George S. Fishman

Monte Carlo

Concepts, Algorithms, and Applications

With 98 Illustrations

Springer

2

Estimating Volume and Count

This chapter introduces the reader to fundamental issues that arise when applying the Monte Carlo method to solving a commonly encountered problem in numerical computation. In its most basic form the problem is to evaluate the volume of a bounded region in multi-dimensional euclidean space. The more general problem is to evaluate the integral of a function on such a region. The Monte Carlo method often offers a competitive and sometimes the only useful solution to the problem. The appeal of the Monte Carlo method arises when the shape of the region of interest makes solution by analytical methods impossible and, in the case of the more general function integration, when little is known about the smoothness and variational properties of the integrand, or what is known precludes the applications of alternative numerical evaluation techniques.

Evaluating count represents an equally important basic problem of numerical computation. Circumstances often arise where one needs to know the number (count) of subsets of objects, belonging to a set of size m ($< \infty$), that exhibit a specified property. For many of these problems, the count increases exponentially in m. Therefore, even if an efficient procedure exists for generating each individual subset of objects and determining whether or not it exhibits the property, the potential exponential growth with m severely limits one's ability to enumerate all subsets and compute an exact count. This limitation prompts one to resort to the Monte Carlo method for a solution.

Although volume applies to continuous regions and count applies to discrete objects, this chapter emphasizes that each merely represents different facets of the same fundamental problem. Accordingly, a lesson learned about volume estima-

tion regularly carries over to count estimation, and vice versa. For simplicity of exposition, we often use the word volume to include count as well.

The present chapter focuses on issues related to point estimation and interval estimation in the form of a confidence interval. A *point estimator* denotes a mathematical expression into which one enters data derived from a sampling experiment to produce a single number, called a *point estimate*, that serves as an approximation to the true, but unknown, value of the volume or, more generally, integral of interest. An *interval estimator* or *confidence interval* denotes an expression into which one enters the data to produce a numerical interval that, with a user-specified probability, covers the true value of the unknown volume. In practice, a confidence interval provides an assessment of how representative the point estimate is of the true value of the volume.

A procedural problem that arises in virtually every Monte Carlo sampling study concerns the number of independent trials or replications that must be performed in order to guarantee a specified bound on error. This number, called the *sample size*, depends on the tolerable error specification and on the peculiar features of the problem under study. Based on a relatively limited knowledge of these features, one often can compute, prior to sampling, a *worst-case sample size* which guarantees that, if one performs this number of trials, the resulting point estimate will meet the error specification. Although the worst-case sample sizes generally is larger than what is actually needed, its availability prior to sampling often can influence the Monte Carlo user's strategy in a constructive way. If computing time is not a key consideration, he or she may choose to collect this sample size, being assured of the desired error bound for the resulting point estimate with no need for a supplemental error analysis. Conversely, if computing time is at a premium, a priori worst-case sample sizes for several experiments, all of which must be run, provide the experimenter with guidance as to where to aim at improving computational efficiency.

Sometimes one also can compute a *best-case sample size* which, based on available knowledge, is the minimal number of observations that can conceivably guarantee the error specification. The ratio of worst-case to best-case sample sizes indicates how excessive taking the worst-case sample size may actually be. While a ratio close to unity encourages one to use the worst-case approach, an experimenter with a limited computing budget may be considerably more circumspect when encountering a substantially larger ratio.

Improving computational efficiency is a constant theme throughout this book. The present chapter introduces the topic by describing how a user of the Monte Carlo method can exploit information available prior to sampling to modify the sampling plan in a way that produces an estimate of volume with specified accuracy at less cost than is possible if the prior information is not used. Techniques for effecting this reduction are called *variance-reducing methods*, and worst-case variances, available for alternative sampling plans prior to experimentation, provide useful information about the relative desirability of each plan. However, the term variance-reducing methods is a misnomer. In reality, the cost of implementing a

sampling plan and the variance it induces collectively dictate its appeal relative to an alternative plan. To combat the misplaced emphasis, we call this topic *efficiency-improving techniques*. It is the availability of efficiency-improving techniques that distinguishes the Monte Carlo method from the simple-minded sampling experiments that historically preceded it. The present chapter introduces several basic concepts for devising efficiency-improving techniques, and Ch. 4 gives a comprehensive account of these concepts.

2.1 Volume

Let $\mathscr{R}$ denote a region of unknown volume $\lambda(\mathscr{R})$ in the m-dimensional unit hypercube $\mathscr{I}^m = [0,1]^m = [0,1] \times \cdots \times [0,1]$. If the region of interest is arbitrary in size, we assume that a suitable transformation can be made to map it into $\mathscr{I}^m$. Suppose that a series of inequalities and implicit relationships among the spatial variables $0 \leq x_i \leq 1$, $i = 1, \ldots, m$, defines the shape of $\mathscr{R}$ in a way that makes the exact computation of $\lambda(\mathscr{R})$ intractable. Assuming that a procedure exists for generating a sequence of points $\mathscr{X}_{m,n} = \{\mathbf{x}^{(j)} = (x_1^{(j)}, \ldots, x_m^{(j)}) \in \mathscr{I}^m; j = 1, \ldots, n\}$, one can approximate $\lambda(\mathscr{R})$ by $\bar{\lambda}_n(\mathscr{R})$ by performing the steps in Algorithm VOLUME.

The numerical accuracy of $\bar{\lambda}_n(\mathscr{R})$ depends on the properties of $\mathscr{R}$ and $\mathscr{X}_{m,n}$. For example, suppose that one generates evaluation points from the n-point m-dimensional mesh or hypercubical lattice

$$\mathscr{X}_{m,n} = \{\mathbf{x} = (x_1, \ldots, x_m): x_i = (z_i + 1/2)/k; z_i = 0, 1, \ldots, k-1; i = 1, \ldots, m, n = k^m\}. \tag{1}$$

Algorithm VOLUME

Purpose: To compute an approximation to $\lambda(\mathscr{R})$.
Input: Region $\mathscr{R}$ in $\mathscr{I}^m$ and n = number of evaluation points.
Output: $\bar{\lambda}_n(\mathscr{R})$.
Method:

1. $j \leftarrow 1$ and $S \leftarrow 0$.
2. While $j \leq n$:
   a. Generate $\mathbf{x}^{(j)}$ from $\mathscr{X}_{m,n}$.
   b. $\phi(\mathbf{x}^{(j)}) \leftarrow 0$.
   c. If $\mathbf{x}^{(j)} \in \mathscr{R}$, $\phi(\mathbf{x}^{(j)}) \leftarrow 1$.
   d. $S \leftarrow S + \phi(\mathbf{x}^{(j)})$.
   e. $j \leftarrow j + 1$.
3. Compute $\bar{\lambda}_n(\mathscr{R}) = S/n$ as an approximation to $\lambda(\mathscr{R})$.

Then each of the n-points in $\mathscr{I}^m$ is the center of an m-dimensional hypercube of volume $1/k^m = 1/n$ so that S/n, the total volume of the S m-cubes with centers in $\mathscr{R}$, is the approximation to $\lambda(\mathscr{R})$. Figure 2.1 illustrates this concept for $m = 2$. Observe that the shaded region in Fig. 2.1(b) has area S/n and that the summation of all striped and dotted regions in Fig. 2.1(c) represents the error $\bar{\lambda}_n(\mathscr{R}) - \lambda(\mathscr{R})$ encountered in approximating $\lambda(\mathscr{R})$ by this method. Let $s(\mathscr{R})$ denote the length of the boundary of the two-dimensional region $\mathscr{R}$. Then the absolute error has the upper bound

$$|\bar{\lambda}_n(\mathscr{R}) - \lambda(\mathscr{R})| \leq s(\mathscr{R})/k.$$

In effect, one lays out $s(\mathscr{R})$ as the maximal length and $1/k$ as the maximal width of a rectangle containing the errors on the boundary and uses the resulting area $s(\mathscr{R})/k$ as the worst-case error that can arise.

More generally, if $s(\mathscr{R})$ denotes the surface area of the region $\mathscr{R}$, then the corresponding worst-case error is

$$|\bar{\lambda}_n(\mathscr{R}) - \lambda(\mathscr{R})| \leq s(\mathscr{R})/k. \tag{2}$$

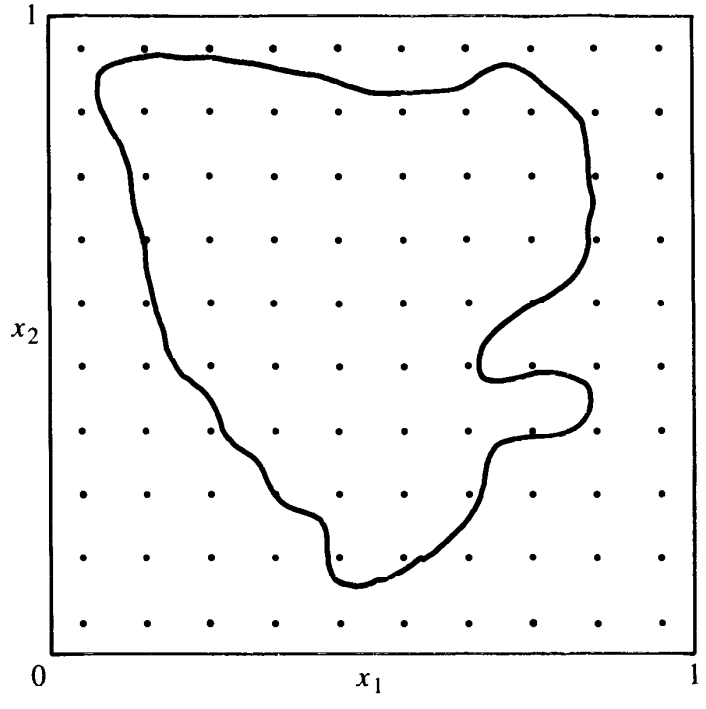
Since $n = k^m$, expression (2) has the alternative form

$$|\bar{\lambda}_n(\mathscr{R}) - \lambda(\mathscr{R})| \leq s(\mathscr{R})/n^{1/m}, \tag{3}$$

revealing a convergence rate that diminishes with increasing m for a given n. To guarantee an absolute error no larger than a fixed $\varepsilon \in (0, 1)$ requires an m-mesh with

$$n(\varepsilon) = \lceil [s(\mathscr{R})/\varepsilon]^m \rceil$$

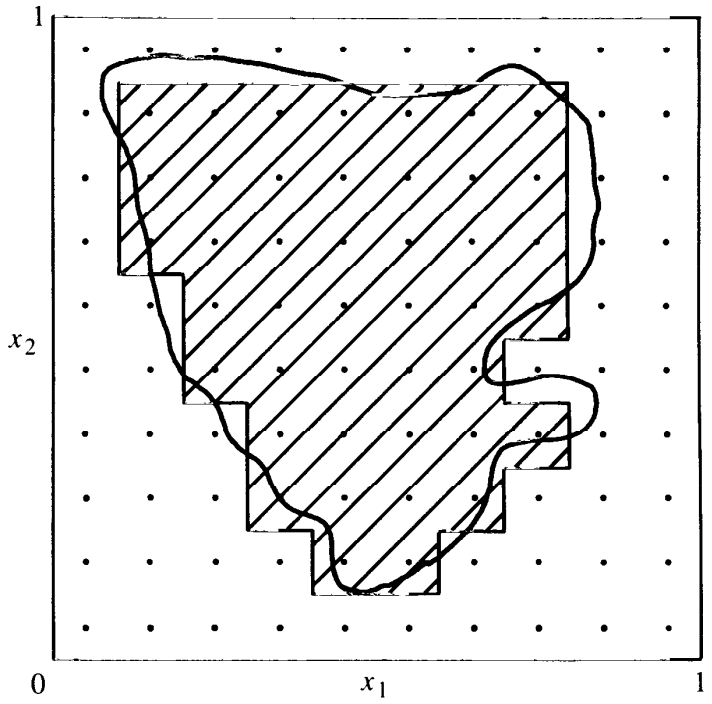
points, where $\lceil \theta \rceil$ denotes the smallest integer greater than or equal to θ.



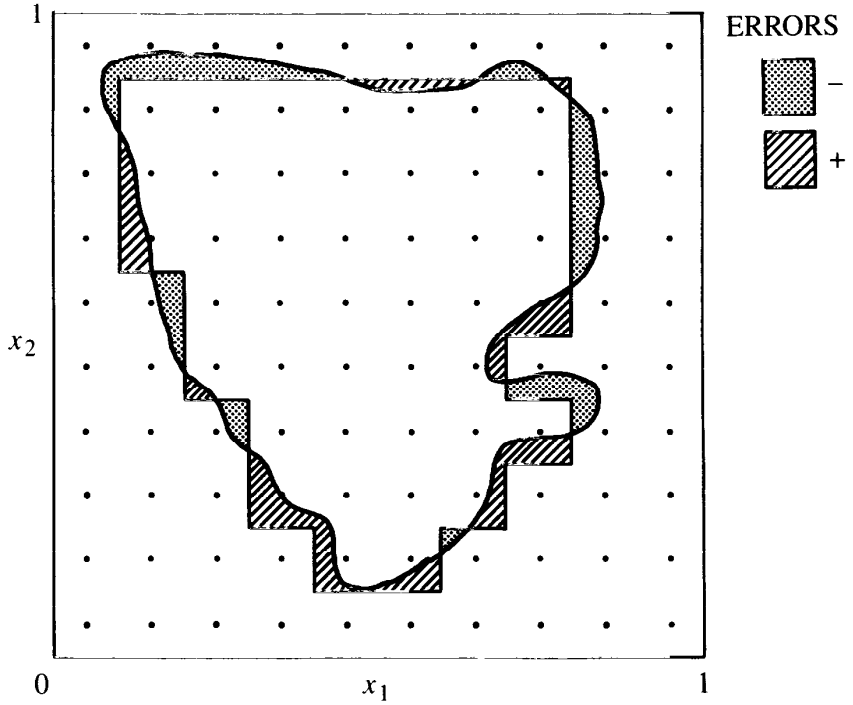
(a)

Figure 2.1. Region $\mathscr{R}$ in the Unit Square

Figure 2.1 (cont.)



(b)



(c)

To illustrate the significance of this result, suppose that the objective is to approximate the volumes of a sequence of regions $\mathscr{R}_2, \mathscr{R}_3, \ldots$ of increasing dimension m but all of which have the same surface area. Then, to achieve the same worst-case absolute error ε for $\mathscr{R}_m$ takes $n(m, \varepsilon) = O(\varepsilon^{-m})$ points in the m-mesh in equation (1) as $m \to \infty$, where $O\lceil u(z) \rceil$ denotes a function $\{v(z)\}$ for which there exist constants $c > 0$ and z_0 such that $v(z) \leq cu(z)$ for all $z \geq z_0$. In the present case, $z = m$, $v(z) = n(m, \varepsilon)$, and $u(z) = \varepsilon^{-m}$. This exponential growth in number of mesh points reveals the serious limitation of this deterministic approach for moderate and large m.

Generating each evaluation $\mathbf{x}^{(j)}$ in step 2a and evaluating $\phi(\mathbf{x}^{(j)})$ in step 2c account for the principal work in executing Algorithm VOLUME. Generating each point usually takes $O(m^{\alpha})$ time as m increases, with $\alpha \geq 1$. Each evaluation in step 2c requires one to determine if $\mathbf{x}^{(j)} \in \mathscr{R}$, work which customarily takes $O(m^{\beta})$ time with $\beta \geq 1$. Therefore, apart from the surface area, the total work required to achieve an absolute error ε takes $O(m^{\gamma}\varepsilon^{-m})$ time in the worst case where $\gamma = \max(\alpha, \beta)$, severely limiting one's capacity to approximate $\lambda(\mathscr{R})$ in this way as m increases.

The most elementary form of the Monte Carlo method solves this approximation problem using Algorithm VOLUME with one modification. It replaces the deterministic points $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$ by a sequence of independent random samples $\{\mathbf{X}^{(j)} = (X_1^{(j)}, \ldots, X_m^{(j)}); j = 1, \ldots, n\}$ each drawn from the uniform distribution on $\mathscr{S}^m$. If $\mathbf{X} = (X_1, \ldots, X_m)$ is such a point, then X_i has the probability density function (p.d.f.)

$$f(x) = \begin{cases} 1, & 0 \leq x < 1 \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

denoted by $\mathscr{U}(0, 1)$, and $\mathbf{X}$ has the p.d.f.

$$f(\mathbf{x}) = \begin{cases} 1, & 0 \leq x_i \leq 1 \quad \text{for } i = 1, \ldots, m \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Algorithm MC describes the modification. Hereafter, we refer to this approach as standard Monte Carlo sampling.

As before, $\bar{\lambda}_n(\mathscr{R})$ is an approximation to $\lambda(\mathscr{R})$; but now $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(n)}$ are each independent random samples drawn from the p.d.f. in equation (5). Observe that $\phi(\mathbf{X}^{(1)}), \ldots, \phi(\mathbf{X}^{(n)})$ are independent Bernoulli random variables with

$$\text{pr}[\phi(\mathbf{X}^{(j)}) = 1] = \int_{\mathscr{R}} f(\mathbf{x})\, d\mathbf{x} = \lambda(\mathscr{R})$$

and

$$\text{pr}[\phi(\mathbf{X}^{(j)}) = 0] = \int_{\mathscr{S}^m \setminus \mathscr{R}} f(\mathbf{x})\, d\mathbf{x} = 1 - \lambda(\mathscr{R}) \qquad j = 1, \ldots, n.$$

Algorithm MC

Purpose: To estimate $\lambda(\mathscr{R})$.

Input: Region $\mathscr{R}$ in $\mathscr{S}^m$, sample size n, and confidence level $1 - \delta$.

Output: Unbiased point estimates for $\lambda(\mathscr{R})$ and $\operatorname{var}\bar{\lambda}_n(\mathscr{R})$ and $100 \times (1 - \delta)$ percent confidence interval for $\lambda(\mathscr{R})$.

Method:

1. $j \leftarrow 1$ and $S \leftarrow 0$.
2. While $j \leq n$:
   a. $i \leftarrow 1$. While $i \leq m$: sample $X_i^{(j)}$ from $\mathscr{U}(0, 1)$; $i \leftarrow i + 1$.
   b. $\phi(\mathbf{X}^{(j)}) \leftarrow 0$.
   c. If $\mathbf{X}^{(j)} \in \mathscr{R}$, $\phi(\mathbf{X}^{(j)}) \leftarrow 1$.
   d. $S \leftarrow S + \phi(\mathbf{X}^{(j)})$.
   e. $j \leftarrow j + 1$.
3. Compute summary statistics
   a. $\bar{\lambda}_n(\mathscr{R}) \leftarrow S/n$ as a point estimate of $\lambda(\mathscr{R})$.
   b. $V[\bar{\lambda}_n(\mathscr{R})] \leftarrow (S/n)(1 - S/n)/(n - 1)$ as a point estimate of $\operatorname{var}\bar{\lambda}_n(\mathscr{R})$.
   c. $[I_1(S, n, \delta), I_2(S, n, \delta)]$ as a $100 \times (1 - \delta)$ percent confidence interval for $\lambda(\mathscr{R})$.

Then, $S = \phi(\mathbf{X}^{(1)}) + \cdots + \phi(\mathbf{X}^{(n)})$ has the binomial distribution

$$\operatorname{pr}(S = i) = f_i(n, \lambda) = \binom{n}{i} \lambda^i (1 - \lambda)^{n-i} \qquad \lambda = \lambda(\mathscr{R}) \qquad i = 0, 1, \ldots, n,$$

denoted by $\mathscr{B}(n, \lambda)$, with $ES = n\lambda$ and $\operatorname{var} S = n\lambda(1 - \lambda)$. As a result, $\bar{\lambda}_n = \bar{\lambda}_n(\mathscr{R})$ is an unbiased estimator of λ with

$$\operatorname{var}\bar{\lambda}_n = \frac{1}{n^2} \operatorname{var} S = \lambda(1 - \lambda)/n. \tag{6}$$

Since λ is unknown, the value of this expression lies merely in knowing its form.

2.2 Error and Sample Size Considerations

As in the deterministic case, the error in $\bar{\lambda}_n$ in Algorithm MC decreases as n increases. However, the term error now focuses on statistical sampling variation rather than on mathematical approximation. In particular, it is known that

$$\operatorname{pr}\left(\lim_{n \to \infty} \bar{\lambda}_n = \lambda\right) = 1,$$

denoted equivalently as

$$\lim_{n\to\infty} \bar{\lambda}_n = \lambda \quad \text{w.p.1}$$

$$\lim_{n\to\infty} \bar{\lambda}_n = \lambda \quad \text{a.s.}$$

or

$$\lim_{n\to\infty} \bar{\lambda}_n = \lambda \quad \text{a.e,}$$

where w.p.1, a.s., and a.e. denote, respectively, *with probability* 1, *almost surely*, and *almost everywhere*. This property represents the strongest form of probabilistic convergence and when it holds for an estimator, as, for example, for $\bar{\lambda}_n$ in Algorithm MC, we say that $\bar{\lambda}_n$ is a *strongly consistent estimator* of λ. While a most desirable limiting property, convergence w.p.1 does not address the practical case of assessing error for finite n. This section begins a discussion of this issue.

In contrast to the deterministic method, the Monte Carlo method based on independent trials enables one to use the sample data $\phi(\mathbf{X}^{(1)}), \ldots, \phi(\mathbf{X}^{(n)})$ to assess the accuracy of the point estimate $\bar{\lambda}_n$. One way is through estimation of equation (6). Since

$$\begin{aligned} \mathrm{E}[(S/n)(1 - S/n)] &= \frac{1}{n}\mathrm{E}S - \frac{1}{n^2}(\operatorname{var} S + \mathrm{E}^2 S) \\ &= \lambda(1 - \lambda)(n - 1)/n, \end{aligned}$$

$V(\bar{\lambda}_n)$, computed in step 3b, is an unbiased estimator of $\operatorname{var} \bar{\lambda}_n$.

The quantity $\sqrt{V(\bar{\lambda}_n)}$, called the *standard error* of $\bar{\lambda}_n$, sometimes serves as a rough measure of the statistical error in $\bar{\lambda}_n$. However, the fact that $V(\bar{\lambda}_n)$ is itself an estimate introduces a source of sampling error which limits its value in practice. Observe that (Ex. 1)

$$\lim_{n\to\infty} \operatorname{corr}[\bar{\lambda}_n, V(\bar{\lambda}_n)] = \begin{cases} 1, & \text{if } 0 \le \lambda < 1/2 \\ 0, & \text{if } \lambda = 1/2 \\ -1, & \text{if } 1/2 < \lambda \le 1, \end{cases} \tag{7}$$

implying that, for $\lambda < 1/2$, a $V(\bar{\lambda}_n)$ greater (less) than $\operatorname{var} \bar{\lambda}_n$ tends to accompany a $\bar{\lambda}_n$ greater (less) than λ. This behavior leads one to regard $\bar{\lambda}_n$ as less (more) accurate than is actually true. Conversely, $\lambda > 1/2$ implies that a $V(\bar{\lambda}_n)$ greater (less) than $\operatorname{var} \bar{\lambda}_n$ tends to accompany a $\bar{\lambda}_n$ less (greater) than λ, again misleading one about the accuracy of $\bar{\lambda}_n$. Most disturbingly, this correlation does not vanish with increased sample size. These observations encourage one to regard $V(\bar{\lambda}_n)$ and $\sqrt{V(\bar{\lambda}_n)}$ as merely giving rough assessments of error.

The confidence interval $[I_1(S, n, \delta), I_2(S, n, \delta)]$ in step 3c of Algorithm MC provides a considerably more meaningful way to assess the accuracy of $\bar{\lambda}_n$. Several alternative methods exist for computing a confidence interval, each based on differ-

ent aspects of statistical theory. Moreover, this same theory lays the foundation for the determination of the sample size needed to obtain an estimate that meets a specified error criterion. Since it is somewhat easier from a pedagogic viewpoint to describe these sample size considerations first and confidence interval considerations second, we adopt this approach here.

Chebyshev's inequality provides the basis for the first method of error assessment.

Theorem 2.1 (Chebyshev's Inequality). *Let Z denote a random variable with distribution function F defined on $(-\infty, \infty)$, $EZ = 0$, and $\sigma^2 = \operatorname{var} Z = EZ^2 < \infty$. Then, for $\beta > 0$,*

$$\operatorname{pr}\left(\frac{|Z|}{\sigma} \geq \beta\right) \leq 1/\beta^2. \tag{8}$$

PROOF. Observe that, for $\varepsilon > 0$,

$$\begin{aligned}\operatorname{pr}(|Z| \geq \varepsilon) &= \int_{-\infty}^{-\varepsilon} dF(z) + \int_{\varepsilon}^{\infty} dF(z) \\ &\leq \int_{-\infty}^{-\varepsilon} \frac{z^2}{\varepsilon^2} dF(z) + \int_{\varepsilon}^{\infty} \frac{z^2}{\varepsilon^2} dF(z) \\ &\leq \frac{1}{\varepsilon^2} \int_{-\infty}^{\infty} z^2 \, dF(z) = \sigma^2/\varepsilon^2.\end{aligned}$$

Putting $\beta = \varepsilon/\sigma$ establishes equation (8). □

In the present case, $Z = S/n - \lambda$ and $\sigma^2 = \lambda(1 - \lambda)/n$, so that

$$\operatorname{pr}(|\bar{\lambda}_n - \lambda| < \varepsilon) \geq 1 - \lambda(1 - \lambda)/n\varepsilon^2. \tag{9}$$

Expression (9) implies

$$\lim_{n \to \infty} \operatorname{pr}(|\bar{\lambda}_n - \lambda| \geq \varepsilon) = 0,$$

which is called *convergence in probability*. Convergence w.p.1 implies convergence in probability, but the reverse is not true. Therefore, convergence in probability is the weaker property. Nevertheless, expression (9) provides a convenient first basis for assessing sample size requirements.

In contrast to the deterministic method, specifying ε in expression (9) alone does not suffice to determine the smallest n that guarantees an error no larger than ε. To account for randomness, one must also specify a *confidence level* $1 - \delta$ with $0 < \delta < 1$ so that by Chebyshev's inequality all samples sizes n greater than or equal to

$$n_C(\varepsilon, \delta, \lambda) = \lceil \lambda(1 - \lambda)/\delta\varepsilon^2 \rceil \tag{10}$$

satisfy the error specification $\operatorname{pr}[|\bar{\lambda}_n - \lambda| < \varepsilon] \geq 1 - \delta$. We call this specification

the (ε, δ) *absolute error criterion.* Since λ is unknown and since $\lambda(1-\lambda) \leq 1/4$, this criterion, when applied to expression (9), leads to the *worst-case sample size*

$$n_C(\varepsilon, \delta) = \lceil 1/4\delta\varepsilon^2 \rceil \tag{11}$$

for all $\lambda \in [0, 1]$.

Expression (11) reveals one of the most appealing features of the Monte Carlo method, namely the invariance of $n_C(\varepsilon, \delta)$ with respect to the dimension m. Moreover, $n_C(\varepsilon, \delta, \lambda)$ in equation (10) depends on $\mathscr{R}$ merely through its volume $\lambda(\mathscr{R})$. When compared to the deterministic m-mesh in expression (1), the Monte Carlo method becomes increasingly favored as the dimenion m increases for $m > 2$. Since pure random sampling in step 2a takes $O(m)$ and evaluation in step 2c takes $O(m^\beta)$ time with $\beta \geq 1$, achieving this (ε, δ) absolute error criterion takes $O[m^\beta \lambda(1-\lambda)/\delta\varepsilon^2] = O(m^\beta/4\delta\varepsilon^2)$ time. Observe that this time is polynomial in m, in contrast to Algorithm VOLUME, which has an exponential bound in m.

Although Chebyshev's inequality provides expression (11) as a definitive guide to sample size, it customarily specifies a larger sample size than is necessary. This property motivates a search for alternative approaches that achieve the (ε, δ) absolute error criterion with a smaller sample size. It is well known that as n increases the standarized quantity $(S - n\lambda)/[n\lambda(1-\lambda)]^{1/2}$ has a probability distribution function that converges (as $n \to \infty$) to the standard normal distribution function

$$\Phi(z) = (2\pi)^{-1/2} \int_{-\infty}^{z} e^{-y^2/2}\, dy \qquad -\infty < z < \infty,$$

with 0 mean and unit variance and denoted by $\mathscr{N}(0, 1)$. Equivalently, one says that $(S - n\lambda)/\sqrt{n\lambda(1-\lambda)}$ converges to a random variable from $\mathscr{N}(0, 1)$. Proof of this classical *Central Limit Theorem* appears in many textbooks. For example, see Feller (1968, Ch. 5). Let

$$n_N(\varepsilon, \delta, \lambda) = \lceil \lambda(1-\lambda)[\Phi^{-1}(1 - \delta/2)/\varepsilon]^2 \rceil \tag{12a}$$

where

$$\Phi^{-1}(\theta) = \inf\left[z: (2\pi)^{-1/2} \int_{-\infty}^{z} e^{-y^2/2}\, dy = \theta, 0 < \theta < 1 \right].$$

Then, as $\varepsilon \to 0$, taking the sample size to be expression (12a) guarantees the (ε, δ) absolute error

$$\lim_{\varepsilon \to 0} \text{pr}[|S/n_N(\varepsilon, \delta, \lambda) - \lambda|/\varepsilon \leq 1] = 1 - \delta.$$

The corresponding worst-case normal sample size is

$$n_N(\varepsilon, \delta) = \lceil [\Phi^{-1}(1 - \delta/2)/2\varepsilon]^2 \rceil. \tag{12b}$$

Column 1 of Table 2.1 lists the ratio $1/\delta[\Phi^{-1}(1 - \delta/2)]^2$ for $\delta = .001$, $.01$, and $.05$. It reveals the substantially greater sample size that Chebyshev's inequality requires and encourages one to use the normal result [expression (12b)] when ε is