  and the conditional PDF  $p(x|y)$  [see eqs. (1.18) and (1.20)],

$$q(y) \equiv \int p(x, y) dx, \quad p(x|y) = \frac{p(x, y)}{q(y)},$$

the two-variate distribution can be expressed as

$$p(x, y) = q(y) p(x|y). \quad (1.49)$$

It is now evident that to generate random points  $(x, y)$  from  $p(x, y)$  we can first sample  $y$  from  $q(y)$  and then  $x$  from  $p(x|y)$ . Hence, two-dimensional random variables can be generated by using single-variable sampling methods. This is also true for multivariate distributions, because an  $n$ -dimensional PDF can always be expressed as the product of a single-variable marginal distribution and an  $(n - 1)$ -dimensional conditional PDF.

From the definition of the marginal PDF of  $x$ ,

$$q(x) \equiv \int p(x, y) dy = \int q(y) p(x|y) dy, \quad (1.50)$$

it is clear that if we sample  $y$  from  $q(y)$  and, then,  $x$  from  $p(x|y)$ , the generated values of  $x$  are distributed according to  $q(x)$ . This idea is the basis of the *composition* methods, which are applicable when  $p(x)$ , the distribution to be simulated, is a probability mixture of several PDFs. More specifically, we consider that  $p(x)$  can be expressed as

$$p(x) = \int w(y) p_y(x) dy, \quad (1.51)$$

where  $w(y)$  is a continuous distribution and  $p_y(x)$  is a family of one-parameter PDFs, where  $y$  is the parameter identifying a unique distribution. Notice that if the parameter  $y$  takes only integer values  $y = i$  with point probabilities  $w_i$ , we would write

$$p(x) = \sum_i w_i p_i(x). \quad (1.52)$$

The composition method for random sampling from the PDF  $p(x)$  is as follows. First, a value of  $y$  (or  $i$ ) is drawn from the PDF  $w(y)$  and then  $x$  is sampled from the PDF  $p_y(x)$  for that chosen  $y$ .

This technique may be applied to generate random values from complex distributions obtained by combining simpler distributions that are themselves easily generated, by the inverse transform method or by rejection methods.

Devising fast, exact methods for random sampling from a given PDF is an interesting technical challenge. The ultimate criterion for the quality of a sampling algorithm is its speed in actual simulations: the best algorithm is the fastest. However, programming simplicity and elegance may justify the use of slower algorithms. For simple analytical distributions that have an analytical inverse cumulative distribution function, the inverse transform method is usually satisfactory. This is the case for a few elementary distributions (e.g. the uniform and exponential distributions considered above). The inverse transform method is also adequate for discrete distributions and for continuous PDFs given in numerical form. By combining the inverse transform, rejection and composition methods we can devise sampling algorithms for virtually any (single- or multivariate) PDF.

### Example 1. Sampling from the normal distribution

Frequently, we need to generate random values from the normal (or Gaussian) distribution

$$p_G(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2). \quad (1.53)$$

Since the cumulative distribution function cannot be inverted analytically, the inverse transform method is not appropriate. The easiest (but not the fastest) method to sample from the normal distribution consists of generating two independent random variables at a time, as follows. Let  $x_1$  and  $x_2$  be two independent normal variables. They determine a random point in the plane with PDF

$$p_{2G}(x_1, x_2) = p_G(x_1) p_G(x_2) = \frac{1}{2\pi} \exp[-(x_1^2 + x_2^2)/2].$$

Introducing the polar coordinates  $r$  and  $\phi$ ,

$$x_1 = r \cos \phi, \quad x_2 = r \sin \phi,$$

the PDF can be expressed as

$$p_{2G}(x_1, x_2) dx_1 dx_2 = \frac{1}{2\pi} \exp(-r^2/2) r dr d\phi = [\exp(-r^2/2) r dr] \left[ \frac{1}{2\pi} d\phi \right].$$

We see that  $r$  and  $\phi$  are independent random variables. The angle  $\phi$  is distributed uniformly on  $(0, 2\pi)$  and can be sampled as  $\phi = 2\pi\xi$ . The PDF of  $r$  is  $\exp(-r^2/2) r$  and



the corresponding cumulative distribution function is  $\mathcal{P}(r) = 1 - \exp(-r^2/2)$ . Therefore,  $r$  can be generated by the inverse transform method as

$$r = \sqrt{-2 \ln(1 - \xi)} = \sqrt{-2 \ln \xi}.$$

The two independent normal random variables are given by

$$\begin{aligned} x_1 &= \sqrt{-2 \ln \xi_1} \cos(2\pi \xi_2), \\ x_2 &= \sqrt{-2 \ln \xi_1} \sin(2\pi \xi_2), \end{aligned} \quad (1.54)$$

where  $\xi_1$  and  $\xi_2$  are two independent random numbers. This procedure is known as the Box-Müller method. It has the advantages of being exact and easy to program (it can be coded as a single FORTRAN statement).

The mean and variance of the normal variable are  $\langle x \rangle = 0$  and  $\text{var}(x) = 1$ . The linear transformation

$$X = m + \sigma x \quad (\sigma > 0) \quad (1.55)$$

defines a new random variable. From the properties (1.14) and (1.29), we have

$$\langle X \rangle = m \quad \text{and} \quad \text{var}(X) = \sigma^2. \quad (1.56)$$

The PDF of  $X$  is

$$p(X) = p_G(x) \frac{dx}{dX} = \frac{1}{\sigma\sqrt{2\pi}} \exp \left[ -\frac{(X - m)^2}{2\sigma^2} \right], \quad (1.57)$$

i.e.  $X$  is normally distributed with mean  $m$  and variance  $\sigma^2$ . Hence, to generate  $X$  we only have to sample  $x$  using the Box-Müller method and apply the transformation (1.55).

### Example 2. Uniform distribution on the unit sphere

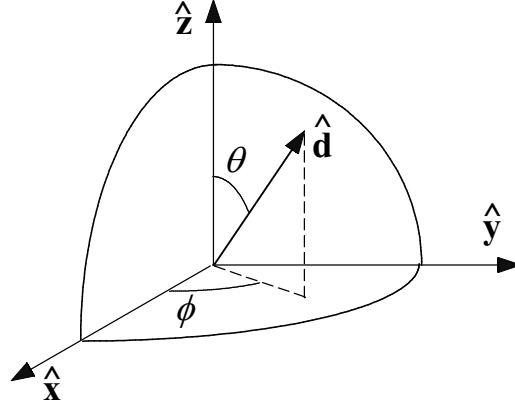
In radiation transport, the direction of motion of a particle is described by a unit vector  $\hat{\mathbf{d}}$ . Given a certain frame of reference, the direction  $\hat{\mathbf{d}}$  can be specified by giving either its direction cosines  $(u, v, w)$  (i.e. the projections of  $\hat{\mathbf{d}}$  on the directions of the coordinate axes) or the polar angle  $\theta$  and the azimuthal angle  $\phi$ , defined as in fig. 1.5,

$$\hat{\mathbf{d}} = (u, v, w) = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta). \quad (1.58)$$

Notice that  $\theta \in (0, \pi)$  and  $\phi \in (0, 2\pi)$ .

A direction vector can be regarded as a point on the surface of the unit sphere. Consider an isotropic source of particles, i.e. such that the initial direction  $(\theta, \phi)$  of emitted particles is a random point uniformly distributed on the surface of the sphere. The PDF is

$$p(\theta, \phi) d\theta d\phi = \frac{1}{4\pi} \sin \theta d\theta d\phi = \left[ \frac{\sin \theta}{2} d\theta \right] \left[ \frac{1}{2\pi} d\phi \right]. \quad (1.59)$$



**Figure 1.5:** Polar and azimuthal angles of a direction vector.

That is,  $\theta$  and  $\phi$  are independent random variables with PDFs  $p_\theta(\theta) = \sin \theta/2$  and  $p_\phi(\phi) = 1/(2\pi)$ , respectively. Therefore, the initial direction of a particle from an isotropic source can be generated by applying the inverse transform method to these PDFs,

$$\theta = \arccos(1 - 2\xi_1), \quad \phi = 2\pi\xi_2. \quad (1.60)$$

In some cases, it is convenient to replace the polar angle  $\theta$  by the variable

$$\mu = (1 - \cos \theta)/2, \quad (1.61)$$

which varies from 0 ( $\theta = 0$ ) to 1 ( $\theta = \pi$ ). In the case of an isotropic distribution, the PDF of  $\mu$  is

$$p_\mu(\mu) = p_\theta(\theta) \left( \frac{d\mu}{d\theta} \right)^{-1} = 1. \quad (1.62)$$

That is, a set of random points  $(\mu, \phi)$  uniformly distributed on the rectangle  $(0, 1) \times (0, 2\pi)$  corresponds to a set of random directions  $(\theta, \phi)$  uniformly distributed on the unit sphere.

### 1.3 Monte Carlo integration

As pointed out by James (1980), at least in a formal sense, all Monte Carlo calculations are equivalent to integrations. This equivalence permits a formal theoretical foundation for Monte Carlo techniques. An important aspect of simulation is the evaluation of the statistical uncertainties of the calculated quantities. We shall derive the basic formulae by considering the simplest Monte Carlo calculation, namely, the evaluation of a unidimensional integral. Evidently, the results are also valid for multidimensional integrals.

Consider the integral

$$I = \int_a^b F(x) dx, \quad (1.63)$$

which we recast in the form of an expectation value,

$$I = \int f(x) p(x) dx \equiv \langle f \rangle, \quad (1.64)$$

by introducing an arbitrary PDF  $p(x)$  and setting  $f(x) = F(x)/p(x)$  [it is assumed that  $p(x) > 0$  in  $(a, b)$  and  $p(x) = 0$  outside this interval]. The Monte Carlo evaluation of the integral  $I$  is very simple: generate a large number  $N$  of random points  $x_i$  from the PDF  $p(x)$  and accumulate the sum of values  $f(x_i)$  in a counter. At the end of the calculation the expected value of  $f$  is estimated as

$$\bar{f} \equiv \frac{1}{N} \sum_{i=1}^N f(x_i). \quad (1.65)$$

The law of large numbers says that, as  $N$  becomes very large,

$$\bar{f} \rightarrow I \quad (\text{in probability}). \quad (1.66)$$

In statistical terminology, this means that  $\bar{f}$ , the Monte Carlo result, is a *consistent estimator* of the integral (1.63). This is valid for any function  $f(x)$  that is finite and piecewise continuous, i.e. with a finite number of discontinuities.

The law of large numbers (1.66) can be restated as

$$\langle f \rangle = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N f(x_i). \quad (1.67)$$

By applying this law to the integral that defines the variance of  $f(x)$  [cf. eq. (1.16)]

$$\text{var}\{f(x)\} = \int f^2(x) p(x) dx - \langle f \rangle^2, \quad (1.68)$$

we obtain

$$\text{var}\{f(x)\} = \lim_{N \rightarrow \infty} \left\{ \frac{1}{N} \sum_{i=1}^N [f(x_i)]^2 - \left[ \frac{1}{N} \sum_{i=1}^N f(x_i) \right]^2 \right\}. \quad (1.69)$$

The expression in curly brackets is a consistent estimator of the variance of  $f(x)$ . It is advisable (see below) to accumulate the squared function values  $[f(x_i)]^2$  in a counter and, at the end of the simulation, estimate  $\text{var}\{f(x)\}$  according to eq. (1.69).

It is clear that different Monte Carlo runs [with different, independent sequences of  $N$  random numbers  $x_i$  from  $p(x)$ ] will yield different estimates  $\bar{f}$ . This implies that the outcome of our Monte Carlo code is affected by statistical uncertainties, similar to those found in laboratory experiments, which need to be properly evaluated to determine the “accuracy” of the Monte Carlo result. For this purpose, we may consider  $\bar{f}$  as a random

variable, the PDF of which is, in principle, unknown. Its mean and variance are given by

$$\langle \bar{f} \rangle = \left\langle \frac{1}{N} \sum_{i=1}^N f(x_i) \right\rangle = \frac{1}{N} \sum_{i=1}^N \langle f \rangle = \langle f \rangle \quad (1.70)$$

and

$$\text{var}(\bar{f}) = \text{var} \left[ \frac{1}{N} \sum_{i=1}^N f(x_i) \right] = \frac{1}{N^2} \sum_{i=1}^N \text{var}\{f(x)\} = \frac{1}{N} \text{var}\{f(x)\}, \quad (1.71)$$

where use has been made of properties of the expectation and variance operators. The standard deviation (or standard error) of  $\bar{f}$ ,

$$\sigma_f \equiv \sqrt{\text{var}(\bar{f})} = \sqrt{\frac{\text{var}\{f(x)\}}{N}}, \quad (1.72)$$

gives a measure of the statistical uncertainty of the Monte Carlo estimate  $\bar{f}$ . The result (1.72) has an important practical implication: in order to reduce the statistical uncertainty by a factor of 10, we have to increase the sample size  $N$  by a factor of 100. Evidently, this sets a limit to the accuracy that can be attained with the available computer power.

We can now invoke the central limit theorem (see e.g. James, 1980), which establishes that, in the limit  $N \rightarrow \infty$ , the PDF of  $\bar{f}$  is a normal (Gaussian) distribution with mean  $\langle f \rangle$  and standard deviation  $\sigma_f$ ,

$$p(\bar{f}) = \frac{1}{\sigma_f \sqrt{2\pi}} \exp \left( -\frac{(\bar{f} - \langle f \rangle)^2}{2\sigma_f^2} \right). \quad (1.73)$$

It follows that, for sufficiently large values of  $N$ , for which the theorem is applicable, the interval  $\bar{f} \pm n\sigma_f$  contains the exact value  $\langle f \rangle$  with a probability of 68.3% if  $n = 1$ , 95.4% if  $n = 2$  and 99.7% if  $n = 3$  ( $3\sigma$  rule).

The central limit theorem is a very powerful tool, since it predicts that the generated values of  $\bar{f}$  follow a specific distribution, but it applies only asymptotically. The minimum number  $N$  of sampled values needed to apply the theorem with confidence depends on the problem under consideration. If, in the case of our problem, the third central moment of  $f$ ,

$$\mu_3 \equiv \int [f(x) - \langle f \rangle]^3 p(x) dx, \quad (1.74)$$

exists, the theorem is essentially satisfied when

$$|\mu_3| \ll \sigma_f^3 \sqrt{N}. \quad (1.75)$$

In general, it is advisable to study the distribution of the estimator to ascertain the applicability of the central limit theorem. In most Monte Carlo calculations, however, statistical errors are estimated by simply assuming that the theorem is satisfied, irrespective of the sample size. We shall adopt this practice and report Monte Carlo results in the form  $\bar{f} \pm 3\sigma_f$ . In simulations of radiation transport, this is empirically validated

by the fact that simulated continuous distributions do “look” continuous (i.e. the “error bars” define a smooth band).

Each possible  $p(x)$  defines a Monte Carlo algorithm to calculate the integral  $I$ , eq. (1.63). The simplest algorithm (crude Monte Carlo) is obtained by using the uniform distribution  $p(x) = 1/(b-a)$ . Evidently,  $p(x)$  determines not only the density of sampled points  $x_i$ , but also the magnitude of the variance  $\text{var}\{f(x)\}$ , eq. (1.68),

$$\text{var}\{f(x)\} = \int_a^b p(x) \left[ \frac{F(x)}{p(x)} \right]^2 dx - I^2 = \int_a^b F(x) \left[ \frac{F(x)}{p(x)} - I \right] dx. \quad (1.76)$$

As a measure of the effectiveness of a Monte Carlo algorithm, it is common to use the efficiency  $\epsilon$ , which is defined by

$$\epsilon = 1/[\sigma_f^2 T], \quad (1.77)$$

where  $T$  is the computing time (or any other measure of the calculation effort) needed to get the simulation result. Since  $\sigma_f^2$  and  $T$  are roughly proportional to  $N^{-1}$  and  $N$ , respectively,  $\epsilon$  is a constant (i.e. it is independent of  $N$ ), on average.

The so-called variance-reduction methods are techniques that aim to optimize the *efficiency* of the simulation through an adequate choice of the PDF  $p(x)$ . Improving the efficiency of the algorithms is an important, and delicate, part of the art of Monte Carlo simulation. The interested reader is addressed to the specialized bibliography (e.g. Rubinstein, 1981). Although of common use, the term “variance reduction” is somewhat misleading, since a reduction in variance does not necessarily lead to improved efficiency. To make this clear, consider that a Monte Carlo algorithm, based on a certain PDF  $p(x)$ , has a variance that is less than that of crude Monte Carlo (i.e. with the uniform distribution); if the generation of  $x$ -values from  $p(x)$  takes a longer time than for the uniform distribution, the “variance-reduced” algorithm may be less efficient than crude Monte Carlo. Hence, one should avoid using PDFs that are too difficult to sample.

## 1.4 Simulation of radiation transport

In this section, we describe the essentials of Monte Carlo simulation of radiation transport. For the sake of simplicity, we limit our considerations to the detailed simulation method, where all the interaction events experienced by a particle are simulated in chronological succession, and we disregard the production of secondary particles, so that only one kind of particle is transported.

### 1.4.1 Scattering model and probability distribution functions

Consider a particle with energy  $E$  (kinetic energy, in the case of electrons and positrons) moving in a given medium. We limit our considerations to homogeneous “random scattering” media, such as gases, liquids and amorphous solids, where the “molecules”

are distributed at random with uniform density. The composition of the medium is specified by its stoichiometric formula, i.e. atomic number  $Z_i$  and number of atoms per molecule  $n_i$  of all the elements present. The stoichiometric indices  $n_i$  need not have integer values. In the case of alloys, for instance, they may be set equal to the percentage in number of each element and then a “molecule” is a group of 100 atoms with the appropriate proportion of each element. The “molecular weight” is  $A_M = \sum n_i A_i$ , where  $A_i$  is the atomic weight of the  $i$ -th element. The number of molecules per unit volume is given by

$$\mathcal{N} = N_A \frac{\rho}{A_M}, \quad (1.78)$$

where  $N_A$  is Avogadro’s number and  $\rho$  is the mass density of the material.

In each interaction, the particle may lose energy  $W$  and/or change its direction of movement. The angular deflection is determined by the polar scattering angle  $\theta$ , i.e. the angle between the directions of the particle before and after the interaction, and the azimuthal angle  $\phi$ . Let us assume that the particle can interact with the medium through two independent mechanisms, denoted as “A” and “B” (for instance, elastic and inelastic scattering, in the case of low-energy electrons). The scattering model is completely specified by the molecular differential cross sections (DCS)

$$\frac{d^2\sigma_A}{dWd\Omega}(E; W, \theta) \quad \text{and} \quad \frac{d^2\sigma_B}{dWd\Omega}(E; W, \theta), \quad (1.79)$$

where  $d\Omega$  is a solid angle element in the direction  $(\theta, \phi)$ . We have made the parametric dependence of the DCSs on the particle energy  $E$  explicit. Considering that the molecules in the medium are oriented at random, the DCS is independent of the azimuthal scattering angle, i.e. the angular distribution of scattered particles is axially symmetrical around the direction of incidence. The total cross sections (per molecule) are

$$\sigma_{A,B}(E) = \int_0^E dW \int_0^\pi 2\pi \sin \theta d\theta \frac{d^2\sigma_{A,B}}{dWd\Omega}(E; W, \theta). \quad (1.80)$$

The PDFs of the energy loss and the polar scattering angle in individual scattering events are

$$p_{A,B}(E; W, \theta) = \frac{2\pi \sin \theta}{\sigma_{A,B}(E)} \frac{d^2\sigma_{A,B}}{dWd\Omega}(E; W, \theta). \quad (1.81)$$

Notice that  $p_A(E; W, \theta)dWd\theta$  gives the (normalized) probability that, in a scattering event of type A, the particle loses energy in the interval  $(W, W + dW)$  and is deflected into directions with polar angle (relative to the initial direction) in the interval  $(\theta, \theta + d\theta)$ . The azimuthal scattering angle in each collision is uniformly distributed in the interval  $(0, 2\pi)$ , i.e.

$$p(\phi) = \frac{1}{2\pi}. \quad (1.82)$$

The total interaction cross section is

$$\sigma_T(E) = \sigma_A(E) + \sigma_B(E). \quad (1.83)$$

When the particle interacts with the medium, the kind of interaction that occurs is a discrete random variable, that takes the values “A” and “B” with probabilities

$$p_A = \sigma_A / \sigma_T \quad \text{and} \quad p_B = \sigma_B / \sigma_T. \quad (1.84)$$

It is worth recalling that this kind of single scattering model is only valid when diffraction effects resulting from coherent scattering from several centres (e.g. Bragg diffraction, channelling of charged particles) are negligible. This means that the simulation is applicable only to amorphous media and, with some care, to polycrystalline solids.

To get an intuitive picture of the scattering process, we can imagine each molecule as a sphere of radius  $r_s$  such that the cross-sectional area  $\pi r_s^2$  equals the total cross section  $\sigma_T$ . Now, assume that a particle impinges normally on a very thin material foil of thickness  $ds$ . What the particle sees in front of it is a uniform distribution of  $\mathcal{N} ds$  spheres per unit surface. An interaction takes place when the particle strikes one of these spheres. Therefore, the probability of interaction within the foil equals the fractional area covered by the spheres,  $\mathcal{N} \sigma_T ds$ . In other words,  $\mathcal{N} \sigma_T$  is the interaction probability per unit path length. Its inverse,

$$\lambda_T \equiv (\mathcal{N} \sigma_T)^{-1}, \quad (1.85)$$

is the (total) mean free path between interactions.

Let us now consider a particle that moves within an unbound medium. The PDF  $p(s)$  of the path length  $s$  of the particle from its current position to the site of the next interaction may be obtained as follows. The probability that the particle travels a path length  $s$  without interacting is

$$\mathcal{F}(s) = \int_s^\infty p(s') ds'. \quad (1.86)$$

The probability  $p(s) ds$  of having the next interaction when the travelled length is in the interval  $(s, s + ds)$  equals the product of  $\mathcal{F}(s)$  (the probability of arrival at  $s$  without interacting) and  $\lambda_T^{-1} ds$  (the probability of interacting within  $ds$ ). It then follows that

$$p(s) = \lambda_T^{-1} \int_s^\infty p(s') ds'. \quad (1.87)$$

The solution of this integral equation, with the boundary condition  $p(\infty) = 0$ , is the familiar exponential distribution

$$p(s) = \lambda_T^{-1} \exp(-s/\lambda_T). \quad (1.88)$$

Notice that the mean free path  $\lambda_T$  coincides with the average path length between collisions:

$$\langle s \rangle = \int_0^\infty s p(s) ds = \lambda_T. \quad (1.89)$$

The differential inverse mean free path for the interaction process A is defined as

$$\frac{d^2 \lambda_A^{-1}}{dW d\Omega}(E; W, \theta) = \mathcal{N} \frac{d^2 \sigma_A}{dW d\Omega}(E; W, \theta). \quad (1.90)$$

Evidently, the integral of the differential inverse mean free path gives the inverse mean free path for the process,

$$\lambda_A^{-1} = \int dW \int 2\pi \sin \theta d\theta \frac{d^2 \lambda_A^{-1}}{dW d\Omega}(E; W, \theta) = \mathcal{N} \sigma_A. \quad (1.91)$$

In the literature, the product  $\mathcal{N} \sigma_A$  is frequently called the *macroscopic cross section*, although this name is not appropriate for a quantity that has the dimensions of inverse length. Notice that the total inverse mean free path is the sum of the inverse mean free paths of the different active interaction mechanisms,

$$\lambda_T^{-1} = \lambda_A^{-1} + \lambda_B^{-1}. \quad (1.92)$$

### 1.4.2 Generation of random tracks

Each particle track starts off at a given position, with initial direction and energy in accordance with the characteristics of the source. The “state” of a particle immediately after an interaction (or after entering the sample or starting its trajectory) is defined by its position coordinates  $\mathbf{r} = (x, y, z)$ , energy  $E$  and direction cosines of the direction of flight, i.e. the components of the unit vector  $\hat{\mathbf{d}} = (u, v, w)$ , as seen from the laboratory reference frame. Each simulated track is thus characterized by a series of states  $\mathbf{r}_n, E_n, \hat{\mathbf{d}}_n$ , where  $\mathbf{r}_n$  is the position of the  $n$ -th scattering event and  $E_n$  and  $\hat{\mathbf{d}}_n$  are the energy and direction cosines of the direction of movement just *after* that event.

The generation of random tracks proceeds as follows. Let us assume that a track has already been simulated up to a state  $\mathbf{r}_n, E_n, \hat{\mathbf{d}}_n$ . The length  $s$  of the free path to the next collision, the involved scattering mechanism, the change of direction and the energy loss in this collision are random variables that are sampled from the corresponding PDFs, using the methods described in section 1.2. Hereafter,  $\xi$  stands for a random number uniformly distributed in the interval (0,1).

The length of the free flight is distributed according to the PDF given by eq. (1.88). Random values of  $s$  are generated by using the sampling formula [see eq. (1.36)]

$$s = -\lambda_T \ln \xi. \quad (1.93)$$

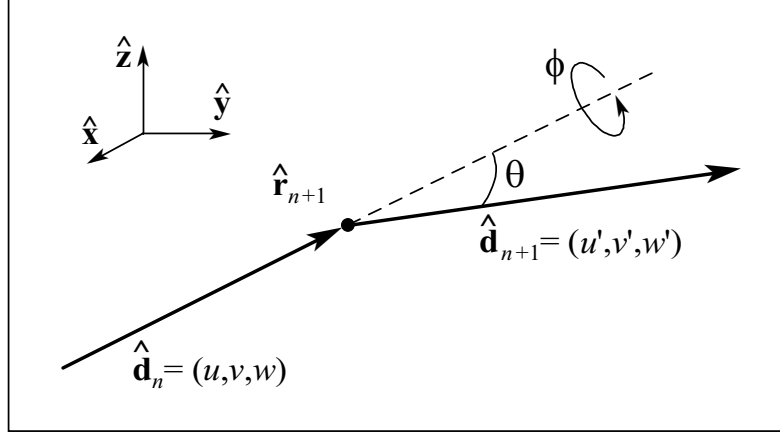
The following interaction occurs at the position

$$\mathbf{r}_{n+1} = \mathbf{r}_n + s \hat{\mathbf{d}}_n. \quad (1.94)$$

The type of this interaction (“A” or “B”) is selected from the point probabilities given by eq. (1.84) using the inverse transform method (section 1.2.2). The energy loss  $W$  and the polar scattering angle  $\theta$  are sampled from the distribution  $p_{A,B}(E; W, \theta)$ , eq. (1.81), by using a suitable sampling technique. The azimuthal scattering angle is generated, according to the uniform distribution in  $(0, 2\pi)$ , as  $\phi = 2\pi\xi$ .

After sampling the values of  $W$ ,  $\theta$  and  $\phi$ , the energy of the particle is reduced,  $E_{n+1} = E_n - W$ , and the direction of movement after the interaction  $\hat{\mathbf{d}}_{n+1} = (u', v', w')$





**Figure 1.6:** Angular deflections in single-scattering events.

is obtained by performing a rotation of  $\hat{\mathbf{d}}_n = (u, v, w)$  (see fig. 1.6). The rotation matrix  $R(\theta, \phi)$  is determined by the polar and azimuthal scattering angles. To explicitly obtain the direction vector  $\hat{\mathbf{d}}_{n+1} = R(\theta, \phi)\hat{\mathbf{d}}_n$  after the interaction, we first note that, if the initial direction is along the  $z$ -axis, the direction after the collision is

$$\begin{pmatrix} \sin \theta \cos \phi \\ \sin \theta \sin \phi \\ \cos \theta \end{pmatrix} = R_z(\phi)R_y(\theta)\hat{\mathbf{z}}, \quad (1.95)$$

where  $\hat{\mathbf{z}} = (0, 0, 1)$  and

$$R_y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \quad \text{and} \quad R_z(\phi) = \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1.96)$$

are rotation matrices corresponding to active rotations of angles  $\theta$  and  $\phi$  about the  $y$ - and  $z$ -axes, respectively. On the other hand, if  $\vartheta$  and  $\varphi$  are the polar and azimuthal angles of the initial direction

$$\hat{\mathbf{d}}_n = (\sin \vartheta \cos \varphi, \sin \vartheta \sin \varphi, \cos \vartheta), \quad (1.97)$$

the rotation  $R_y(-\vartheta)R_z(-\varphi)$  transforms the vector  $\hat{\mathbf{d}}_n$  into  $\hat{\mathbf{z}}$ . It is then clear that the final direction vector  $\hat{\mathbf{d}}_{n+1}$  can be obtained by performing the following sequence of rotations of the initial direction vector: 1)  $R_y(-\vartheta)R_z(-\varphi)$ , which transforms  $\hat{\mathbf{d}}_n$  into  $\hat{\mathbf{z}}$ ; 2)  $R_z(\phi)R_y(\theta)$ , which rotates  $\hat{\mathbf{z}}$  according to the sampled polar and azimuthal scattering angles; and 3)  $R_z(\varphi)R_y(\vartheta)$ , which inverts the rotation of the first step. Hence

$$R(\theta, \phi) = R_z(\varphi)R_y(\vartheta)R_z(\phi)R_y(\theta)R_y(-\vartheta)R_z(-\varphi). \quad (1.98)$$

The final direction vector is

$$\hat{\mathbf{d}}_{n+1} = R(\theta, \phi) \hat{\mathbf{d}}_n = R_z(\varphi) R_y(\vartheta) \begin{pmatrix} \sin \theta \cos \phi \\ \sin \theta \sin \phi \\ \cos \theta \end{pmatrix} \quad (1.99)$$

and its direction cosines are

$$\begin{aligned} u' &= u \cos \theta + \frac{\sin \theta}{\sqrt{1-w^2}} [uw \cos \phi - v \sin \phi], \\ v' &= v \cos \theta + \frac{\sin \theta}{\sqrt{1-w^2}} [vw \cos \phi + u \sin \phi], \\ w' &= w \cos \theta - \sqrt{1-w^2} \sin \theta \cos \phi. \end{aligned} \quad (1.100)$$

These equations are indeterminate when  $w \simeq \pm 1$ , i.e. when the initial direction is nearly parallel or antiparallel to the  $z$ -axis; in this case we can simply set

$$u = \pm \sin \theta \cos \phi, \quad v = \pm \sin \theta \sin \phi, \quad w = \pm \cos \theta. \quad (1.101)$$

Moreover, eqs. (1.100) are not very stable numerically and the normalization of  $\hat{\mathbf{d}}_{n+1}$  tends to drift from 1 after repeated usage. This must be remedied by periodically renormalizing  $\hat{\mathbf{d}}_{n+1}$ . The change of direction expressed by eqs. (1.100) and (1.101) is performed by the subroutine **DIRECT** (see the **PENELOPE** source listing).

The simulation of the track then proceeds by repeating these steps. A track is finished either when it leaves the material system or when the energy becomes smaller than a given energy  $E_{\text{abs}}$ , which is the energy where particles are assumed to be effectively stopped and absorbed in the medium.

### 1.4.3 Particle transport as a Markov process

The foregoing concepts, definitions and simulation scheme rest on the assumption that particle transport can be modelled as a Markov process<sup>2</sup>, i.e. “future values of a random variable (interaction event) are statistically determined by present events and depend only on the event immediately preceeding”. Owing to the Markovian character of the transport, we can stop the generation of a particle history at an arbitrary state (any point of the track) and resume the simulation from this state without introducing any bias in the results.

In mixed simulations of electron/positron transport, it is necessary to limit the length  $s$  of each “free jump” so that it does not exceed a given value  $s_{\text{max}}$ . To accomplish this, we still sample the free path length  $s$  to the next interaction from the exponential PDF

---

<sup>2</sup>The quoted definition is from the Webster’s Encyclopedic Unabridged Dictionary of the English Language (Portland House, New York, 1989).

(1.88), but when  $s > s_{\max}$  we only let the particle advance a distance  $s_{\max}$  along the direction of motion. At the end of the truncated free jump we do nothing (i.e. the particle keeps its energy and direction of motion unaltered); however, for programming convenience, we shall say that the particle suffers a *delta interaction* (actually, a “non-interaction”). When the sampled value of  $s$  is less than  $s_{\max}$ , a real interaction is simulated. After the interaction (either real or delta), we sample a new free path  $s$ , move the particle a distance  $s' = \min(s, s_{\max})$ , etc. From the Markovian character of the transport, it is clear that the insertion of delta interactions keeps the simulation unbiased. If you do not see it so clearly, here comes a direct proof. First we note that the probability that a free jump ends with a delta interaction is

$$p_{\delta} = \int_{s_{\max}}^{\infty} p(s) ds = \exp(-s_{\max}/\lambda_T). \quad (1.102)$$

To obtain the probability  $p(s)ds$  of having the first real interaction at a distance in the interval  $(s, s + ds)$ , we write  $s = ns_{\max} + s'$  with  $n = [s/s_{\max}]$  and, hence,  $s' < s_{\max}$ . The sought probability is then equal to the probability of having  $n$  successive delta interactions followed by a real interaction at a distance in  $(s', s' + ds)$  from the last,  $n$ -th, delta interaction,

$$p(s) ds = p_{\delta}^n \lambda_T^{-1} \exp(-s'/\lambda_T) ds = \lambda_T^{-1} \exp(-s/\lambda_T) ds, \quad (1.103)$$

which is the correct value [cf. eq. (1.88)].

Up to this point, we have considered transport in a single homogeneous medium. In practical cases, however, the material structure where radiation is transported may consist of various regions with different compositions. We assume that the interfaces between contiguous media are sharp (i.e. there is no diffusion of chemical species across them) and passive (which amounts to neglecting e.g. surface plasmon excitation and transition radiation). In the simulation code, when a particle arrives at an interface, it is stopped there and the simulation is resumed with the interaction properties of the new medium. Obviously, this procedure is consistent with the Markovian property of the transport process.

Consider two homogeneous media, 1 and 2 (with corresponding mean free paths  $\lambda_{T,1}$  and  $\lambda_{T,2}$ ), separated by an interface, which is crossed by particles that move from the first medium to the second. The average path length between the last real interaction in medium 1 and the first real interaction in medium 2 is  $\lambda_{T,1} + \lambda_{T,2}$ , as can be easily verified by simulation. This result seemed paradoxical to some authors and induced confusion in the past. In fact, there is nothing odd here as you may easily verify (again by simulation) as follows. Assume particles being transported within a single homogeneous medium with an imaginary plane that acts as a “virtual” interface, splitting the medium into two halves. In the simulation, the particles do not see this interface, i.e. they do not stop when crossing. Every time a particle crosses the plane, we score the length  $s_{\text{plane}}$  of the track segment between the two real interactions immediately before and after the crossing. It is found that the average value of  $s_{\text{plane}}$  is  $2\lambda_T$ , in spite of the fact that the free path length between consecutive collisions was sampled from an exponential PDF

with the mean free path  $\lambda_T$  [yes, the scored values  $s_{\text{plane}}$  were generated from this PDF!]. The explanation of this result is that, as a consequence of the Markovian character, the average path length from the plane (an arbitrary *fixed* point in the track) back to the last collision (or up to the next collision) is  $\lambda_T$ .

## 1.5 Statistical averages and uncertainties

For the sake of being more specific, let us consider the simulation of a high-energy electron beam impinging on the surface of a semi-infinite water phantom. Each primary electron originates a shower of electrons and photons, which are individually tracked down to the corresponding absorption energy. Any quantity of interest  $Q$  is evaluated as the average score of a large number  $N$  of simulated random showers. Formally,  $Q$  can be expressed as an integral of the form (1.64),

$$Q = \int q p(q) dq, \quad (1.104)$$

where the PDF  $p(q)$  is usually unknown. The simulation of individual showers provides a practical method to sample  $q$  from the “natural” PDF  $p(q)$ : from each generated shower we get a random value  $q_i$  distributed according to  $p(q)$ . The only difference to the case of Monte Carlo integration considered above is that now the PDF  $p(q)$  describes a cascade of random interaction events, each with its characteristic PDF. The Monte Carlo estimate of  $Q$  is

$$\overline{Q} = \frac{1}{N} \sum_{i=1}^N q_i. \quad (1.105)$$

Thus, for instance, the average energy  $E_{\text{dep}}$  deposited within the water phantom per incident electron is obtained as

$$E_{\text{dep}} = \frac{1}{N} \sum_{i=1}^N e_i, \quad (1.106)$$

where  $e_i$  is the energy deposited by *all* the particles of the  $i$ -th shower. The statistical uncertainty (standard deviation) of the Monte Carlo estimate [eq. (1.72)] is

$$\sigma_Q = \sqrt{\frac{\text{var}(q)}{N}} = \sqrt{\frac{1}{N} \left[ \frac{1}{N} \sum_{i=1}^N q_i^2 - \overline{Q}^2 \right]}. \quad (1.107)$$

As mentioned above, we shall usually express the simulation result in the form  $\overline{Q} \pm 3\sigma_Q$ , so that the interval  $(\overline{Q} - 3\sigma_Q, \overline{Q} + 3\sigma_Q)$  contains the true value  $Q$  with 99.7% probability. Notice that to evaluate the standard deviation (1.107) we must score the squared contributions  $q_i^2$ . In certain cases, the contributions  $q_i$  can only take the values 0 and 1, and the standard error can be determined without scoring the squares,

$$\sigma_Q = \sqrt{\frac{1}{N} \overline{Q}(1 - \overline{Q})}. \quad (1.108)$$

Simulation/scoring can also be used to compute continuous distributions. The simplest method is to “discretize” the distributions, by treating them as histograms, and to determine the “heights” of the different bars. To make the arguments clear, let us consider the depth-dose distribution  $D(z)$ , defined as the average energy deposited per unit depth and per incident electron within the water phantom.  $D(z)dz$  is the average energy deposited at depths between  $z$  and  $z+dz$  per incident electron, and the integral of  $D(z)$  from 0 to  $\infty$  is the average deposited energy  $E_{\text{dep}}$  (again, per incident electron). Since part of the energy is reflected back from the water phantom (through backscattered radiation),  $E_{\text{dep}}$  is less than the kinetic energy  $E_{\text{inc}}$  of the incident electrons. We are interested in determining  $D(z)$  in a limited depth interval, say from  $z = 0$  to  $z = z_{\text{max}}$ . The calculation proceeds as follows. First of all, we have to select a partition of the interval  $(0, z_{\text{max}})$  into  $M$  different depth bins  $(z_{k-1}, z_k)$ , with  $0 = z_0 < z_1 < \dots < z_M = z_{\text{max}}$ . Let  $e_{ij,k}$  denote the amount of energy deposited into the  $k$ -th bin by the  $j$ -th particle of the  $i$ -th shower (each incident electron may produce multiple secondary particles). The average energy deposited into the  $k$ -th bin (per incident electron) is obtained as

$$E_k = \frac{1}{N} \sum_{i=1}^N e_{i,k} \quad \text{with} \quad e_{i,k} \equiv \sum_j e_{ij,k}, \quad (1.109)$$

and is affected by a statistical uncertainty

$$\sigma_{Ek} = \sqrt{\frac{1}{N} \left[ \frac{1}{N} \sum_{i=1}^N e_{i,k}^2 - E_k^2 \right]}. \quad (1.110)$$

The Monte Carlo depth-dose distribution  $D_{\text{MC}}(z)$  is a stepwise constant function,

$$D_{\text{MC}}(z) = D_k \pm 3\sigma_{Dk} \quad \text{for } z_{k-1} < z < z_k \quad (1.111)$$

with

$$D_k \equiv \frac{1}{z_k - z_{k-1}} E_k, \quad \sigma_{Dk} \equiv \frac{1}{z_k - z_{k-1}} \sigma_{Ek}. \quad (1.112)$$

Notice that the bin average and standard deviation have to be divided by the bin width to obtain the final Monte Carlo distribution. Defined in this way,  $D_{\text{MC}}(z)$  is an unbiased estimator of the *average* dose in each bin. The limitation here is that we are approximating the continuous distribution  $D(z)$  as a histogram with finite bar widths. In principle, we could obtain a closer approximation by using narrower bins. However, care has to be taken in selecting the bin widths since statistical uncertainties may completely hide the information in narrow bins.

A few words regarding programming details are in order. To evaluate the average deposited energy and its standard deviation for each bin, eqs. (1.109) and (1.110), we must score the shower contributions  $e_{i,k}$  and their squares  $e_{i,k}^2$ . There are cases in which a senseless literal application of this recipe may take a large fraction of the simulation time. Consider, for instance, the simulation of the 3D dose distribution in the phantom, which may involve several thousand volume bins. For each bin, the energies  $e_{ij,k}$  deposited by the individual particles of a shower must be accumulated in a partial counter to obtain

the shower contribution  $e_{i,k}$  and, after completion of the whole shower, the value  $e_{i,k}$  and its square must be added to the accumulated counters. As only a small fraction of the bins receive energy from a single shower, it is not practical to treat all bin counters on an equal footing. The fastest method is to transfer partial scores to the accumulated counters only when the partial counter is going to receive a contribution from a new shower. This can be easily implemented in a computer program as follows. For each quantity of interest, say  $Q$ , we define three real counters,  $Q$ ,  $Q2$  and  $QP$ , and an integer label  $LQ$ ; all these quantities are initially set to zero. The partial scores  $q_{ij}$  of the particles of a shower are accumulated in the partial counter  $QP$ , whereas the global shower contribution  $q_i$  and its square are accumulated in  $Q$  and  $Q2$ , respectively. Each shower is assigned a label, for instance its order number  $i$ , which is stored in  $LQ$  the first time that the shower contributes to  $QP$ . In the course of the simulation, the value of  $QP$  is transferred to the global counters  $Q$  and  $Q2$  only when it is necessary to store a contribution  $q_{ij}$  from a new shower. Explicitly, the FORTRAN code for scoring  $Q$  is

```

      IF (i.NE.LQ) THEN
        Q=Q+QP
        Q2=Q2+QP**2
        QP=qij
        LQ=i
      ELSE
        QP=QP+qij
      ENDIF

```

At the end of the simulation, the residual contents of  $QP$  must be transferred to the global counters.

For some quantities (e.g. the mean number of scattering events per track, the depth-dose function, ...) almost all the simulated tracks contribute to the score and the inherent statistical uncertainties of the simulation results are comparatively small. Other quantities (e.g. angle and energy distributions of the particles transmitted through a thick foil) have considerable statistical uncertainties (i.e. large variances) because only a small fraction of the simulated tracks contribute to the partial scores.

## 1.6 Variance reduction

In principle, the statistical error of a quantity may be somewhat reduced (without increasing the computer simulation time) by using variance-reduction techniques. Unfortunately, these optimization techniques are extremely problem-dependent, and general recipes to minimize the variance cannot be given. On the other hand, the importance of variance reduction should not be overvalued. In many cases, analogue<sup>3</sup> simulation does the work in a reasonable time. Spending manhours by complicating the program, to get a modest reduction in computing time may not be a good investment. It is

---

<sup>3</sup>We use the term “analogue” to refer to detailed, condensed or mixed simulations that do not incorporate variance-reduction procedures.

also important to realize that an efficient variance-reduction method usually lowers the statistical error of a given quantity  $Q$  at the expense of increasing the uncertainties of other quantities. Thus, variance-reduction techniques are not recommended when a global description of the transport process is sought. Here we give a brief description of those techniques which, with a modest programming effort, can be useful in improving the solution of some ill-conditioned problems. For the sake of generality, we consider that secondary particles can be generated in the interactions with the medium. A nice, and practically oriented, review of variance-reduction methods in radiation transport has been given by Bielajew and Rogers (1988).

### 1.6.1 Interaction forcing

Sometimes, a high variance results from an extremely low interaction probability. Consider, for instance, the simulation of the energy spectrum of bremsstrahlung photons emitted by medium energy ( $\sim 100$  keV) electrons in a thin foil of a certain material. As radiative events are much less probable than elastic and inelastic scattering, the uncertainty of the simulated photon spectrum will be relatively large. In such cases, an efficient variance-reduction method is to artificially increase the interaction probability of the process A of interest. Our practical implementation of interaction forcing consists of replacing the mean free path  $\lambda_A$  of the real process by a shorter one,  $\lambda_{A,f}$ , i.e. we force A interactions to occur more frequently than for the real process. We consider that the PDF for the energy loss, the angular deflections (and the directions of emitted secondary particles, if any) in the forced interactions is the same as for the real interactions. To sample the length of the free jump to the next interaction, we use the exponential distribution with the reduced mean free path  $\lambda_{A,f}$ . This is equivalent to increasing the interaction probability per unit path length of the process A by a factor

$$\mathcal{F} = \frac{\lambda_A}{\lambda_{A,f}} > 1. \quad (1.113)$$

To keep the simulation unbiased, we must correct for the introduced distortion as follows:

- (i) A weight  $w_p^{(1)} = 1$  is associated with each primary particle. Secondary particles produced in forced interactions have an associated weight  $w_p^{(2)} = w_p^{(1)}/\mathcal{F}$ ; the weights of successive generations of forced secondaries are  $w_p^{(k)} = w_p^{(k-1)}/\mathcal{F}$ . Secondary particles generated in non-forced interactions (i.e. of types other than A) are given a weight equal to that of their parent particle.
- (ii) A weight  $w_E^{(k)} = w_p^{(k)}/\mathcal{F}$  is given to the deposited energy (and to any other alteration of the medium such as e.g. charge deposition) that results from forced interactions of a particle with weight  $w_p^{(k)}$ . For non-forced interactions  $w_E^{(k)} = w_p^{(k)}$ .
- (iii) Forced interactions are simulated to determine the energy loss and possible emission of secondary radiation, but the state variables of the interacting particle are altered only with probability  $1/\mathcal{F}$ . That is, the energy  $E$  and direction of movement  $\hat{\mathbf{d}}$  of the projectile are varied only when the value  $\xi$  of a random number falls below  $1/\mathcal{F}$ , otherwise  $E$  and  $\hat{\mathbf{d}}$  are kept unchanged.

Of course, interaction forcing should be applied only to interactions that are dynamically allowed, i.e. for particles with energy above the corresponding “reaction” threshold.

Let  $w_{i1}$  and  $q_{i1}$  denote the weight and the contribution to the score of the  $i$ -th primary, and let  $w_{ij}$  and  $q_{ij}$  ( $j > 1$ ) represent the weights and contributions of the  $j$ -th secondary particles generated by the  $i$ -th primary. The Monte Carlo estimate of  $Q$  obtained from the  $N$  simulated histories is

$$\bar{Q} = \frac{1}{N} \sum_{i,j} w_{ij} q_{ij}. \quad (1.114)$$

Evidently, the estimates  $\bar{Q}$  obtained with interaction forcing and from an analogue simulation are equal (in the statistical sense, i.e. in the limit  $N \rightarrow \infty$ , their difference tends to zero). The standard deviation is given by

$$\sigma_Q = \sqrt{\frac{1}{N} \left[ \frac{1}{N} \sum_i \left( \sum_j w_{ij} q_{ij} \right)^2 - \bar{Q}^2 \right]}. \quad (1.115)$$

Quantities directly related to the forced interactions will have a reduced statistical error, due to the increase in number of these interactions. However, for a given simulation time, other quantities may exhibit standard deviations larger than those of the analogue simulation, because of the time spent in simulating the forced interactions.

### 1.6.2 Splitting and Russian roulette

These two techniques, which are normally used in conjunction, are effective in problems where interest is focused on a localized spatial region. Typical examples are the calculation of dose functions in deep regions of irradiated objects and, in the case of collimated radiation beams, the evaluation of radial doses far from the beam axis. The basic idea of splitting and Russian roulette methods is to favour the flux of radiation towards the region of interest and inhibit the radiation that leaves that region. These techniques are also useful in other problems where only a partial description of the transport process is required. The “region of interest” may then be a limited volume in the space of state variables  $(\mathbf{r}, E, \hat{\mathbf{d}})$ . Thus, in studies of radiation backscattering, the region of interest may be selected as the spatial region of the sample close to the irradiated surface *and* the set of particle directions that point towards this surface.

As in the case of interaction forcing, variance reduction is accomplished by modifying the weights of the particles. It is assumed that primary particles start moving with unit weight and each secondary particle produced by a primary one is assigned an initial weight equal to that of the primary. Splitting consists of transforming a particle, with weight  $w_0$  and in a certain state, into a number  $\mathcal{S} > 1$  of identical particles with weights  $w = w_0/\mathcal{S}$  in the same initial state. Splitting should be applied when the particle “approaches” the region of interest. The Russian roulette technique is, in a way, the reverse process: when a particle tends to move away from the region of interest it is



“killed” with a certain probability,  $\mathcal{K} < 1$ , and, if it survives, its weight is increased by a factor  $1/(1 - \mathcal{K})$ . Here, killing means that the particle is just discarded (and does not contribute to the scores anymore). Evidently, splitting and killing leave the simulation unbiased. The mean and standard deviation of the calculated quantities are given by eqs. (1.114) and (1.115). The effectiveness of these methods relies on the adopted values of the parameters  $\mathcal{S}$  and  $\mathcal{K}$ , and on the strategy used to decide when splitting and killing are to be applied. These details can only be dictated by the user’s experience.

### 1.6.3 Other methods

Very frequently, an effective “reduction of variance” may be obtained by simply avoiding unnecessary calculations. This is usually true for simulation codes that incorporate “general-purpose” geometry packages. In the case of simple (e.g. planar, spherical, cylindrical) geometries the program may be substantially simplified and this may speed up the simulation appreciably. In general, the clever use of possible symmetries of the problem under consideration may lead to spectacular variance reductions. As a last example, we can quote the so-called “range rejection” method, which simply consists of absorbing a particle when it (and its possible secondaries) cannot leave (or reach) the regions of interest. Range rejection is useful e.g. when computing the total energy deposition of electrons or positrons in a given spatial region. When the residual range of a particle is less than the distance to the nearest limiting surface of the region of interest, the particle will deposit all its energy inside or outside the considered region (depending of its current position) and the simulation of the track can be stopped. Range rejection is not adequate for photon transport simulation, since the concept of photon range is not well defined (or, to be more precise, photon path length fluctuations are very large).