Chapter 2

Monte Carlo Integration

This chapter gives an introduction to Monte Carlo integration. The main goals are to review some basic concepts of probability theory, to define the notation and terminology that we will be using, and to summarize the variance reduction techniques that have proven most useful in computer graphics.

Good references on Monte Carlo methods include Kalos & Whitlock [1986], Hammersley & Handscomb [1964], and Rubinstein [1981]. Sobol' [1994] is a good starting point for those with little background in probability and statistics. Spanier & Gelbard [1969] is the classic reference for Monte Carlo applications to neutron transport problems; Lewis & Miller [1984] is a good source of background information in this area. For quasi-Monte Carlo methods, see Niederreiter [1992], Beck & Chen [1987], and Kuipers & Niederreiter [1974].

2.1 A brief history

Monte Carlo methods originated at the Los Alamos National Laboratory in the early years after World War II. The first electronic computer in the United States had just been completed (the ENIAC), and the scientists at Los Alamos were considering how to use it for the design of thermonuclear weapons (the H-bomb). In late 1946 Stanislaw Ulam suggested the use of random sampling to simulate the flight paths of neutrons, and John von Neumann developed a detailed proposal in early 1947. This led to small-scale simulations whose results were indispensable in completing the project. Metropolis & Ulam [1949] published a paper in 1949 describing their ideas, which sparked to a great deal of research in the 1950's [Meyer 1956]. The name of the Monte Carlo method comes from a city in Monaco, famous for its casinos (as suggested by Nick Metropolis, another Monte Carlo pioneer).

In isolated instances, random sampling had been used much earlier to solve numerical problems [Kalos & Whitlock 1986]. For example, in 1777 the Comte de Buffon performed an experiment in which a needle was dropped many times onto a board marked with equidistant parallel lines. Letting L be the length of the needle and $d > L$ be the distance between the lines, he showed that the probability of the needle intersecting a line is

$$p = \frac{2L}{\pi d}.$$

Many years later, Laplace pointed out that this could be used as a crude means of estimating the value of π.

Similarly, Lord Kelvin used what we would now call a Monte Carlo method to study some aspects of the kinetic theory of gases. His random number generator consisted of drawing slips of paper out of a glass jar. The possibility of bias was a significant concern; he worried that the papers might not be mixed well enough due to static electricity. Another early Monte Carlo experimenter was Student (an alias for W. S. Gosset), who used random sampling as an aid to guessing the form of his famous t-distribution.

An excellent reference on the origins of Monte Carlo methods is the special issue of *Los Alamos Science* published in memory of Stanislaw Ulam [Ulam 1987]. The books by Kalos & Whitlock [1986] and Hammersley & Handscomb [1964] also contain brief histories, including information on the pre-war random sampling experiments described above.

2.2 Quadrature rules for numerical integration

In this section we explain why standard numerical integration techniques do not work very well on high-dimensional domains, especially when the integrand is not smooth.

Consider an integral of the form

$$I = \int_\Omega f(x)\, d\mu(x)\,, \tag{2.1}$$

where Ω is the domain of integration, $f : \Omega \to \mathbb{R}$ is a real-valued function, and μ is a measure function on Ω.[1] For now, let the domain be the s-dimensional unit hypercube,

$$\Omega = [0,1]^s\,,$$

and let the measure function be

$$d\mu(x) = dx^1 \cdots dx^s\,,$$

where x^j denotes the j-th component of the point $x = (x^1, \ldots, x^s) \in [0,1]^s$.

Integrals of this sort are often approximated using a *quadrature rule*, which is simply a sum of the form

$$\hat{I} = \sum_{i=1}^{N} w_i\, f(x_i) \tag{2.2}$$

where the weights w_i and sample locations x_i are determined in advance. Common examples of one-dimensional quadrature rules include the *Newton-Cotes rules* (i.e. the midpoint rule, the trapezoid rule, Simpson's rule, and so on), and the *Gauss-Legendre rules* (see Davis & Rabinowitz [1984] for further details). The n-point forms of these rules typically obtain a convergence rate of $O(n^{-r})$ for some integer $r \geq 1$, provided that the integrand has sufficiently many continuous derivatives. For example, the error using Simpson's rule is $O(n^{-4})$, provided that f has at least four continuous derivatives [Davis & Rabinowitz 1984].

Although these quadrature rules typically work very well for one-dimensional integrals, problems occur when extending them to higher dimensions. For example, a common approach is to use *tensor product rules* of the form

$$\hat{I} = \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \cdots \sum_{i_s=1}^{n} w_{i_1} w_{i_2} \cdots w_{i_s}\, f(x_{i_1}, x_{i_2}, \ldots, x_{i_s})$$

where s is the dimension, and the w_i and x_i are the weights and sample locations for a given

[1] Familiar examples of measures include length, surface area, volume, and solid angle; see Halmos [1950] for an introduction to measure theory.

one-dimensional rule. This method has the same convergence rate as the one-dimensional rule on which it is based (let this be $O(n^{-r})$), however it uses a much larger number of sample points (namely $N = n^s$). Thus in terms of the total number of samples, the convergence rate is only $O(N^{-r/s})$. This implies that the efficiency of tensor product rules diminishes rapidly with dimension, a fact that is often called the *curse of dimensionality* [Niederreiter 1992, p. 2].

The convergence rate can be increased by using a one-dimensional rule with a larger value of r, however this has two problems. First, the total number of samples $N = n^s$ can become impractical in high dimensions, since n increases linearly with r (specifically, $n \geq r/2$). For example, two-point Guass quadrature requires at least 2^s samples, while Simpson's rule requires at least 3^s samples. Second, faster convergence rates require more smoothness in the integrand. For example, if the function f has a discontinuity, then the convergence rate of any one-dimensional quadrature rule is at best $O(n^{-1})$ (assuming that the location of the discontinuity is not known in advance), so that the corresponding tensor product rule converges at a rate no better than $O(N^{-1/s})$.

Of course, not all multidimensional integration rules take the form of tensor products. However, there is an important result which limits the convergence rate of any deterministic quadrature rule, called *Bakhvalov's theorem* [Davis & Rabinowitz 1984, p. 354]. Essentially, it says that given any s-dimensional quadrature rule, there is function f with r continuous and bounded derivatives, for which the error is proportional to $N^{-r/s}$. Specifically, let C_M^r denote the set of functions $f : [0,1]^s \to \mathbb{R}$ such that

$$\left| \frac{\partial^r f}{\partial(x^1)^{a_1} \cdots \partial(x^s)^{a_s}} \right| \;\leq\; M$$

for all $a_1, \ldots, a_s$ with $\sum a_i = r$, recalling that x^j denotes the j-th coordinate of the vector x. Now consider any N-point quadrature rule

$$\hat{I}(f) \;=\; \sum_{i=1}^{N} w_i \, f(x_i)$$

where each x_i is a point in $[0,1]^s$, and suppose that we wish to approximate some integral

$$I(f) \;=\; \int_{[0,1]^s} f(x^1, \ldots, x^s) \, dx^1 \cdots dx^s \, .$$

Then according to Bakhvalov's theorem, there is a function $f \in C^r_M$ such that the error is

$$\left|\hat{I}(f) - I(f)\right| \;>\; k \cdot N^{-r/s}\,,$$

where the constant $k > 0$ depends only on M and r. Thus even if f has a bounded, continuous first derivative, no quadrature rule has an error bound better than $O(N^{-1/s})$.

2.3 A bit of probability theory

Before describing Monte Carlo integration, we review a few concepts from probability and statistics. See Pitman [1993] for an introduction to probability, and Halmos [1950] for an introduction to measure theory. Brief introductions to probability theory can also be found in the Monte Carlo references cited above.

2.3.1 Cumulative distributions and density functions

Recall that the *cumulative distribution function* of a real-valued random variable X is defined as

$$P(x) \;=\; Pr\left\{X \le x\right\}\,,$$

and that the corresponding *probability density function* is

$$p(x) \;=\; \frac{dP}{dx}(x)$$

(also known as the *density function* or *pdf*). This leads to the important relationship

$$Pr\left\{\alpha \le X \le \beta\right\} \;=\; \int_\alpha^\beta p(x)\,dx \;=\; P(\beta) - P(\alpha)\,. \tag{2.3}$$

The corresponding notions for a multidimensional random vector $(X^1, \ldots, X^s)$ are the *joint cumulative distribution function*

$$P(x^1, \ldots, x^s) \;=\; Pr\left\{X^i \le x^i \text{ for all } i = 1, \ldots, s\right\}$$

and the *joint density function*

$$p(x^1, \ldots, x^s) \;=\; \frac{\partial^s P}{\partial x^1 \cdots \partial x^s}(x^1, \ldots, x^s)\,,$$

so that we have the relationship

$$Pr\,\{x \in D\} \;=\; \int_D p(x^1, \ldots, x^s)\, dx^1 \cdots dx^s \tag{2.4}$$

for any Lebesgue measurable subset $D \subset \mathbb{R}^s$.

More generally, for a random variable X with values in an arbitrary domain Ω, its *probability measure* (also known as a *probability distribution* or *distribution*) is a measure function P such that

$$P(D) \;=\; Pr\,\{X \in D\}$$

for any measurable set $D \subset \Omega$. In particular, a probability measure must satisfy $P(\Omega) = 1$. The corresponding density function p is defined as the *Radon-Nikodym derivative*

$$p(x) \;=\; \frac{dP}{d\mu}(x)\,,$$

which is simply the function p that satisfies

$$P(D) \;=\; \int_D p(x)\, d\mu(x)\,. \tag{2.5}$$

Thus, the probability that $X \in D$ can be obtained by integrating $p(x)$ over the given region D. This should be compared with equations (2.3) and (2.4), which are simply special cases of the more general relationship (2.5).

Note that the density function p depends on the measure μ used to define it. We will use the notation $p = P_\mu$ to denote the density with respect to a particular measure μ, corresponding to the notation $u_x = \partial u \,/\, \partial x$ that is often used in analysis. This notation will be useful when there are several relevant measure function defined on the same domain Ω (for example, the solid angle and projected solid angle measures that will be described in Chapter 3). See Halmos [1950] for further information on measure spaces and Radon-Nikodym derivatives.

2.3.2 Expected value and variance

The *expected value* or *expectation* of a random variable $Y = f(X)$ is defined as

$$E[Y] \;=\; \int_{\Omega} f(x)\, p(x)\, d\mu(x)\,, \tag{2.6}$$

while its *variance* is

$$V[Y] \;=\; E\Big[(Y - E[Y])^2\Big]\,. \tag{2.7}$$

We will always assume that expected value and variance of every random variable exist (i.e. the corresponding integral is finite).

From these definitions, it is easy to see that for any constant a we have

$$\begin{aligned} E[a\,Y] &= a\,E[Y]\,, \\ V[a\,Y] &= a^2\,V[Y]\,. \end{aligned}$$

The following identity is also useful:

$$E\left[\sum_{i=1}^{N} Y_i\right] \;=\; \sum_{i=1}^{N} E[Y_i]\,,$$

which holds for any random variables $Y_1, \ldots, Y_N$. On the other hand, the following identity holds only if the variables Y_i are independent:

$$V\left[\sum_{i=1}^{N} Y_i\right] \;=\; \sum_{i=1}^{N} V[Y_i]\,.$$

Notice that from these rules, we can derive a simpler expression for the variance:

$$V[Y] \;=\; E[(Y - E[Y])^2] \;=\; E[Y^2] - E[Y]^2\,.$$

Another useful quantity is the *standard deviation* of a random variable, which is simply the square root of its variance:

$$\sigma[Y] \;=\; \sqrt{V[Y]}\,.$$

This is also known as the *RMS error*.

2.3.3 Conditional and marginal densities

Let $X \in \Omega_1$ and $Y \in \Omega_2$ be a pair of random variables, so that

$$(X, Y) \in \Omega$$

where $\Omega = \Omega_1 \times \Omega_2$. Let P be the joint probability measure of (X, Y), so that $P(D)$ represents the probability that $(X, Y) \in D$ for any measurable subset $D \subset \Omega$. Then the corresponding joint density function $p(x, y)$ satisfies

$$P(D) \;=\; \int_D p(x, y)\, d\mu_1(x)\, d\mu_2(y)\,,$$

where μ_1 and μ_2 are measures on Ω_1 and Ω_2 respectively. Hereafter we will drop the measure function notation, and simply write

$$P(D) \;=\; \int_D p(x, y)\, dx\, dy\,.$$

The *marginal density function* of X is now defined as

$$p(x) \;=\; \int_{\Omega_2} p(x, y)\, dy\,, \tag{2.8}$$

while the *conditional density function* $p(y \,|\, x)$ is defined as

$$p(y \,|\, x) \;=\; p(x, y) \,/\, p(x)\,. \tag{2.9}$$

The marginal density $p(y)$ and conditional density $p(x \mid y)$ are defined in a similar way, leading to the useful identity

$$p(x, y) \;=\; p(y \,|\, x)\, p(x) \;=\; p(x \,|\, y)\, p(y)\,.$$

Another important concept is the *conditional expectation* of a random variable $G = g(X, Y)$, defined as

$$E[G \,|\, x] \;=\; \int_{\Omega_2} g(x, y)\, p(y \,|\, x)\, dy \;=\; \frac{\int g(x, y)\, p(x, y)\, dy}{\int p(x, y)\, dy}\,. \tag{2.10}$$

We will also use the notation $E_Y[G]$ for the conditional expectation, which emphasizes the fact that Y is the random variable whose density function is being integrated.

There is a very useful expression for the variance of G in terms of its conditional expectation and variance, namely

$$V[G] \;=\; E_X V_Y G + V_X E_Y G\,. \tag{2.11}$$

In other words, $V[G]$ is the mean of the conditional variance, plus the variance of the conditional mean. To prove this identity, recall that

$$V[F] \;=\; E[F^2] - E[F]^2\,,$$

and observe that

$$\begin{aligned} E_X V_Y G + V_X E_Y G &= E_X \left\{ E_Y[G^2] - [E_Y G]^2 \right\} + E_X[E_Y G]^2 - [E_X E_Y G]^2 \\ &= E_X E_Y[G^2] - [E_X E_Y G]^2 \\ &= V[G]\,. \end{aligned}$$

We will use this identity below to analyze certain variance reduction techniques, including stratified sampling and the use of expected values.

2.4 Basic Monte Carlo integration

The idea of Monte Carlo integration is to evaluate the integral

$$I \;=\; \int_\Omega f(x)\, d\mu(x)$$

using random sampling. In its basic form, this is done by independently sampling N points $X_1, \ldots, X_N$ according to some convenient density function p, and then computing the estimate

$$F_N \;=\; \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}\,. \tag{2.12}$$

Here we have used the notation F_N rather than $\hat{I}$ to emphasize that the result is a random variable, and that its properties depend on how many sample points were chosen. Note that this type of estimator was first used in the survey sampling literature (for discrete rather than continuous domains), where it is known as the *Horvitz-Thompson estimator* [Horvitz

& Thompson 1952].

For example, suppose that the domain is $\Omega = [0,1]^s$ and that the samples X_i are chosen independently and uniformly at random. In this case, the estimator (2.12) reduces to

$$F_N \;=\; \frac{1}{N} \sum_{i=1}^{N} f(X_i)\,,$$

which has the same form as a quadrature rule except that the sample locations are random.

It is straightforward to show the estimator F_N gives the correct result on average. Specifically, we have

$$\begin{aligned} E[F_N] \;&=\; E\left[\frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}\right] \\ &=\; \frac{1}{N} \sum_{i=1}^{N} \int_\Omega \frac{f(x)}{p(x)}\, p(x)\, d\mu(x) \\ &=\; \int_\Omega f(x)\, d\mu(x) \\ &=\; I\,, \end{aligned}$$

provided that $f(x)/p(x)$ is finite whenever $f(x) \neq 0$.

Advantages of Monte Carlo integration. Monte Carlo integration has the following major advantages. First, it converges at a rate of $O(N^{-1/2})$ in any dimension, regardless of the smoothness of the integrand. This makes it particularly useful in graphics, where we often need to calculate multi-dimensional integrals of discontinuous functions. The convergence rate is discussed in Section 2.4.1 below.

Second, Monte Carlo integration is simple. Only two basic operations are required, namely sampling and point evaluation. This encourages the use of object-oriented *black box* interfaces, which allow great flexibility in the design of Monte Carlo software. In the context of computer graphics, for example, it is straightforward to include effects such motion blur, depth of field, participating media, procedural surfaces, and so on.

Third, Monte Carlo is general. Again, this stems from the fact that it is based on random sampling. Sampling can be used even on domains that do not have a natural correspondence with $[0,1]^s$, and are thus not well-suited to numerical quadrature. As an example of this in graphics, we observe that the light transport problem can be naturally expressed as an integral over the space of all transport paths (Chapter 8). This domain is technically an infinite-dimensional space (which would be difficult to handle with numerical quadrature), but it is straightforward to handle with Monte Carlo.

Finally, Monte Carlo methods are better suited than quadrature methods for integrands with singularities. Importance sampling (see Section 2.5.2) can be applied to handle such integrands effectively, even in situations where there is no analytic transformation to remove the singularity (see the discussion of rejection sampling and the Metropolis method below).

In the remainder of this section, we discuss the convergence rate of Monte Carlo integration, and give a brief review of sampling techniques for random variables. We then discuss the properties of more general kinds of Monte Carlo estimators.

2.4.1 Convergence rates

To determine the convergence rate of Monte Carlo integration, we start by computing the variance of F_N. To simplify the notation let $Y_i = f(X_i)/p(X_i)$, so that

$$F_N \;=\; \frac{1}{N} \sum_{i=1}^{N} Y_i \,.$$

Also let $Y = Y_1$. We then have

$$V[Y] \;=\; E[Y^2] - E[Y]^2 \;=\; \int_\Omega \frac{f^2(x)}{p(x)}\, d\mu(x) \;-\; I^2 \,.$$

Assuming that this quantity is finite, it is easy to check that the variance of $V[F_N]$ decreases linearly with N:

$$V[F_N] \;=\; V\left[\frac{1}{N} \sum_{i=1}^{N} Y_i\right] \;=\; \frac{1}{N^2} V\left[\sum_{i=1}^{N} Y_i\right] \;=\; \frac{1}{N^2} \sum_{i=1}^{N} V[Y_i] \;=\; \frac{1}{N} V[Y] \qquad (2.13)$$

where we have used $V[a\,Y] \;=\; a^2\, V[Y]$ and the fact that the Y_i are independent samples. Thus the standard deviation is

$$\sigma[F_N] \;=\; \frac{1}{\sqrt{N}}\, \sigma Y \,,$$

which immediately shows that the RMS error converges at a rate of $O(N^{-1/2})$.

It is also possible to obtain probabilitistic bounds on the absolute error, using *Chebychev's inequality*:

$$Pr\left\{|F - E[F]| \;\geq\; \left(\frac{V[F]}{\delta}\right)^{1/2}\right\} \;\leq\; \delta\,,$$

which holds for any random variable F such that $V[F] < \infty$. Applying this inequality to the variance (2.13), we obtain

$$Pr\left\{|F_N - I| \;\geq\; N^{-1/2}\left(\frac{V[Y]}{\delta}\right)^{1/2}\right\} \;\leq\; \delta\,.$$

Thus for any fixed threshold δ, the absolute error decreases at the rate $O(N^{-1/2})$.

Tighter bounds on the absolute error can be obtained using the *central limit theorem*, which states that F_N converges to a normal distribution in the limit as $N \to \infty$. Specifically, it states that

$$\lim_{N\to\infty} Pr\left\{\frac{1}{N}\sum_{i=1}^{N} Y_i - E[Y] \;\leq\; t\,\frac{\sigma[Y]}{\sqrt{N}}\right\} \;=\; \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{t} e^{-x^2/2}\,dx\,,$$

where the expression on the right is the (cumulative) normal distribution. This equation can be rearranged to give

$$Pr\left\{|F_N - I| \;\geq\; t\,\sigma[F_N]\right\} \;=\; \sqrt{2/\pi}\int_{t}^{\infty} e^{-x^2/2}\,dx\,.$$

The integral on the right decreases very quickly with t; for example when $t = 3$ the right-hand side is approximately 0.003. Thus, there is only about a 0.3% chance that F_N will differ from its mean by more than three standard deviations, provided that N is large enough for the central limit theorem to apply.

Finally, note that Monte Carlo integration will converge even if the variance $V[Y]$ is infinite, provided that the expectation $E[Y]$ exists (although convergence will be slower). This is guaranteed by the *strong law of large numbers*, which states that

$$Pr\left\{\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N} Y_i \;=\; E[Y]\right\} \;=\; 1\,.$$

2.4.2 Sampling random variables

There are a variety of techniques for sampling random variables, which we briefly review here. Further details can be found in the references given in the introduction.

One method is the *transformation* or *inversion* method. In one dimension, suppose that we want to sample from a density function p. Letting P be the corresponding cumulative distribution function, the inversion method consists of letting $X = P^{-1}(U)$, where U is a uniform random variable on $[0, 1]$. It is easy to verify that X has the required density p. This technique can easily be extended to several dimensions, either by computing marginal and conditional distributions and inverting each dimension separately, or more generally by deriving a transformation $x = g(u)$ with an appropriate Jacobian determinant (such that $|\det(J_g(x))|^{-1} = p(x)$, where J_g denotes the Jacobian of g).

The main advantage of the transformation technique is that it allows samples to be stratified easily, by stratifying the parameter space $[0, 1]^s$ and mapping these samples into Ω (see Section 2.6.1). Another advantage is that the technique has a fixed cost per sample, which can easily be estimated. The main disadvantage is that the density $p(x)$ must be integrated analytically, which is not always possible. It is also preferable for the cumulative distribution to have an analytic inverse, since numerical inversion is typically slower.

A second sampling technique is the *rejection method*, due to von Neumann [Ulam 1987]. The idea is to sample from some convenient density q such that

$$p(x) \;\leq\; M\, q(x)$$

for some constant M. Generally, the samples from q are generated by the transformation method. We then apply the following procedure:

function REJECTION-SAMPLING()
  for $i = 1$ **to** ∞
    Sample X_i according to q.
    Sample U_i uniformly on $[0, 1]$.
    if $U_i \;\leq\; p(X_i) \,/\, (M\, q(X_i))$
      then return X_i

It is easy to verify that this procedure generates a sample X whose density function is p.

The main advantage of rejection sampling is that it can be used with any density function, even those that cannot be integrated analytically. However, we still need to be able to integrate some function Mq that is an upper bound for p. Furthermore, this bound should be reasonably tight, since the average number of samples that must be taken before acceptance is M. Thus, the efficiency of rejection sampling can be very low if it is applied naively. Another disadvantage is that it is difficult to apply with stratification: the closest approximation is to stratify the domain of the random vector (X, U), but the resulting stratification is not as good as the transformation method.

A third general sampling technique is the *Metropolis method* (also known as *Markov chain Monte Carlo*), which will be described in Chapter 11. This technique is useful for sampling arbitrary densities on high-dimensional spaces, and has the advantage that the density function does not need to be normalized. The main disadvantage of the Metropolis method is that the samples it generates are not independent; in fact they are highly correlated. Thus, it is most useful when we need to generate a long sequence of samples from the given density p.

Finally, there are various techniques for sampling from specific distributions (see Rubinstein [1981]). For example, if X is the maximum of k independent uniform random variables $U_1, \ldots, U_k$, then X has the density function $p(x) = kx^{k-1}$ (where $0 \le x \le 1$). Such "tricks" can be used to sample many of the standard distributions in statistics, such as the normal distribution [Rubinstein 1981].

2.4.3 Estimators and their properties

So far we have only discussed one way to estimate an integral using random samples, namely the standard technique (2.12). However, there are actually a great variety of techniques available, which are encompassed by the concept of a *Monte Carlo estimator*. We review the various properties of estimators and why they are desirable.

The purpose of a Monte Carlo estimator is to approximate the value of some *quantity of interest* Q (also called the *estimand*). Normally we will define Q as the value of a given integral, although more general situations are possible (e.g. Q could be the ratio of two integrals). An *estimator* is then defined to be a function of the form

$$F_N \;=\; F_N(X_1, \ldots, X_N)\,, \tag{2.14}$$

where the X_i are random variables. A particular numerical value of F_N is called an *estimate*. Note that the X_i are not necessarily independent, and can have different distributions.

Note that there are some differences in the standard terminology for computer graphics, as compared to statistics. In statistics, the value of each X_i is called an *observation*, the vector $(X_1, \ldots, X_N)$ is called the *sample*, and N is called the *sample size*. In computer graphics, on the other hand, typically each of the individual X_i is referred to as a sample, and N is the number of samples. We will normally use the graphics conventions.

We now define a number of useful properties of Monte Carlo estimators. The quantity $F_N - Q$ is called the *error*, and its expected value is called the *bias*:

$$\beta[F_N] \;=\; E[F_N - Q]\,. \tag{2.15}$$

An estimator is called *unbiased* if $\beta[F_N] = 0$ for all sample sizes N, or in other words if

$$E[F_N] \;=\; Q \quad \text{for all } N \geq 1\,. \tag{2.16}$$

For example, the random variable

$$F_N \;=\; \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}$$

is an unbiased estimator of the integral $I = \int_\Omega f(x)\, d\mu(x)$ (as we saw in Section 2.4).

An estimator is called *consistent* if the error $F_N - Q$ goes to zero with probability one, or in other words if

$$Pr\left\{ \lim_{N\to\infty} F_N = Q \right\} \;=\; 1\,. \tag{2.17}$$

For an estimator to be consistent, a sufficient condition is that the bias and variance both go to zero as N is increased:

$$\lim_{N\to\infty} \beta[F_N] \;=\; \lim_{N\to\infty} V[F_N] \;=\; 0\,.$$

In particular, an unbiased estimator is consistent as long as its variance decreases to zero as N goes to infinity.

The main reason for preferring unbiased estimators is that it is easier to estimate the error. Typically our goal is to minimize the *mean squared error* (MSE), defined by

$$MSE[F] \;=\; E[(F-Q)^2] \tag{2.18}$$

(where we have dropped the subscript N). In general, the mean squared error can be rewritten as

$$\begin{aligned} MSE[F] &= E[(F-Q)^2] \\ &= E[(F-E[F])^2] + 2E[F-E[F]](E[F]-Q) + (E[F]-Q)^2 \\ &= V[F] + \beta[F]^2\,, \end{aligned}$$

so that to estimate the error we must have an upper bound on the possible bias. In general, this requires additional knowledge about the estimand Q, and it is often difficult to find a suitable bound.

On the other hand, for unbiased estimators we have $E[F] = Q$, so that the mean squared error is identical to the variance:

$$MSE[F] \;=\; V[F] \;=\; E[(F-E[F])^2]\,.$$

This makes it far easier to obtain error estimates, by simply taking several independent samples. Letting $Y_1, \ldots, Y_N$ be independent samples of an unbiased estimator Y, and letting

$$F_N \;=\; \frac{1}{N}\sum_{i=1}^{N} Y_i$$

as before (which is also an unbiased estimator), then the quantity

$$\hat{V}[F_N] \;=\; \frac{1}{N-1}\left\{\left(\frac{1}{N}\sum_{i=1}^{N} Y_i^2\right) - \left(\frac{1}{N}\sum_{i=1}^{N} Y_i\right)^2\right\}$$

is an unbiased estimator of the variance $V[F_N]$ (see Kalos & Whitlock [1986]). Thus, error estimates are easy to obtain for unbiased estimators.

Notice that by taking many independent samples, the error of an unbiased estimator can be made as small as desired, since

$$V[F_N] \;=\; V[F_1]\,/\,N\,.$$

However, this will also increase the running time by a factor of N. Ideally, we would like to find estimators whose variance and running time are both small. This tradeoff is summarized by the *efficiency* of a Monte Carlo estimator:

$$\epsilon[F] \;=\; \frac{1}{V[F]\,T[F]} \tag{2.19}$$

where $T[F]$ is the time required to evaluate F. Thus the more efficient an estimator is, the lower the variance that can be obtained in a given fixed running time.

2.5 Variance reduction I: Analytic integration

The design of efficient estimators is a fundamental goal of Monte Carlo research. A wide variety of techniques have been developed, which are often simply called *variance reduction methods*. In the following sections, we describe the variance reduction methods that have proven most useful in computer graphics.[2] These methods can be grouped into several categories, based around four main ideas:

- analytically integrating a function that is similar to the integrand;
- uniformly placing sample points across the integration domain;
- adaptively controlling the sample density based on information gathered during sampling; and
- combining samples from two or more estimators whose values are correlated.

[2]Note that some variance reduction methods are useful only for one-dimensional integrals, or only for smooth integrands (e.g. certain antithetic variates transformations [Hammersley & Handscomb 1964]). Since these situations are usually better handled by numerical quadrature, we do not discuss such methods here.

We start by discussing methods based on analytic integration. There are actually several ways to take advantage of this idea, including *the use of expected values*, *importance sampling*, and *control variates*. These are some of the most powerful and useful methods for computer graphics problems.

Note that many variance reduction methods were first proposed in the survey sampling literature, long before Monte Carlo methods were invented. For example, techniques such as stratified sampling, importance sampling, and control variates were all first used in survey sampling [Cochran 1963].

2.5.1 The use of expected values

Perhaps the most obvious way to reduce variance is to reduce the dimension of the sample space, by integrating analytically with respect to one or more variables of the domain. This idea is commonly referred to as *the use of expected values* or *reducing the dimensionality*. Specifically, it consists of replacing an estimator of the form

$$F \;=\; f(X,Y)\,/\,p(X,Y) \tag{2.20}$$

with one of the form

$$F' \;=\; f'(X)\,/\,p(X)\,, \tag{2.21}$$

where $f'(x)$ and $p(x)$ are defined by

$$\begin{aligned} f'(x) &= \int f(x,y)\,dy \\ p(x) &= \int p(x,y)\,dy\,. \end{aligned}$$

Thus, to apply this technique we must be able to integrate both f and p with respect to y. We also must be able to sample from the marginal density $p(x)$, but this can be done by simply generating (X,Y) as before, and ignoring the value of Y.

The name of this technique comes from the fact that the estimator F' is simply the conditional expected value of F:

$$F' \;=\; E_Y\left[\frac{f(X,Y)}{p(X,Y)}\right]$$

$$
\begin{aligned}
&= \int \frac{f(X,y)}{p(X,y)}\, p(y \,|\, X)\, dy \\
&= \int \frac{f(X,y)}{p(X,y)}\, \frac{p(X,y)}{\int p(X,y')\, dy'}\, dy \\
&= f(X) \,/\, p(X)\,.
\end{aligned}
$$

This makes the variance reduction easy to analyze. Recalling the identity

$$V[F] \;=\; E_X V_Y F + V_X E_Y F$$

from equation (2.11), and using the fact that $F' = E_Y F$, we immediately obtain

$$V[F] - V[F'] \;=\; E_X V_Y F\,.$$

This quantity is always non-negative, and represents the component of the variance of F that was caused by the random sampling of Y (as one might expect).

The use of expected values is the preferred variance reduction technique, as long as it is not too expensive to evaluate and sample the analytically integrated quantities. However, note that if expected values are used for only one part of a larger calculation, then variance can actually increase. Spanier & Gelbard [1969] give an example of this in the context of neutron transport problems, by comparing the variance of the *absorption estimator* (which records a sample only when a particle is absorbed) to that of the *collision estimator* (which records the expected value of absorption at each collision along a particle's path). They show that there are conditions where each of these estimators can have lower variance than the other.

2.5.2 Importance sampling

Importance sampling refers to the principle of choosing a density function p that is similar to the integrand f. It is a well-known fact that the best choice is to let $p(x) = c f(x)$, where the constant of proportionality is

$$c \;=\; \frac{1}{\int_\Omega f(y)\, d\mu(y)} \tag{2.22}$$

(to ensure that p integrates to one).[3] This leads to an estimator with zero variance, since

$$F = \frac{f(X)}{p(X)} = \frac{1}{c}$$

for all sample points X.

Unfortunately this technique is not practical, since we must already know the value of the desired integral in order to compute the normalization constant c. Nevertheless, by choosing a density function p whose shape is similar to f, variance can be reduced. Typically this is done by discarding or approximating some factors of f in order to obtain a function g that can be integrated analytically, and then letting $p \propto g$. It is also important to choose p such that there is a convenient method of generating samples from it. For low-dimensional integration problems, a useful strategy is to construct a discrete approximation of f (e.g. a piecewise constant or linear function). This can be done either during a separate initialization phase, or adaptively as the algorithm proceeds. The integral of such an approximation can be computed and maintained quite cheaply, and sampling can be done efficiently by means of tree structures or partial sums.

In summary, importance sampling is one of the most useful and powerful techniques of Monte Carlo integration. It is particularly helpful for integrands that have large values on a relatively small part of the domain, e.g. due to singularities.

2.5.3 Control variates

With *control variates*, the idea is to find a function g that can be integrated analytically and is similar to the integrand, and then subtract it. Effectively, the integral is rewritten as

$$I = \int_\Omega g(x)\, d\mu(x) + \int_\Omega f(x) - g(x)\, d\mu(x)\,,$$

and then sampled with an estimator of the form

$$F = \int_\Omega g(x)\, d\mu(x) + \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i) - g(X_i)}{p(X_i)}$$

[3] We assume that f is non-negative in this discussion. Otherwise the best choice is to let $p \propto |f|$, however the variance obtained this way is no longer zero [Kalos & Whitlock 1986].

where the value of the first integral is known exactly. (As usual p is the density function from which the X_i are chosen.) This estimator will have a lower variance than the basic estimator (2.12) whenever

$$V\left[\frac{f(X_i) - g(X_i)}{p(X_i)}\right] \;\le\; V\left[\frac{f(X_i)}{p(X_i)}\right] .$$

In particular, notice that if g is proportional to p, then the two estimators differ only by a constant, and their variance is the same. This implies that if g is already being used for importance sampling (up to a constant of proportionality), then it is not helpful to use it as a control variate as well.[4] From another point of view, given some function g that is an approximation to f, we must decide whether to use it as a control variate or as a density function for importance sampling. It is possible to show that either one of these choice could be the best, depending on the particular f and g. In general, if $f - g$ is nearly a constant function, then g should be used as a control variate; while if f/g is nearly constant, then g should be used for importance sampling [Kalos & Whitlock 1986].

As with importance sampling, control variates can be obtained by approximating some factors of f or by constructing a discrete approximation. Since there is no need to generate samples from g, such functions can be slightly easier to construct. However, note that for g to be useful as a control variate, it must take into account all of the significant factors of f. For example, consider an integral of the form $f(x) = f_1(x)\, f_2(x)$, and suppose that $f_1(x)$ represents the reflectivity of a surface at the point x, while $f_2(x)$ represents the incident power per unit area. Without some estimate of the magnitude of f_2, observe that f_1 is virtually useless as a control variate. On the other hand, f_1 can be used for importance sampling without any difficulties.

Control variates have had very few applications in graphics so far (e.g. see Lafortune & Willems [1995a]). One problem with the technique is the possibility of obtaining negative sample values, even for an integrand that is strictly positive. This can lead to large relative errors for integrals whose true value is close to zero (e.g. pixels in the dark regions of an image). On the other hand, the method is straightforward to apply, and can potentially give a modest variance reduction at little cost.

[4] See the discussion under Russian roulette below.

2.6 Variance reduction II: Uniform sample placement

Another important strategy for reducing variance is to ensure that samples are distributed more or less uniformly over the domain. We will examine several techniques for doing this, namely *stratified sampling*, *Latin hypercube sampling*, *orthogonal array sampling*, and *quasi-Monte Carlo methods*.

For these techniques, it is typically assumed that the domain is the s-dimensional unit cube $[0,1]^s$. Other domains can be handled by defining an appropriate transformation of the form $T : [0,1]^s \to \Omega$. Note that by choosing different mappings T, the transformed samples can be given different density functions. This makes it straightforward to apply importance sampling to the techniques described below.[5]

2.6.1 Stratified sampling

The idea of *stratified sampling* is to subdivide the domain Ω into several non-overlapping regions $\Omega_1, \ldots, \Omega_n$ such that

$$\bigcup_{i=1}^{n} \Omega_i \;=\; \Omega\,.$$

Each region Ω_i is called a *stratum*. A fixed number of samples n_i is then taken within each Ω_i, according to some given density function p_i.

For simplicity, assume that $\Omega = [0,1]^s$ and that p_i is simply the constant function on Ω_i. This leads to an estimate of the form

$$F' \;=\; \sum_{i=1}^{n} v_i\, F_i \tag{2.23}$$

$$\text{where} \qquad F_i \;=\; \frac{1}{n_i} \sum_{j=1}^{n_i} f(X_{i,j})\,. \tag{2.24}$$

Here $v_i = \mu(\Omega_i)$ is the volume of region Ω_i, and each $X_{i,j}$ is an independent sample from

[5]Note that if the desired density $p(x)$ is complex, it may be difficult to find a transformation T that generates it. This can be solved with rejection sampling, but the resulting samples will not be stratified as well.

p_i. The variance of this estimator is

$$V[F'] \;=\; \sum_{i=1}^{n} v_i^2\, \sigma_i^2 \,/\, n_i\,, \tag{2.25}$$

where $\sigma_i^2 = V[f(X_{i,j})]$ denotes the variance of f within Ω_i.

To compare this against unstratified sampling, suppose that $n_i = v_i N$, where N is the total number of samples taken. Equation (2.25) then simplifies to

$$V[F'] \;=\; \frac{1}{N} \sum_{i=1}^{n} v_i\, \sigma_i^2\,.$$

On the other hand, the variance of the corresponding unstratified estimator is[6]

$$V[F] \;=\; \frac{1}{N}\left[\sum_{i=1}^{n} v_i\, \sigma_i^2 \;+\; \sum_{i=1}^{n} v_i(\mu_i - I)^2\right], \tag{2.26}$$

where μ_i is the mean value of f in region Ω_i, and I the mean value of f over the whole domain. Since the right-hand sum is always non-negative, stratified sampling can never increase variance.

However, from (2.26) we see that variance is only reduced when the strata have different means; thus, the strata should be chosen to make these means as different as possible. Ideally, this would be achieved by stratifying the *range* of the integrand, by finding strata such that $x_i \in \Omega_i$ implies $x_1 \le x_2 \le \cdots \le x_N$.

Another point of view is to analyze the convergence rate. For functions with a bounded first derivative, the variance of stratified sampling converges at a rate of $O(N^{-1-2/s})$, while if the function is only piecewise continuous then the variance is $O(N^{-1-1/s})$ [Mitchell 1996]. (The convergence rate for the standard deviation is obtained by dividing these exponents by two.) Thus, stratified sampling can increase the convergence rate noticeably in low-dimensional domains, but has little effect in high-dimensional domains.

In summary, stratified sampling is a useful, inexpensive variance reduction technique.

[6] To obtain this result, observe that an unstratified sample in $[0,1]^s$ is equivalent to first choosing a random stratum I_j (according to the discrete probabilities v_i), and then randomly choosing X_j within Ω_{I_j}. From this point of view, X_j is chosen conditionally on I_j. This lets us apply the identity (2.11) to express the variance as a sum of two components, yielding equation (2.26).

It is mainly effective for low-dimensional integration problems where the integrand is reasonably well-behaved. If the dimension is high, or if the integrand has singularities or rapid oscillations in value (e.g. a texture with fine details), then stratified sampling will not help significantly. This is especially true for problems in graphics, where the number of samples taken for each integral is relatively small.

2.6.2 Latin hypercube sampling

Suppose that a total of N samples will be taken. The idea of *Latin hypercube sampling* is to subdivide the domain $[0,1]^s$ into N subintervals along each dimension, and to ensure that one sample lies in each subinterval. This can be done by choosing s permutations $\pi_1, \ldots, \pi_s$ of $\{1,\ldots,N\}$, and letting the sample locations be

$$X_i^j \;=\; \frac{\pi_j(i) - U_{i,j}}{N}\,, \tag{2.27}$$

where X_i^j denotes the j-th coordinate of the sample X_i, and the $U_{i,j}$ are independent and uniformly distributed on $[0,1]$. In two dimensions, the sample pattern corresponds to the occurrences of a single symbol in a *Latin square* (i.e. an $N \times N$ array of N symbols such that no symbol appears twice in the same row or column).

Latin hypercube sampling was first proposed as a Monte Carlo integration technique by McKay et al. [1979]. It is closely related to Latin square sampling methods, which have been used in the design of statistical experiments since at least the 1920's (e.g. in agricultural research [Fisher 1925, Fisher 1926]). Yates [1953] and Patterson [1954] extended these techniques to arbitrary dimensions, and also analyzed their variance-reduction properties (in the context of survey sampling and experimental design). In computer graphics, Latin square sampling was introduced by Shirley [1990a] under the name of *N-rooks sampling* [Shirley 1990a, Shirley 1991].

The first satisfactory variance analysis of Latin hypercube sampling for Monte Carlo integration was given by Stein [1987]. First, we define a function $g(x)$ to be *additive* if it has the form

$$g(x) \;=\; \sum_{j=1}^{s} g_j(x^j)\,, \tag{2.28}$$

where x^j denotes the j-th component of $x \in [0,1]^s$. Next, let f_{add} denote the best additive approximation to f, i.e. the function of the form (2.28) which minimizes the mean squared error

$$\int_\Omega (f_{\mathrm{add}}(x) - f(x))^2 \, d\mu(x) \, .$$

We can then write f as the sum of two components

$$f(x) \;=\; f_{\mathrm{add}}(x) + f_{\mathrm{res}}(x) \, ,$$

where f_{res} is orthogonal to all additive functions, i.e.

$$\int_\Omega f_{\mathrm{res}}(x) \, g(x) \, d\mu(x) \;=\; 0$$

for any additive function g.

Stein [1987] was then able to show that variance of Latin hypercube sampling is

$$V[F'] \;=\; \frac{1}{N} \int_\Omega f_{\mathrm{res}}^2(x) \, d\mu(x) \;+\; o(1/N) \, , \tag{2.29}$$

where $o(1/N)$ denotes a function that decreases faster than $1/N$. This expression should be compared to the variance using N independent samples, which is

$$V[F] \;=\; \frac{1}{N} \left(\int_\Omega f_{\mathrm{res}}^2(x) \, d\mu(x) \;+\; \int_\Omega (f_{\mathrm{add}}(x) - I)^2 \, d\mu(x) \right) \, .$$

The variance in the second case is always larger (for sufficiently large N). Thus Latin hypercube sampling improves the convergence rate for the additive component of the integrand. Furthermore, it is never significantly worse than using independent samples [Owen 1997a]:

$$V[F'] \;\le\; \frac{N}{N-1} V[F] \qquad \text{for } N \ge 2 \, .$$

Latin hypercube sampling is easy to implement and works very well for functions that are nearly additive. However, it does not work that well for image sampling, because the samples are not well-stratified in two dimensions. Except in special cases (e.g. pixels with vertical or horizontal edges), it has the same $O(1/N)$ variance that would be obtained with independent samples. This is inferior to stratified sampling, for which the variance is $O(N^{-2})$ for smooth functions and $O(N^{-3/2})$ for piecewise continuous functions.

2.6.3 Orthogonal array sampling

Orthogonal array sampling [Owen 1992, Tang 1993] is an important generalization of Latin hypercube sampling that addresses some of these deficiencies. Rather than stratifying all of the one-dimensional projections of the samples, it stratifies all of the t-dimensional projections for some $t \geq 2$. This increases the rate of convergence for the components of f that depend on t or fewer variables.

An *orthogonal array of strength* t is an $N \times s$ array of symbols, drawn from an alphabet of size b, such that every $N \times t$ submatrix contains the same number of copies of each of the b^t possible rows. (The submatrix is not necessarily contiguous; it can contain any subset of the columns.) If we let λ denote the number of times that each row appears (where λ is known as the *index* of the array), it is clear that $N = \lambda b^t$. The following table gives an example of an orthogonal array whose parameters are $OA(N, s, b, t) = (9, 4, 3, 2)$:

0	0	0	0
0	1	1	2
0	2	2	1
1	0	1	1
1	1	2	0
1	2	0	2
2	0	2	2
2	1	0	1
2	2	1	0

Various methods are known for constructing orthogonal arrays of strength $t = 2$ [Bose 1938, Bose & Bush 1952, Addelman & Kempthorne 1961], strength $t = 3$ [Bose & Bush 1952, Bush 1952], and arbitrary strengths $t \geq 3$ [Bush 1952]. Implementations of these methods are publicly available [Owen 1995a].

Let A be an $N \times s$ orthogonal array of strength t, where the symbols in the array are $\{0, 1, \ldots, b-1\}$. The first step of orthogonal array sampling is to randomize the array, by applying a permutation to the alphabet in each column. That is, we let

$$\hat{A}_{i,j} = \pi_j(A_{i,j}) \qquad \text{for all } i, j \,,$$

where $\pi_1, \ldots, \pi_s$ are random permutations of the symbols $\{0, \ldots, b-1\}$. It is easy to check that $\hat{A}$ is an orthogonal array with the same parameters (N, s, b, t) as the original array A. This step ensures that each of the b^s possible rows occurs in $\hat{A}$ with equal probability.

Now let the domain be $[0,1]^s$, and consider the family of b^s subcubes obtained by splitting each axis into b intervals of equal size. Each row of $\hat{A}$ can be interpreted as an index into this family of subcubes. The idea of orthogonal array sampling is to take one sample in each of the N subcubes specified by the rows of $\hat{A}$. Specifically, the j-th coordinate of sample X_i is

$$X_i^j \;=\; (\hat{A}_{i,j} + U_{i,j}) \,/\, b$$

where the $U_{i,j}$ are independent uniform samples on $[0,1]$. Because of the randomization step above, it is straightforward to show that each X_i is uniformly distributed in $[0,1]^s$, so that $F_N \;=\; (1/N) \sum_{i=1}^N f(X_i)$ is an unbiased estimator of the usual integral I.

To see the advantage of this technique, consider the sample distribution with respect to any t coordinate axes (i.e. project the samples into the subspace spanned by these axes). This subspace can be divided into b^t subcubes by splitting each axis into b intervals. The main property of orthogonal array sampling is that each of these subcubes contains the same number of samples. To see this, observe that the coordinates of the projected samples are specified by a particular $N \times t$ submatrix of the orthogonal array. By the definition of orthogonal arrays, each of the possible b^t rows occurs λ times in this submatrix, so that there will be exactly λ samples in each subcube.

Orthogonal array sampling is clearly a generalization of Latin hypercube sampling. Rather than stratifying the one-dimensional projections of the samples, it stratifies all of the t-dimensional projections simultaneously. (There are $\binom{s}{t}$ such projections in all.)

2.6.3.1 Analysis of variance decompositions

The variance reduction properties of orthogonal array sampling can be analyzed using *continuous analysis of variance (anova) decompositions* [Owen 1994, Owen 1992]. Our description follows [Owen 1992], which in turn is based on [Efron & Stein 1981].

Let $S = \{1, \ldots, s\}$ be the set of all coordinate indices, and let $U \subseteq S$ be any subset of these indices (there are 2^s possible subsets in all). We will use the notation x^U to refer to the set of coordinate variables x^j for $j \in U$. The *anova decomposition* of a given function f can then be written as a sum

$$f(x) \;=\; \sum_{U \subseteq S} f_U(x^U)\,, \tag{2.30}$$

where each function f_U depends only on the variables indexed by U.

The function when $U = \emptyset$ does not depend on any variables, and is called the *grand mean*:

$$I \;=\; f_\emptyset \;=\; \int_{[0,1]^s} f(x)\,dx\,.$$

The other $2^s - 1$ subsets of U are called *sources of variation*. The components of f that depend on just one variable are called the *main effects* and are defined as

$$f_j(x^j) \;=\; \int (f(x) - I) \prod_{i \neq j} dx^i\,.$$

Notice that all of these functions are orthogonal to the constant function $f_\emptyset = I$. Similarly, the *two-factor interactions* are defined by

$$f_{j,k}(x^{j,k}) \;=\; \int \Big(f(x) - I - f_j(x^j) - f_k(x^k)\Big) \prod_{i \neq j,k} dx^i$$

which represent the components of f that depend on two particular variables together. These functions are orthogonal to $f_\emptyset$ and to all the f_j.

In general, f_U is defined by

$$f_U(x^U) \;=\; \int \left(f(x) - \sum_{V \subset U} f_V(x^V)\right) dx^{S-U} \tag{2.31}$$

where the sum is over all proper subsets of U ($V \neq U$). The resulting set of functions is orthogonal, i.e. they satisfy

$$\int f_U(x^U)\, f_V(x^V)\, dx \;=\; 0$$

whenever $U \neq V$. This implies the useful property that

$$\int f^2(x)\,dx \;=\; \sum_{U \subseteq S} \int f_U^2(x^U)\,dx\,,$$

so that the variance of f can be written as

$$\int (f(x) - I)^2 \, dx \;=\; \sum_{|U|>0} \int f_U^2(x^U) \, dx \, .$$

As a particular case of this analysis, the best additive approximation to f is

$$f_{add}(x) = I + \sum_{j=1}^{s} f_j(x^j) \, ,$$

where the residual $f_{res} = f - f_{add}$ is orthogonal to all additive functions. The variance of Latin hypercube sampling can thus be rewritten as

$$\sigma^2_{\text{LH}} \;=\; \frac{1}{N} \sum_{|U|>1} \int f_U^2(x^U) \, dx \;+\; o(1/N) \, ,$$

i.e. the single-variable components of the variance converge at a rate faster than $1/N$.

Orthogonal array sampling generalizes this result; it is possible to show that the variance is [Owen 1992, Owen 1994]

$$\sigma^2_{\text{OA}} \;=\; \frac{1}{N} \sum_{|U|>t} \int f_U^2(x^U) \, dx \;+\; o(1/N) \, ,$$

i.e. the convergence rate is improved with respect to all components of the integrand that depend on t coordinates or less.

The case $t = 2$ is particularly interesting for graphics. For example, if we apply this technique to distribution ray tracing, it ensures that all the two dimensional projections are well stratified (over the pixel, lens aperture, light source, etc). This achieves a similar result to the sampling technique proposed by Cook et al. [1984], except that all combinations of two variables are stratified (including combinations such as the pixel x-coordinate and the aperture x-coordinate, for example).

2.6.3.2 Orthogonal array-based Latin hypercube sampling

Notice that because the t-dimensional margins are well-stratified, the w-dimensional margins are also stratified for any $w < t$. However, the resulting stratification is not as good. For example, in any one-dimensional projectional there will be exactly λb^{t-1} samples in each interval of width $1/b$. This is inferior to Latin hypercube sampling, which places one sample in each interval of width $1/(\lambda b^t)$.

There is a simple modification to orthogonal array sampling that yields the same one-dimensional stratification properties as Latin hypercube sampling. (The result, logically enough, is called *orthogonal array-based Latin hypercube sampling* [Tang 1993].) The idea is to remap the λb^t symbols within each column into a single sequence $\{0, 1, \ldots, \lambda b^t - 1\}$, by mapping the λb^{t-1} identical copies of each symbol m into a random permutation of the symbols

$$\lambda b^{t-1} m, \ \ldots, \ \lambda b^{t-1}(m+1) - 1 \, .$$

This process is repeated for each column separately. Letting $\hat{A}'$ be the modified array, the sample locations are then defined as

$$X_i^j = \frac{\hat{A}'_{i,j} + U_{i,j}}{\lambda b^t} \, .$$

This ensures that the samples are maximally stratified for each one-dimensional projection, as well as for each t-dimensional projection. It is possible to show that this leads to a further reduction in variance [Tang 1993].

This technique is similar to *multi-jittered sampling* [Chiu et al. 1994], which corresponds to the special case where $s = 2$ and $t = 2$.

2.6.4 Quasi-Monte Carlo methods

Quasi-Monte Carlo methods take these ideas a step further, by dispensing with randomness completely. The idea is to distribute the samples as uniformly as possible, by choosing their locations deterministically.

2.6.4.1 Discrepancy

Let $P = \{x_1, \ldots, x_N\}$ be a set of points in $[0, 1]^s$. Typically, the goal of quasi-Monte Carlo methods is minimize the *irregularity of distribution* of the samples with respect to some quantitative measure. One such measure is the *star discrepancy* of P. Let $\mathcal{B}^*$ denote the set of all axis-aligned boxes with one corner at the origin:

$$\mathcal{B}^* \;=\; \{B = [0, u_1] \times \cdots \times [0, u_s] \mid 0 \le u_i \le 1 \text{ for all } i\}\,.$$

Ideally, we would like each box B to contain exactly $\lambda(B)N$ of the points in P, where $\lambda(B) = u_1 \cdots u_s$ is the volume of B. The star discrepancy simply measures how much P deviates from this ideal situation:

$$D^*_N(P) \;=\; \sup_{B \in \mathcal{B}^*} \left| \frac{\#\{P \cap B\}}{N} - \lambda(B) \right|, \tag{2.32}$$

where $\#\{P \cap B\}$ denotes the number of points of P that are inside the box B.

Discrepancy measures can also be defined with respect to other sets of shapes (e.g. arbitrary axis aligned boxes, or convex regions [Niederreiter 1992]). For two-dimensional image sampling, it is particularly useful to measure discrepancy with respect to *edges*, by considering the family of shapes obtained by intersecting $[0, 1]^2$ with an arbitrary half-plane [Mitchell 1992]. The relevance of discrepancy to image sampling was first pointed out by Shirley [1991].

The significance of the star discrepancy is that it is closely related to bounds on the integration error. Specifically, the *Koksma-Hlawka inequality* states that

$$\left| \frac{1}{N} \sum_{i=1}^{N} f(x_i) \;-\; \int_{[0,1]^s} f(x)\, dx \right| \;\le\; V_{HK}(f)\, D^*_N(P)\,,$$

where $V_{HK}(f)$ is the *variation of f in the sense of Hardy and Krause* [Niederreiter 1992]. Thus, the maximum integration error is directly proportional to the discrepancy, provided that the variation $V_{HK}(f)$ is finite. By finding low-discrepancy points sets and sequences, we can ensure that the integration error is small.

It is important to note that for dimensions $s \ge 2$, the variation $V_{HK}(f)$ is infinite whenever f is discontinuous.[7] This severely limits the usefulness of these bounds in computer graphics, where discontinuities are common. Also note that since $V_{HK}(f)$ is typically

[7]More precisely, $V_{HK}(f) = \infty$ whenever f is discontinuous along a surface that is not perpendicular to one of the s coordinate axes. In general, note that f must be at least s times differentiable in order for $V_{HK}(f)$ to be bounded in terms of the partial derivatives of f. That is, letting M be an upper bound on the magnitude of all partial derivatives of degree at most s, then $V_{HK}(f) \le cM$ where the constant c depends only on s [Niederreiter 1992].

harder to evaluate than the original integral, these worst-case bounds are not useful for estimating or bounding the error in practice.

2.6.4.2 Low-discrepancy points sets and sequences

A *low-discrepancy sequence* is an infinite sequence of points $x_1, x_2, \ldots$ such that the star discrepancy is

$$D_N^*(P) \;=\; O\left(\frac{(\log N)^s}{N}\right)$$

for any prefix $P = \{x_1, \ldots, x_N\}$. (Note that P is actually a multiset, i.e. the multiplicity of the elements matters.) This result is achieved by a number of known constructions, and it is widely believed to be the best possible [Niederreiter 1992]. However, it should be noted that the best current lower bound for an arbitrary dimension s is only

$$D_N^*(P) \;\geq\; C(s) \cdot \frac{(\log N)^{s/2}}{N}\,,$$

i.e. there is a significant gap between these bounds.

If we drop the requirement that P is a prefix of an infinite sequence, the discrepancy can be improved slightly. A *low-discrepancy point set* is defined to be a multiset $P = \{x_1, \ldots, x_N\}$ for which

$$D_N^*(P) \;=\; O\left(\frac{(\log N)^{s-1}}{N}\right)\,.$$

(More precisely, this should be the definition of a low-discrepancy point set *construction*, since the bound does not make sense when applied to a single point set P.)

Combining these bounds with the Koksma-Hlawka inequality, the error of quasi-Monte Carlo integration is at most $O((\log N)^{s-1}/N)$ using a low-discrepancy point set, or $O((\log N)^s/N)$ using a prefix of a low-discrepancy sequence.

Note that these bounds are of questionable value unless N is very large, since $(\log N)^s$ is much larger than N for typical values of N and s. In particular, notice that the function $(\log N)^s/N$ is monotonically *increasing* for $N < e^s$ (i.e. the larger the sample size, the worse the error bound). In fact, we should not expect these error bounds to be meaningful until $(\log N)^s < N$ at the very least, since otherwise the error bound is worse than it would be for $N = 2$. To get an idea of how large N must be, consider the case $s = 6$. It is easy to check that $(\log N)^s/N > (\log 2)^s/2$ for all $N < 10^9$, and thus we should not expect meaningful error bounds until N is substantially larger than this.

However, these error bounds are overly pessimistic in practice. Low-discrepancy sequences often give better results than standard Monte Carlo even when N is fairly small, provided that the integrand is reasonably well behaved.

2.6.4.3 Halton sequences and Hammersley points

We now discuss several well-known constructions for low-discrepancy points sets and sequences. In one dimension, the *radical inverse sequence* $x_i = \phi_b(i)$ is obtained by first writing the base-b expansion of i:

$$i = \sum_{k \geq 0} d_{i,k} b^k ,$$

and then reflecting the digits around the decimal point:

$$\phi_b(i) = \sum_{k \geq 0} d_{i,k}\, b^{-1-k} .$$

The special case when $b = 2$ is called the *van der Corput sequence*,

$$\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{5}{8}, \frac{3}{8}, \ldots .$$

The discrepancy of the radical-inverse sequence is $O((\log N)/N)$ in any base b (although the implied constant increases with b).

To obtain a low-discrepancy sequence in several dimensions, we use a different radical inverse sequence in each dimension:

$$x_i = (\phi_{b_1}(i), \phi_{b_2}(i), \ldots, \phi_{b_s}(i))$$

where the bases b_i are all relatively prime. The classic example of this construction is the *Halton sequence*, where the b_i are chosen to be the first s primes:

$$x_i = (\phi_2(i), \phi_3(i), \phi_5(i), \ldots, \phi_{p_s}(i)) .$$

The Halton sequence has a discrepancy of $O((\log N)^s/N)$.

If the number of sample points N is known in advance, this discrepancy can be improved slightly by using equally spaced points i/N in the first dimension. The result is known as the *Hammersley point set*:

$$x_i = (i/N, \phi_2(i), \phi_3(i), \ldots, \phi_{p_{s-1}}(i))$$

where p_i denotes the i-th prime as before. The discrepancy of the Hammersley point set is $O((\log N)^{s-1}/N)$.

2.6.4.4 $(\mathbf{t}, \mathbf{m}, \mathbf{s})$-nets and $(\mathbf{t}, \mathbf{s})$-sequences

Although discrepancy is a useful measure of the irregularity of distribution of a set of points, it does not always accurately predict which sequences will work best for numerical integration. Recently there has been a great deal of interest in (t, m, s)-nets and (t, s)-sequences, which define the irregularity of distribution in a slightly different way. Let E be an *elementary interval in the base* b, which is simply an axis-aligned box of the form

$$E = \prod_{j=1}^{s} \left[\frac{t_j}{b^{k_j}}, \frac{t_j + 1}{b^{k_j}} \right)$$

where the exponents $k_j \geq 0$ are integers, and $0 \leq t_j \leq b^{k_j} - 1$. In other words, each dimension of the box must be a non-positive power of b, and the box must be aligned to an integer multiple of its size in each dimension. The volume of an elementary interval is clearly

$$\lambda(E) = b^{-\sum_{j=1}^{s} k_j} .$$

A $(0, m, s)$*-net in base* b is now defined to be a point set P of size $N = b^m$, such that every elementary interval of volume $1/b^{-m}$ contains exactly one point of P. This implies that a $(0, m, s)$-net is distributed as evenly as possible with respect to such intervals. For example, suppose that P is $(0, 4, 2)$-net in base 5. Then P would contain $N = 625$ points in the unit square $[0, 1]^2$, such that every elementary interval of size $1 \times 1/625$ contains a point of P. Similarly, all the intervals of size $1/5 \times 1/125$, $1/25 \times 1/25$, $1/125 \times 1/5$, and $1/625 \times 1$ would contain exactly one point of P.

The more general notion of a (t, m, s)-*net* is obtained by relaxing this definition somewhat. Rather than requiring every box of size b^{-m} to contain exactly one point, we require every box of size b^{t-m} to contain exactly b^t points. Clearly, smaller values of t are better. The reason for allowing $t > 0$ is to facilitate the construction of such sequences for more values of b and s. (In particular, $(0, m, s)$-nets for $m \geq 2$ can only exist when $s \leq b + 1$ [Niederreiter 1992].)

A (t, s)-*sequence* is then defined to be an infinite sequence $x_1, x_2, \ldots$ such that for all $m \geq 0$ and $k \geq 0$, the subsequence

$$x_{kb^m+1}, \ \ldots, \ x_{kb^{m+1}}$$

is a (t, m, s)-net in the base b. In particular, every prefix $x_1, \ldots, x_N$ of size $N = b^m$ is a (t, m, s)-net. Explicit constructions of such sequences for various values of b and s have been proposed by Sobol', Faure, Niederreiter, and Tezuka (see Niederreiter [1992] and Tezuka [1995]).

Every (t, s)-sequence is a low-discrepancy sequence, and every (t, m, s)-net is a low-discrepancy points set (provided that t is held fixed while m is increased). Thus these constructions have the same worst-case integration bounds as for the Halton sequences and Hammersley points. However, note that (t, s)-sequences and (t, m, s)-nets often work much better in practice, because the discrepancy is lower by a significant constant factor [Niederreiter 1992].

It is interesting to compare the equidistribution properties of (t, m, s)-nets to orthogonal array sampling. For simplicity let $t = 0$, and let A be an orthogonal array of strength m. Then in the terminology of (t, m, s)-nets, orthogonal array sampling ensures that there is one sample in each elementary interval E of volume $1/b^m$, where E has m sides of length $1/b$ and all other sides of length one. The Latin hypercube extension of Tang [1993] ensures that in addition, there is one sample in each elementary interval E that has one side of length $1/b^m$ and all other of length one. Thus the 1- and m-dimensional projections are maximally stratified. For comparison, the $(0, m, s)$-net not only achieves both of these properties, it also ensures that there is one sample in every other kind of elementary interval of volume $1/b^m$, so that the projections of dimension $2, 3, \ldots, t - 1$ are also stratified as well as possible.

2.6.4.5 Randomly permuted $(\mathrm{t}, \mathrm{m}, \mathrm{s})$-nets and (t, s)-sequences

A significant disadvantage of quasi-Monte Carlo methods is that the sample locations are deterministic. In computer graphics, this leads to significant aliasing artifacts [Mitchell 1992]. It also makes it difficult to compute error estimates, since unlike with Monte Carlo methods we cannot simply take several independent samples.

These difficulties can be resolved by using *randomly permuted* (t, m, s)*-nets and* (t, s)*-sequences* [Owen 1995b] (also called *scrambled nets and sequences*). These are obtained by applying random permutations to the digits of ordinary (t, m, s)-nets and (t, s)-sequences, in such a way that their equidistribution properties are preserved [Owen 1995b]. The idea is straightforward to implement, although its analysis is more involved.

Scrambled nets have several advantages. Most importantly, the resulting estimators are unbiased, since the sample points are uniformly distributed over the domain $[0, 1]^s$. This makes it possible to obtain unbiased error estimates by taking several independent random samples (e.g. using different digit permutations of the same original (t, m, s)-net). (See Owen [1997a] for additional discussion of variance estimates.) In the context of computer graphics, scrambled nets also provide a way to eliminate the systematic aliasing artifacts typically encountered with quasi-Monte Carlo integration.

Second, it is possible to show that for smooth functions, scrambled nets lead to a variance of

$$V[\hat{I}] \;=\; O\left(\frac{(\log N)^{s-1}}{N^3}\right) ,$$

and thus an expected error of $O((\log N)^{(s-1)/2} N^{-3/2})$ in probability [Owen 1997b]. This is an improvement over both the Monte Carlo rate of $O(N^{-1/2})$ and the quasi-Monte Carlo rate of $O((\log N)^{s-1} N^{-1})$. In all cases, these bounds apply to a worst-case function f (of sufficient smoothness), but note that the quasi-Monte Carlo rate uses a deterministic set of points while the other bounds are averages over random choices made by the sampling algorithm.

Scrambled nets can improve the variance over ordinary Monte Carlo even when the function f is not smooth [Owen 1997b]. With respect to the analysis of variance decomposition described above, scrambled nets provide the greatest improvement on the components f_U where the number of variables $|U|$ is small. These functions f_U can be smooth even when f itself is not (due to integration over the variables in $S - U$), leading to fast convergence on these components.

2.6.4.6 Discussion

The convergence rates of quasi-Monte Carlo methods are rarely meaningful in computer graphics, due to smoothness requirements on the integrand and the relatively small sample sizes that are typically used. Other problems include the difficulty of estimating the variation $V_{HK}(f)$, and the fact that $(\log N)^{s-1}$ is typically much larger than N in practice. The lack of randomness in quasi-Monte Carlo methods is a distinct disadvantage, since it causes aliasing and precludes error estimation.

Hybrids of Monte Carlo and quasi-Monte Carlo seem promising, such as the scrambled (t, m, s)-nets described above. Although such methods do not necessarily work any better than standard Monte Carlo for discontinuous integrands, at least they are not worse. In particular, they do not introduce aliasing artifacts, and error estimates are available.

Keller [1996, 1997] has applied quasi-Monte Carlo methods to the radiosity problem (a special case of the light transport problem where all surfaces are diffuse). He uses a particle-tracing algorithm (similar to Pattanaik & Mudur [1993]), except that the directions for scattering are determined by a Halton sequence. He has reported a convergence rate that is slightly better than standard Monte Carlo on simple test scenes. The main benefit appears to be due to the sampling of the first four dimensions of each random walk (which control the selection of the initial point on a light source and the direction of emission).

2.7 Variance reduction III: Adaptive sample placement

A third family of variance reduction methods is based on the idea of adaptively controlling the sample density, in order to place more samples where they are most useful (e.g. where the integrand is large or changes rapidly). We discuss two different approaches to doing this. One is *adaptive sampling*, which can introduce bias unless special precautions are taken. The other approach consists of two closely related techniques called *Russian roulette* and *splitting*, which do not introduce bias and are especially useful for light transport problems.

2.7.1 Adaptive sampling

The idea of *adaptive sampling* (also called *sequential sampling*) is to take more samples where the integrand has the most variation. This is done by examining the samples that have been taken so far, and using this information to control the placement of future samples. Typically this involves computing the variance of the samples in a given region, which is then refined by taking more samples if the variance exceeds a given threshold. A number of such techniques have been proposed in graphics for image sampling (for example, see Lee et al. [1985], Purgathofer [1986], Kajiya [1986], [Mitchell 1987], Painter & Sloan [1989]).

Like importance sampling, the goal of adaptive sampling is to concentrate samples where they will do the most good. However, there are two important differences. First, importance sampling attempts to place more samples in regions where the integrand is large, while adaptive sampling attempts to places more samples where the variance is large. (Of course, with adaptive sampling we are free to use other criteria as well.) A second important difference is that with adaptive sampling, the sample density is changed "on the fly" rather than using *a priori* information.

The main disadvantage of adaptive sampling is that it can introduce bias, which in turn can lead to image artifacts. Bias can be avoided using *two-stage sampling* [Kirk & Arvo 1991], which consists of first drawing a small sample of size n from a representative region $R \subset \Omega$, and then using this information to determine the sample size N for the remaining portion $\Omega - R$ of the domain.[8] Although this technique eliminates bias, it also eliminates some of the advantages of adaptive sampling, since it cannot react to unusual samples encountered during the second stage of sampling.

Another problem with adaptive sampling is that it is not very effective for high-dimensional problems. The same problems are encountered as with stratified sampling: there are too many possible dimensions to refine. For example, if we split the region to be refined into two pieces along each axis, there will be 2^s new regions to sample. If most of the sampling error is due to variation along only one or two of these axes, the refinement will be very inefficient.

[8] Alternatively, two samples of size n and N could be drawn over the entire domain, where the first sample is used only to determine the value of N and is then discarded.

2.7.2 Russian roulette and splitting

Russian roulette and *splitting* are two closely related techniques that are often used in particle transport problems. Their purpose is to decrease the sample density where the integrand is small, and increase it where the integrand is large. Unlike adaptive sampling, however, these techniques do not introduce any bias. The applications of these methods in computer graphics have been described by Arvo & Kirk [1990].

Russian roulette. Russian roulette is usually applied to estimators that are a sum of many terms:

$$F \;=\; F_1 + \cdots + F_N \,.$$

For example, F might represent the radiance reflected from a surface along a particular viewing ray, and each F_i might represent the contribution of a particular light source.

The problem with this type of estimator is that typically most of the contributions are very small, and yet all of the F_i are equally expensive to evaluate. The basic idea of Russian roulette is to randomly skip most of the evaluations associated with small contributions, by replacing these F_i with new estimators of the form

$$F_i' \;=\; \begin{cases} \frac{1}{q_i} F_i & \text{with probability } q_i \,, \\ 0 & \text{otherwise} \,. \end{cases}$$

The evaluation probability q_i is chosen for each F_i separately, based on some convenient estimate of its contribution. Notice that the estimator F_i' is unbiased whenever F_i is, since

$$\begin{aligned} E[F_i'] &= q_i \cdot \frac{1}{q_i} E[F_i] + (1 - q_i) \cdot 0 \\ &= E[F_i] \,. \end{aligned}$$

Obviously this technique increases variance; it is basically the inverse of the *expected values* method described earlier. Nevertheless, Russian roulette can still increase efficiency, by reducing the average time required to evaluate F.

For example, suppose that each F_i represents the contribution of a particular light source to the radiance reflected from a surface. To reduce the number of visibility tests using Russian roulette, we first compute a tentative contribution t_i for each F_i by assuming that the light source is fully visible. Then a fixed threshold δ is typically chosen, and the probabilities q_i are set to

$$q_i \;=\; \min(1, t_i \,/\, \delta)\,.$$

Thus contributions larger than δ are always evaluated, while smaller contributions are randomly skipped in a way that does not cause bias.

Russian roulette is also used to terminate the random walks that occur particle transport calculations. (This was the original purpose of the method, as introduced by Kahn — see [Hammersley & Handscomb 1964, p. 99].) Similar to the previous example, the idea is to randomly terminate the walks whose estimated contributions are relatively small. That is, given the current walk $\mathbf{x}_0\mathbf{x}_1\cdots\mathbf{x}_k$, the probability of extending it is chosen to be proportional to the estimated contribution that would be obtained by extending the path further, i.e. the contribution of paths of the form $\mathbf{x}_0\cdots\mathbf{x}_{k'}$ where $k' > k$. This has the effect of terminating walks that have entered unproductive regions of the domain. In computer graphics, this technique is used extensively in ray tracing and Monte Carlo light transport calculations.

Splitting. Russian roulette is closely related to *splitting*, a technique in which an estimator F_i is replaced by one of the form

$$F_i' \;=\; \frac{1}{k}\sum_{j=1}^{k} F_{i,j}\,,$$

where the $F_{i,j}$ are independent samples from F_i. As with Russian roulette, the splitting factor k is chosen based on the estimated contribution of the sample F_i. (A larger estimated contribution generally corresponds to a larger value of k.) It is easy to verify that this transformation is unbiased, i.e.

$$E[F_i'] \;=\; E[F_i]\,.$$

In the context of particle transport calculations, this has the effect of splitting a single particle into k new particles which follow independent paths. Each particle is assigned a weight that is a fraction $1/k$ of the weight of the original particle. Typically this technique is applied when a particle enters a high-contribution region of the domain, e.g. if we are trying to measure leakage through a reactor shield, then splitting might be applied to neutrons that have already penetrated most of the way through the shield.

The basic idea behind both of these techniques is the same: given the current state $\mathbf{x}_0\mathbf{x}_1 \cdots \mathbf{x}_k$ of a random walk, we are free to use any function of this state in deciding how many samples of $\mathbf{x}_{k+1}$ will be taken. If we predict that the contribution of the path $\mathbf{x}_0 \cdots \mathbf{x}_{k+1}$ will be low, then most of the time we will take no samples at all; while if the contribution is high, we may decide to take several independent samples. If this is applied at every vertex, the resulting structure is a tree of paths.

In general, Russian roulette and splitting can be applied to any process where each sample is determined by a sequence of random steps. We can use any prefix of this sequence to estimate the importance of the final sample. This is then used to decide whether the current state should be discarded (if the importance is low) or replicated (if the importance is high). Although this idea is superficially similar to adaptive sampling, it does not introduce any bias.

Russian roulette is an indispensable technique in transport calculations, since it allows otherwise infinite random walks to be terminated without bias. Splitting is also useful if it is judiciously applied [Arvo & Kirk 1990]. In combination, these techniques can be very effective at directing sampling effort into the most productive regions of the domain.

2.8 Variance reduction IV: Correlated estimators

The last family of variance reduction methods we will discuss is based on the idea of finding two or more estimators whose values are correlated. So far these methods have not found significant uses in graphics, so our discussion will be brief.

2.8.1 Antithetic variates

The idea of *antithetic variates* is to find two estimators F_1 and F_2 whose values are negatively correlated, and add them. For example, suppose that the desired integral is $\int_0^1 f(x)\,dx$, and consider the estimator

$$F \;=\; (f(U) + f(1-U))\,/\,2$$

where U is uniformly distributed on $[0, 1]$. If the function f is monotonically increasing (or monotonically decreasing), then $f(U)$ and $f(1 - U)$ will be negatively correlated, so that F will have lower variance than if the two samples were independent [Rubinstein 1981, p. 135]. Furthermore, the estimator F is exact whenever the integrand is a linear function (i.e. $f(x) = ax + b$).

This idea can be easily adapted to the domain $[0, 1]^s$, by considering pairs of sample points of the form

$$X_1 = (U_1, \ldots, U_s) \qquad \text{and} \qquad X_2 = (1 - U_1, \ldots, 1 - U_s)\,.$$

Again, this strategy is exact for linear integrands. If more than two samples are desired, the domain can be subdivided into several rectangular regions Ω_i, and a pair of samples of the form above can be taken in each region.

Antithetic variates of this type are most useful for smooth integrands, where f is approximately linear on each subregion Ω_i. For many graphics problems, on the other hand, variance is mainly due to discontinuities and singularities of the integrand. These contributions tend to overwhelm any variance improvements on the smooth regions of the integrand, so that antithetic variates are of limited usefulness.

2.8.2 Regression methods

Regression methods are a more advanced way to take advantage of several correlated estimators. Suppose that we are given several unbiased estimators F_1, …, F_n for the desired quantity I, and that the F_i are correlated in some way (e.g. because they use different transformations of the same random numbers, as in the antithetic variates example). The idea is to take several samples from each estimator, and apply standard linear regression techniques in order to determine the best estimate for I that takes all sources of correlation into account.

Specifically, the technique works by taking N samples from each estimator (where the j-th samples from F_i is denoted $F_{i,j}$). We then compute the sample means

$$\hat{I}_i = \frac{1}{N} \sum_{j=1}^{N} F_{i,j} \quad \text{for } i = 1, \ldots, n\,,$$

and the sampling variance-covariance matrix $\hat{\mathbf{V}}$, a square $n \times n$ array whose entries are

$$\hat{V}_{i,j} \;=\; \frac{1}{N-1} \sum_{k=1}^{N} (F_{i,k} - \hat{I}_i)\,(F_{j,k} - \hat{I}_j)\,.$$

The final estimate F is then given by

$$F \;=\; (\mathbf{X}^* \, \hat{\mathbf{V}}^{-1} \, \mathbf{X})^{-1} \, \mathbf{X}^* \, \hat{\mathbf{V}}^{-1} \, \hat{\mathbf{I}}\,, \tag{2.33}$$

where $\mathbf{X}^*$ denotes the transpose of $\mathbf{X}$, $\mathbf{X} \;=\; [1 \ldots 1]^*$ is a column vector of length n, and $\hat{\mathbf{I}} = [\hat{I}_1 \ldots \hat{I}_n]^*$ is the column vector of sample means. Equation (2.33) is the standard minimum-variance unbiased linear estimator of the desired mean I, except that we have replaced the true variance-covariance matrix $\mathbf{V}$ by an approximation $\hat{\mathbf{V}}$. Further details can be found in Hammersley & Handscomb [1964].

Note that this technique introduces some bias, due to the fact that the same random samples are used to estimate both the sample means $\hat{I}_i$ and the variance-covariance matrix entries $\hat{V}_{i,j}$ (which are used to weight the $\hat{I}_i$). This bias could be avoided by using different random samples for these two purposes (of course, this would increase the cost).

The main problem with regression methods is in finding a suitable set of correlated estimators. If the integrand has discontinuities or singularities, then simple transformations of the form $f(U)$ and $f(1-U)$ will not produce a significant amount of correlation. Another problem is that this method requires that a substantial number of samples be taken, in order to estimate the covariance matrix with any reasonable accuracy.