Acta Numerica (1998), pp. 1–49

Monte Carlo and quasi-Monte Carlo methods

Russel E. Caflisch*
Mathematics Department, UCLA,
Los Angeles, CA 90095-1555, USA
E-mail: caflisch@math.ucla.edu

Monte Carlo is one of the most versatile and widely used numerical methods. Its convergence rate, $O(N^{-1/2})$, is independent of dimension, which shows Monte Carlo to be very robust but also slow. This article presents an introduction to Monte Carlo methods for integration problems, including convergence theory, sampling methods and variance reduction techniques. Accelerated convergence for Monte Carlo quadrature is attained using quasi-random (also called low-discrepancy) sequences, which are a deterministic alternative to random or pseudo-random sequences. The points in a quasi-random sequence are correlated to provide greater uniformity. The resulting quadrature method, called quasi-Monte Carlo, has a convergence rate of approximately $O((\log N)^k N^{-1})$. For quasi-Monte Carlo, both theoretical error estimates and practical limitations are presented. Although the emphasis in this article is on integration, Monte Carlo simulation of rarefied gas dynamics is also discussed. In the limit of small mean free path (that is, the fluid dynamic limit), Monte Carlo loses its effectiveness because the collisional distance is much less than the fluid dynamic length scale. Computational examples are presented throughout the text to illustrate the theory. A number of open problems are described.

* Research supported in part by the Army Research Office under grant number DAAH04-95-1-0155.

CONTENTS

1. Introduction

Monte Carlo provides a direct method for performing simulation and integration. Because it is simple and direct, Monte Carlo is easy to use. It is also robust, since its accuracy depends on only the crudest measure of the complexity of the problem. For example, Monte Carlo integration converges at a rate $O(N^{-1/2})$ that is independent of the dimension of the integral. For this reason, Monte Carlo is the only viable method for a wide range of high-dimensional problems, ranging from atomic physics to finance.

The price for its robustness is that Monte Carlo can be extremely slow. The order $O(N^{-1/2})$ convergence rate is decelerating, since an additional factor of 4 increase in computational effort only provides an additional factor of 2 improvement in accuracy. The result of this combination of ease of use, wide range of applicability and slow convergence is that an enormous amount of computer time is spent on Monte Carlo computations.

This represents a great opportunity for researchers in computational science. Even modest improvements in the Monte Carlo method can have substantial impact on the efficiency and range of applicability for Monte Carlo methods. Indeed, much of the effort in the development of Monte Carlo has been in construction of variance reduction methods which speed up the computation. A description of some of the most common variance reduction methods is given in Section 4.

Variance reduction methods accelerate the convergence rate by reducing the constant in front of the $O(N^{-1/2})$ for Monte Carlo methods using random or pseudo-random sequences. An alternative approach to acceleration is to change the choice of sequence. Quasi-Monte Carlo methods use quasi-random (also known as low-discrepancy) sequences instead of random or pseudo-random. Unlike pseudo-random sequences, quasi-random sequences do not attempt to imitate the behaviour of random sequences. Instead, the elements of a quasi-random sequence are correlated to make them more uniform than random sequences. For this reason, Monte Carlo integration using quasi-random points converges more rapidly, at a rate $O(N^{-1}(\log N)^k)$, for some constant k. Quasi-random sequences are described in Sections 5 and 6.

In spite of their importance in applications, Monte Carlo methods receive relatively little attention from numerical analysts and applied mathematicians. Instead, it is number theorists and statisticians who design the pseudo-random, quasi-random and other types of sequence that are used in Monte Carlo, while the innovations in Monte Carlo techniques are developed mainly by practitioners, including physicists, systems engineers and statisticians.

The reasons for the near neglect of Monte Carlo in numerical analysis and applied mathematics are related to its robustness. First, Monte Carlo methods require less sophisticated mathematics than other numerical methods. Finite difference and finite element methods, for example, require careful mathematical analysis because of possible stability problems, but stability is not an issue for Monte Carlo. Instead, Monte Carlo nearly always gives an answer that is qualitatively correct, but acceleration (error reduction) is always needed. Second, Monte Carlo methods are often phrased in non-mathematical terms. In rarefied gas dynamics, for example, Monte Carlo allows for direct simulation of the dynamics of the gas of particles, as described in Section 7. Finally, it is often difficult to obtain definitive results on Monte Carlo, because of the random noise. Thus computational improvements often come more from experience than from a particular insightful calculation.

This article is intended to provide an introduction to Monte Carlo methods for numerical analysts and applied mathematicians. In spite of the reasons cited above, there are ample opportunities for this community to make significant contributions to Monte Carlo. First of all, any improvements can have a big impact, because of the prevalence of Monte Carlo computations. Second, the methodology of numerical analysis and applied mathematics, including well controlled computational experiments on canonical problems, is needed for Monte Carlo. Finally, there are some outstanding problems on which a numerical analysis or applied mathematics viewpoint is clearly needed; for example:

- design of Monte Carlo simulation for transport problems in the diffusion limit (Section 7)
- formulation of a quasi-Monte Carlo method for the Metropolis algorithm (Section 6)
- explanation of why quasi-Monte Carlo behaves like standard Monte Carlo when the dimension is large and the number of simulation is of moderate size (Section 6).

Some older, but still very good, general references on Monte Carlo are Kalos and Whitlock (1986) and Hammersley and Handscomb (1965).

The focus of this article is on Monte Carlo for integration problems. Integration problems are simply stated, but they can be extremely challenging. In addition, integration problems contain most of the difficulties that are found in more general Monte Carlo computations, such as simulation and optimization.

The next section formulates the Monte Carlo method for integration and describes its convergence. Section 3 describes random number generators and sampling methods. Variance reduction methods are discussed in Section 4 and quasi-Monte Carlo methods in Section 5. Effective use of quasi-Monte Carlo requires some modification of standard Monte Carlo techniques, as described in Section 6. Monte Carlo methods for rarefied gas dynamics are described in Section 7, with emphasis on the loss of effectiveness for Monte Carlo in the fluid dynamic limit.

2. Monte Carlo integration

The integral of a Lebesgue integrable function $f(x)$ can be expressed as the average or *expectation* of the function f evaluated at a random location. Consider an integral on the one-dimensional unit interval

$$I[f] = \int_0^1 f(x)\,\mathrm{d}x = \bar{f}. \tag{2.1}$$

Let x be a random variable that is uniformly distributed on the unit interval. Then

$$I[f] = E[f(x)]. \tag{2.2}$$

For an integral on the unit cube $I^d = [0,1]^d$ in d dimensions,

$$I[f] = E[f(\boldsymbol{x})] = \int_{I^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \tag{2.3}$$

in which $\boldsymbol{x}$ is a uniformly distributed vector in the unit cube.

The Monte Carlo quadrature formula is based on the probabilistic interpretation of an integral. Consider a sequence $\{x_n\}$ sampled from the uniform distribution. Then an empirical approximation to the expectation is

$$I_N[f] = \frac{1}{N}\sum_{n=1}^{N} f(x_n). \tag{2.4}$$

According to the Strong Law of Large Numbers (Feller 1971), this approximation is convergent with probability one; that is,

$$\lim_{N\to\infty} I_N[f] \to I[f]. \tag{2.5}$$

In addition, it is unbiased, which means that the average of $I_N[f]$ is exactly $I[f]$ for any N; that is,

$$E[I_N[f]] = I[f], \tag{2.6}$$

in which the average is over the choice of the points $\{x_n\}$.

In general, define the Monte Carlo integration error

$$\epsilon_N[f] = I[f] - I_N[f] \tag{2.7}$$

so that the bias is $E[\epsilon_N[f]]$ and the root mean square error (RMSE) is

$$E[\epsilon_N[f]^2]^{1/2}. \tag{2.8}$$

2.1. Accuracy of Monte Carlo

The Central Limit Theorem (CLT) (Feller 1971) describes the size and statistical properties of Monte Carlo integration errors.

Theorem 2.1 For N large,

$$\epsilon_N[f] \approx \sigma N^{-1/2} \nu \tag{2.9}$$

in which ν is a standard normal ($N(0,1)$) random variable and the constant $\sigma = \sigma[f]$ is the square root of the variance of f; that is,

$$\sigma[f] = \left(\int_{I^d} (f(x) - I[f])^2 \, \mathrm{d}x \right)^{1/2}. \tag{2.10}$$

A more precise statement is that

$$\begin{aligned} \lim_{N \to \infty} \mathrm{Prob}\left(a < \frac{\sqrt{N}}{\sigma} \epsilon_N < b \right) &= \mathrm{Prob}(a < \nu < b) \\ &= \int_a^b (2\pi)^{-1/2} \mathrm{e}^{-x^2/2} \, \mathrm{d}x. \end{aligned} \tag{2.11}$$

This says that the error in Monte Carlo integration is of size $O(N^{-1/2})$ with a constant that is just the variance of the integrand f. Moreover, the statistical distribution of the error is approximately a normal random variable. In contrast to the usual results of numerical analysis, this is a probabilistic result. It does not provide an absolute upper bound on the error; rather it says that the error is of a certain size with some probability. On the other hand, this result is an equality, so that the bounds it provides are tight. The use of this result will be discussed at the end of this section.

Now we present a partial proof of the Central Limit Theorem, which proves that the error size is $O(N^{-1/2})$. Derivation of the Gaussian distribution for the error is more difficult (Feller 1971). First define $\xi_i = \sigma^{-1}(f(x_i) - \bar{f})$ for x_i uniformly distributed. Then

$$\begin{aligned} E[\xi_i] &= 0, \\ E[\xi_i^2] &= \int \sigma^{-2}(f(x_i) - \bar{f})^2 \, \mathrm{d}x = 1, \\ E[\xi_i \xi_j] &= 0 \quad \text{if } i \neq j. \end{aligned} \tag{2.12}$$

The last equality is due to the independence of the x_i's.

Now consider the sum

$$S_N = N^{-1}\sum_1^N \xi_i = \sigma^{-1}\varepsilon_N. \tag{2.13}$$

Its variance is

$$\begin{aligned}
E[S_N^2]^{1/2} &= E[N^{-2}\left(\sum_{i=1}^N \xi_i\right)^2]^{1/2}\\
&= N^{-1}\left\{E\left[\sum_{i=1}^N \xi_i^2\right] + E\left[\sum_{i=1}^N\sum_{j\neq i}\xi_i\xi_j\right]\right\}^{1/2}\\
&= N^{-1}\left\{\sum_{i=1}^N 1+0\right\}^{1/2}\\
&= N^{-1/2}.
\end{aligned} \tag{2.14}$$

Therefore

$$E[\varepsilon_N^2] = \sigma N^{-1/2}, \tag{2.15}$$

which shows that the RMSE is of size $O(\sigma N^{-1/2})$.

The converse of the Central Limit Theorem is useful for determining the size of N required for a particular computation. Since the error bound from the CLT is probabilistic, the precision of the Monte Carlo integration method can only be ensured within some confidence level. To ensure an error of size at most ϵ with confidence level c requires the number of sample points N to be

$$N = \epsilon^{-2}\sigma^2 s(c), \tag{2.16}$$

in which s is the confidence function for a normal variable; that is,

$$\begin{aligned}
c &= \int_{-s(c)}^{s(c)} \mathrm{e}^{-x^2/2}\,\mathrm{d}x/\sqrt{2\pi}\\
&= \operatorname{erf}(s(c)/\sqrt{2}).
\end{aligned} \tag{2.17}$$

For example, 95 per cent confidence in the error size requires that $s = 2$, approximately.

In an application, the exact value of the variance is unknown (it is as difficult to compute as the integral itself), so the formula (2.16) cannot be directly used. There is an easy way around this, which is to determine the empirical error and variance (Hogg and Craig 1995). Perform M computations using independent points x_i for $1 \le i \le MN$. For each j obtain values

$I_N^{(j)}$ for $1 \leq j \leq M$. The empirical RMSE is then $\tilde{\varepsilon}_N$, given by

$$\tilde{\varepsilon}_N = \left(M^{-1} \sum_{j=1}^{M} \left(I_N^{(j)} - \bar{I}_N \right)^2 \right)^{1/2}, \tag{2.18}$$

in which

$$\bar{I}_N = M^{-1} \sum_{j=1}^{M} I_N^{(j)}. \tag{2.19}$$

The empirical variance is $\tilde{\sigma}$ given by

$$\tilde{\sigma} = N^{1/2} \tilde{\varepsilon}_N. \tag{2.20}$$

This value can be used for σ in (2.16) to determine the value of N for a given precision level ε and a given confidence level c.

2.2. Comparison to grid-based methods

Most people who see Monte Carlo for the first time are surprised that it is a viable method. How can a random array be better than a grid? There are several ways to answer this question. First, compare the convergence rate of Monte Carlo with that of a grid-based integration method such as Simpson's rule. The convergence rate for grid-based quadrature is $O(N^{-k/d})$ for an order k method in dimension d, since the grid with N points in the unit cube has spacing $N^{-1/d}$. On the other hand, the Monte Carlo convergence rate is $O(N^{-1/2})$ independent of dimension. So Monte Carlo beats a grid in high-dimension d, if

$$k/d < 1/2. \tag{2.21}$$

On the other hand, for an analytic function on a periodic domain, the value of k is infinite, so that this simple explanation fails. A more realistic explanation for the robustness of Monte Carlo is that it is practically impossible to lay down a grid in high dimension. The simplest cubic grid in d dimensions requires at least 2^d points. For $d = 20$, which is not particularly large, this requires more than a million points. Moreover, it is practically impossible to refine a grid in a high dimension, since a refinement requires increasing the number of points by factor 2^d. In contrast to these difficulties for a grid in high dimension, the accuracy of Monte Carlo quadrature is nearly independent of dimension and each additional point added to the Monte Carlo quadrature formula provides an incremental improvement in its accuracy. To be sure, the value of N at which the $O(N^{-1/2})$ error estimate becomes valid (that is, the length of the transient) is difficult to predict, but experience shows that, for problems of moderate complexity in moderate dimension (for instance $d = 20$), the $O(N^{-1/2})$ error size is typically attained for moderate values of N.

Two additional interpretations of Monte Carlo quadrature are worthwhile. Consider the Fourier representation of a periodic function with period one

$$f(x) = \sum_{k=-\infty}^{\infty} \hat{f}(k) e^{2\pi i k x}. \tag{2.22}$$

The integral $I[f]$ is just $\hat{f}(0)$; that is, the contributions to the integral are 0 from all wave-numbers $k \neq 0$. For a grid of spacing $1/n$, the grid-based quadrature formula is

$$I_n^{(g)}[f] = n^{-1} \sum_{i=1}^{n} f(i/n). \tag{2.23}$$

The contributions to this sum are 0 (as they should be) from wave-numbers $k \neq mn$ for some integer m, and the contributions are $\hat{f}(k)$ for wave-numbers $k = mn$. That is, the accuracy of grid-based quadrature is 100 per cent for $k \neq mn$, but 0 per cent for $k = mn$ (with $m \neq 0$). Monte Carlo quadrature using a random array is partially accurate for all k, which is superior to a grid-based method if the Fourier coefficients decay slowly.

Finally, insight into the relative performance of grid-based and Monte Carlo methods is gained by considering the points themselves in a high dimension. For a regular grid, the change from one point to the next is only a variation of one component at a time, that is, $(0,0,\ldots,0,0) \mapsto (0,0,\ldots,0,1/n)$. In many problems, this is an inefficient use of the unvaried components. In a random array, all components are varied in each point, so that the state space is sampled more fully. This accords well with the global nature of Monte Carlo: each point of a Monte Carlo integration formula is an estimate of the integral over the entire domain.

3. Generation and sampling methods

3.1. Random number generators

The numbers generated by computers are not random, but pseudo-random, which means that they are made to have many of the properties of random number sequences (Niederreiter 1992). While this is a well-developed subject, occasional problems still occur, mostly with very long sequences ($N \geq 10^9$). The methods used to generate pseudo-random numbers are mostly linear congruential methods. There is a series of reliable pseudo-random number generators in the popular book *Numerical Recipes* (Press, Teukolsky, Vettering and Flannery 1992). It is important to use the routines *ran0, ran1, ran2, ran3, ran4* from the *second* edition (for instance, *ran1* is recommended for $N < 10^8$); the routines *RAN0, RAN1, RAN2, RAN3* from the first edition of this book had some deficiencies (see Press and Teukolsky (1992)).

For very large problems requiring extremely large values of N (as high as 10^{13}!), reliable sequences can be obtained from the project Scalable Parallel Random Number Generators Library for Parallel Monte Carlo Computations (`http://www.ncsa.uiuc.edu/Apps/SPRNG/`).

3.2. Sampling methods

Standard (pseudo-) random number generators produce uniformly distributed variables. Non-uniform variables can be sampled through transformation of uniform variables. For a non-uniform random variable with density $p(x)$, the expectation of a function $f(x)$ is

$$E[f] = I[f] = \int f(x)p(x)\,\mathrm{d}x. \tag{3.1}$$

For a sequence of random numbers $\{x_n\}$ distributed according to the density p, the empirical approximation to the expectation is

$$I_N[f] = \frac{1}{N}\sum_{n=1}^{N} f(x_n) \tag{3.2}$$

and the resulting quadrature error is

$$\epsilon_N[f] = I[f] - I_N[f]. \tag{3.3}$$

As in the one-dimensional case, the Central Limit Theorem says that

$$\epsilon_N[f] \approx N^{-1/2}\sigma\nu \tag{3.4}$$

in which ν is $N(0,1)$ and

$$\sigma^2 = \int (f-\bar{f})^2 p(x)\,\mathrm{d}x. \tag{3.5}$$

Next we discuss methods for generating non-uniform random variables starting from uniform random variables.

3.3. Transformation method

This is a general method for producing a random variable x with density $p(x)$, through transformation of a uniform random variable. Let y be a uniform variable and look for a function $X(y)$, so that $x = X(y)$ has the desired density $p(x)$.

Define the cumulative distribution function

$$P(x) = \int_{-\infty}^{x} p(x')\,\mathrm{d}x'. \tag{3.6}$$

Determination of the mapping $X(y)$ is through the following computation of the expectation. For any function f,

$$E_p[f(x)] = E_{\mathrm{unif}}[f(X(y))], \tag{3.7}$$

so that, using a change of variables,

$$\begin{aligned}\int f(x)p(x)\,\mathrm{d}x &= \int f(X(y))\,\mathrm{d}y \\ &= \int f(x)(\,\mathrm{d}y/\,\mathrm{d}x)\,\mathrm{d}x. \end{aligned} \tag{3.8}$$

This implies that $p(x) = \mathrm{d}y/\,\mathrm{d}x = 1/X'(y)$ so that $\int^{X(y)} p(x)\,\mathrm{d}x = y$; *i.e.*,

$$\begin{aligned} P(X(y)) &= y \\ X(y) &= P^{-1}(y). \end{aligned} \tag{3.9}$$

This formulation is convenient and explicit but not necessarily easy to implement, as it may be difficult to compute the inverse of the function P.

3.4. Gaussian random variables

As an example of the transformation method, a Gaussian (normal) random variable has density p and cumulative distribution function P given by

$$\begin{aligned} p(x) &= (2\pi)^{-1/2}\mathrm{e}^{-x^2/2}, \\ P(x) &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} \mathrm{e}^{-t^2/2}\,\mathrm{d}t \\ &= \tfrac{1}{2} + \tfrac{1}{2}\,\mathrm{erf}\,(x/\sqrt{2}), \end{aligned} \tag{3.10}$$

in which *erf* is the error function defined by

$$\mathrm{erf}\,(z) = \frac{2}{\sqrt{\pi}}\int_0^z \mathrm{e}^{-t^2}\,\mathrm{d}t. \tag{3.11}$$

One way to sample from a Gaussian distribution is to apply the inverse of the Gaussian cumulative distribution function P to a uniform random variable y. Sample a normal variable x, starting from a uniform variable y, by

$$y = P(x) = \tfrac{1}{2} + \tfrac{1}{2}\,\mathrm{erf}\,(x/\sqrt{2}), \tag{3.12}$$

that is,

$$x = \sqrt{2}\,\mathrm{erf}^{-1}(2y-1). \tag{3.13}$$

Approximate formulas for $P^{(-1)}(y)$ or erf^{-1} are found in Kennedy and Gentle (1980).

For the Gaussian distribution, as well as for a number of other distributions, special transformations are a useful alternative to the general transformation method. The simplest of these is the Box–Muller method. It provides a direct way of generating normal random variables without inverting the error function. Starting with two uniform variables y_1, y_2, two normal variables x_1, x_2 are obtained by

$$\begin{aligned} x_1 &= \sqrt{-2\log(y_1)}\cos(2\pi y_2), \\ x_2 &= \sqrt{-2\log(y_1)}\sin(2\pi y_2). \end{aligned} \tag{3.14}$$

Box–Muller is based on the following observation. First change from rectangular to polar coordinates, that is,

$$(x_1, x_2) = (r\cos\theta, r\sin\theta), \tag{3.15}$$

so that

$$\mathrm{d}x_1\,\mathrm{d}x_2 = r\,\mathrm{d}r\,\mathrm{d}\theta. \tag{3.16}$$

The corresponding transformation of the probability density function is

$$(2\pi)^{-1}\mathrm{e}^{-(x_1^2+x_2^2)/2}\,\mathrm{d}x_1\,\mathrm{d}x_2 = (2\pi)^{-1}\mathrm{e}^{-r^2/2}r\,\mathrm{d}r\,\mathrm{d}\theta. \tag{3.17}$$

This shows that the angular variable $y_1 = \theta/(2\pi)$ is uniformly distributed over the unit interval. Next the variable r is easily sampled, since it has density $r\mathrm{e}^{-r^2/2}$ and corresponding cumulative distribution function

$$P(r) = \int_0^r \mathrm{e}^{-r'^2/2}r'\,\mathrm{d}r' = 1 - \mathrm{e}^{-r^2/2}. \tag{3.18}$$

Therefore r can be sampled by

$$r = P^{-1}(y_2) = \sqrt{-2\log(1-y_2)}. \tag{3.19}$$

After replacing y_2 by $1 - y_2$, the resulting transform

$$(y_1, y_2) \to (r, \theta) \to (x_1, x_2) \tag{3.20}$$

is given in (3.14).

The only disadvantages of the Box–Muller method are that it requires evaluation of transcendental functions and that it generates two random variables when only one may be needed. See Marsaglia (1991) for examples of efficient use of Box–Muller.

3.5. Acceptance–rejection method

Another general way of producing random variables from a given density $p(x)$ is based on a probabilistic approach. This method shows the power and flexibility of stochastic methods. For a given density function $p(x)$, suppose that we know a function $q(x)$ satisfying

$$q(x) \geq p(x), \tag{3.21}$$

and that we have a way to sample from the probability density

$$\hat{q}(x) = q(x)/I[q], \tag{3.22}$$

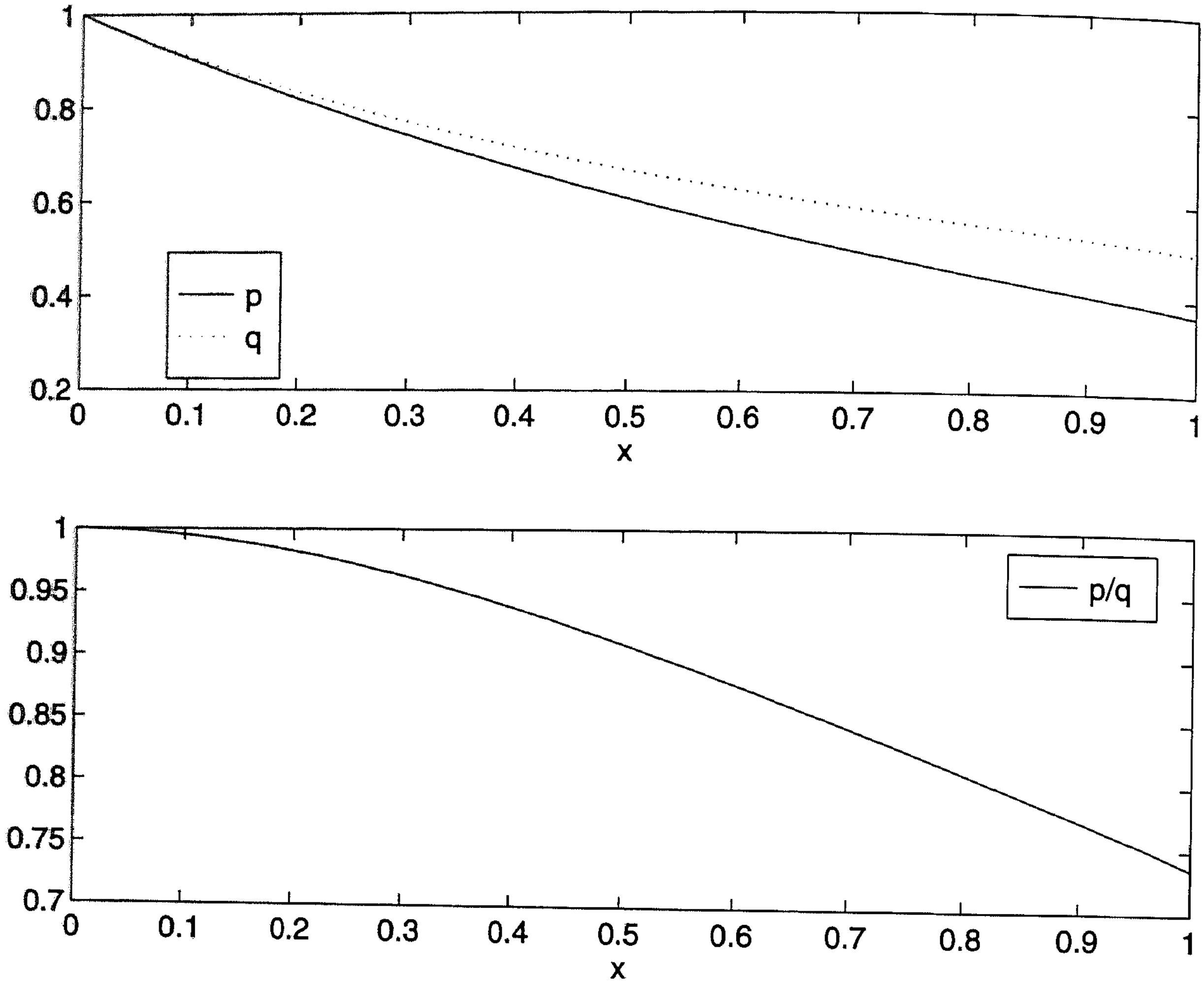


Fig. 1. Typical choice of p, q and p/q. Accept for $y < p/q$ and reject for $y > p/q$

in which $I[q] = \int q(x')\,\mathrm{d}x'$. In practice, q is often chosen to be a constant.

The acceptance–rejection procedure goes as follows. Pick two random variables, x', y, in which x' is a trial variable chosen according to the probability density $\hat{q}(x')$, and y is a decision variable chosen according to the uniform density on $0 < y < 1$. The decision is to

- *accept* if $0 < y < p(x')/q(x')$
- *reject* if $p(x')/q(x') < y < 1$.

This procedure is repeated until a value x' is accepted. Once a value x' is accepted, take

$$x = x'. \tag{3.23}$$

This procedure is depicted graphically in Figure 1.

Here is a derivation of the acceptance–rejection method. Since $0 \le p \le q$,

$$\begin{aligned} p(x) &= \frac{p(x)}{q(x)}\hat{q}(x)I[q] \\ &= \int_0^1 \chi\left(\frac{p(x)}{q(x)} > y\right)\,\mathrm{d}y\;\hat{q}(x)I[q]. \end{aligned} \tag{3.24}$$

So,

$$\begin{aligned} \int f(x)p(x)\,\mathrm{d}x &= \int\int_0^1 f(x)\,\chi\left(\frac{p(x)}{q(x)} > y\right)\hat{q}(x)\,\mathrm{d}y\,\mathrm{d}x\ I[q] \\ &\approx N'^{-1} \sum_{p(x'_n)/q(x'_n)>y_n} f(x'_n)\ I[q] \\ &\approx N^{-1} \sum_{\text{accepted pts}} f(x_n), \end{aligned} \tag{3.25}$$

in which

$$\begin{aligned} N' &= \text{total number of trial points,} \\ N &= \text{total number of accepted points} \\ &\approx N'/I[q]. \end{aligned} \tag{3.26}$$

The acceptance–rejection method has some obvious advantages that often make it the method of choice for generating random variables with a given distribution. It does not require inversion of the cumulative distribution function P. In addition, it works even if the density function p has not been normalized to have integral 1. One disadvantage of the method is that it may be inefficient, requiring many trials before acceptance. In practice this is often not a serious deficiency, since the largest share of the computation is on the integrand rather than on the random point selection.

An extension of the acceptance–rejection method called the Metropolis algorithm (Kalos and Whitlock 1986) is used to find the equilibrium distribution for a stochastic process.

4. Variance reduction

In Monte Carlo integration, the error ϵ and the number N of samples are related by

$$\epsilon = O\left(\sigma N^{-1/2}\right), \tag{4.1}$$

$$N = O(\sigma/\epsilon)^2. \tag{4.2}$$

The computational time required for the method is proportional to the sample number N and thus is of size $O(\sigma/\epsilon)^2$, which shows that computational time grows rapidly as the desired accuracy is tightened. There are two options for acceleration (error reduction) of Monte Carlo. The first is variance reduction, in which the integrand is transformed in a way that reduces the constant variance σ. The second is to modify the statistics, that is, to replace the random variables by an alternative sequence which improves the exponent 1/2. In this section, various strategies for variance reduction are described. In Section 5, we discuss quasi-random variables, an example of the second approach. One note of caution is that the acceleration methods described here may require extra computational time, which must be balanced against savings gained by reduced N. In most examples, however, the savings due to variance reduction are quite significant.

4.1. Antithetic variables

Antithetic variables is one of the simplest and most widely used variance reduction methods. The method is as follows: for each (multi-dimensional) sample value $\boldsymbol{x}$, also use the value $-\boldsymbol{x}$. The resulting Monte Carlo quadrature rule is

$$I_N[f] = \frac{1}{2N} \sum_{n=1}^{N} \{f(\boldsymbol{x}_n) + f(-\boldsymbol{x}_n)\}. \tag{4.3}$$

For example, the vector $\boldsymbol{x}$ could be the discrete states in a random walk, so that the dimension would be the number of time-steps in the walk. When antithetic variables are used for a problem involving a random walk, then for each path $\boldsymbol{x} = (x_1, \ldots, x_k)$, we also use the path $-\boldsymbol{x} = (-x_1, \ldots, -x_k)$.

The use of antithetic variables is motivated by an expansion for small values of the variance. Consider, for example, the expectation $E[f(x)]$ in which x is an $N(0, \sigma^2)$ random variable with σ small. Set

$$x = \sigma \hat{x}. \tag{4.4}$$

The Taylor expansion of $f = f(\sigma\hat{x})$ (for small σ) is

$$f = f(0) + f'(0)\sigma\hat{x} + O(\sigma^2). \tag{4.5}$$

Since the distribution of $\hat{x}$ is symmetric about 0, the average $E[\hat{x}]$ of the linear term is zero. In a standard Monte Carlo integral of f, these terms do not cancel exactly, so that the Monte Carlo error is proportional to σ. With antithetic variables, on the other hand, the linear terms cancel exactly, so that the remaining error is proportional to σ^2.

4.2. Control variates

The idea of control variates is to use an integrand g, which is similar to f and for which the integral $I[g] = \int g(x)\,\mathrm{d}x$ is known. The integral $I[f]$ is then written as

$$\int f(x)\,\mathrm{d}x = \int (f(x) - g(x))\,\mathrm{d}x + \int g(x)\,\mathrm{d}x. \tag{4.6}$$

Monte Carlo quadrature is used on the integral of $f - g$ to obtain the formula

$$I_n[f] = \frac{1}{N} \sum_{n=1}^{N} (f(x_n) - g(x_n)) + I[g]. \tag{4.7}$$

The resulting integration error $\epsilon_N[f] = I[f] - I_N[f]$ is of size

$$\epsilon_N[f] \approx \sigma_{f-g} N^{-1/2}, \tag{4.8}$$

in which the relevant variance is

$$\sigma_{f-g}^2 = \int \left(\tilde{f}(x) - \tilde{g}(x)\right)^2 \mathrm{d}x \tag{4.9}$$

using the notation

$$\tilde{f}(x) = f(x) - I[f], \qquad \tilde{g}(x) = g(x) - I[g]. \tag{4.10}$$

This shows that the control variate method is effective if

$$\sigma_{f-g} \ll \sigma_f. \tag{4.11}$$

Optimal use of a control variate is made by introducing a multiplier λ. For a given function g, write the integral of f as

$$\int f(x)\,\mathrm{d}x = \int (f(x) - \lambda g(x))\,\mathrm{d}x + \lambda \int g(x)\,\mathrm{d}x. \tag{4.12}$$

The error in Monte Carlo quadrature of the first integral is proportional to the variance

$$\sigma_{f-\lambda g}^2 = \int \left(\tilde{f}(x) - \lambda \tilde{g}(x)\right)^2 \mathrm{d}x. \tag{4.13}$$

The optimal value of λ is found by minimizing $\sigma_{f-\lambda g}^2$ to obtain

$$\begin{aligned} \lambda &= E[\tilde{f}\tilde{g}]/E[\tilde{g}^2] \\ &= \left(\int \tilde{f}\tilde{g}\,\mathrm{d}x\right) / \left(\int \tilde{g}^2\,\mathrm{d}x\right). \end{aligned} \tag{4.14}$$

4.3. Matching moments method

Monte Carlo integration error is partly due to statistical sampling error, that is, differences between the desired density function $p(x)$ and the empirical density function of the sampled points $\{x_n\}_{n=1}^N$. Part of this difference can be seen directly through comparison of the moments of the two distributions. Define the first and second moments m_1 and m_2 of p as

$$m_1 = \int x p(x)\,\mathrm{d}x, \qquad m_2 = \int x^2 p(x)\,\mathrm{d}x. \tag{4.15}$$

The first and second moments μ_1 and μ_2 of the sample $\{x_n\}_{n=1}^N$ are

$$\mu_1 = N^{-1} \sum_{n=1}^N x_n, \qquad \mu_2 = N^{-1} \sum_{n=1}^N x_n^2. \tag{4.16}$$

The moment error is due to the inequality of these moments, that is,

$$\mu_1 \neq m_1, \qquad \mu_2 \neq m_2. \tag{4.17}$$

A partial correction to the statistical sampling error is to make the moments exactly match. This can be done by a simple transformation of the sample points. To match the first moment of the sample with that of p, replace x_n by

$$y_n = (x_n - \mu_1) + m_1. \tag{4.18}$$

This satisfies

$$N^{-1}\sum y_n = m_1 \tag{4.19}$$

so that the first moment is exactly right. To match the first two moments, replace x_n by

$$y_n = (x_n - \mu_1)/c + m_1, \qquad c = \sqrt{\frac{m_2 - m_1^2}{\mu_2 - \mu_1^2}}. \tag{4.20}$$

Then

$$N^{-1}\sum y_n = m_1, \qquad N^{-1}\sum y_n^2 = m_2. \tag{4.21}$$

These transformed sequences with the correct moments must be used with some caution. Since the transformation involves μ_1 and μ_2, the new sample points y_n are no longer independent, and Monte Carlo error estimates are not so straightforward. For example, the Central Limit Theorem is not directly applicable, and the method may be biased. Actually, this is an example of the second approach to acceleration of Monte Carlo, through modification of the properties of the random sequence. It is presented here along with variance reduction methods, however, because the resulting improvement in Monte Carlo accuracy is comparable to that gained through variance reduction.

Pullin (1979) has formulated a method in which the sample points are independent and have a prescribed empirical mean and variance, but come from a Gaussian distribution with a randomly chosen mean and variance.

4.4. Stratification

Stratification combines the benefits of a grid with those of random variables. In the simplest case, stratification is based on a regular grid with uniform density in one dimension. Split the integration region $\Omega = [0,1]$ into M pieces Ω_k given by

$$\Omega_k = \left[\frac{(k-1)}{M}, \frac{k}{M}\right]. \tag{4.22}$$

Then, each subset has the same measure, $|\Omega_k| = 1/M$. Define the averages over each Ω_k by

$$\bar{f}(x) = \bar{f}_k = |\Omega_k|^{-1}\int_{\Omega_k} f(x)\,dx \qquad \text{for } x \in \Omega_k. \tag{4.23}$$

For each k, sample $N_k = N/M$ points $\{x_i^{(k)}\}$ uniformly distributed in Ω_k. Then the stratified quadrature formula is just the sum of the quadratures over each subset, that is,

$$I_N = N^{-1} \sum_{k=1}^{M} \sum_{i=1}^{N/M} f\left(x_i^{(k)}\right). \tag{4.24}$$

The Monte Carlo quadrature error for this stratified sum is

$$\begin{aligned} \epsilon &\approx N^{-1/2}\sigma_s, \\ \sigma_s^2 &= \int (f(x) - \bar{f}(x))^2 \, \mathrm{d}x \\ &= \sum_{k=1}^{M} \int_{\Omega_k} (f(x) - \bar{f}_k)^2 \, \mathrm{d}x. \end{aligned} \tag{4.25}$$

For this stratified quadrature rule, there is a simple result. Stratified Monte Carlo quadrature always beats the unstratified quadrature, since

$$\sigma_s \leq \sigma. \tag{4.26}$$

The proof of this inequality is straightforward. For each k, $c = \bar{f}_k$ is the minimizer of

$$\int_{\Omega_k} (f(x) - c)^2 \, \mathrm{d}x. \tag{4.27}$$

In particular,

$$\int_{\Omega_k} (f(x) - \bar{f}_k)^2 \, \mathrm{d}x \leq \int_{\Omega_k} (f(x) - \bar{f})^2 \, \mathrm{d}x. \tag{4.28}$$

Add this over all k to get

$$\begin{aligned} \sigma_s^2 &= \sum_{k=1}^{M} \int_{\Omega_k} (f(x) - \bar{f}_k)^2 \, \mathrm{d}x \\ &\leq \sum_{k=1}^{M} \int_{\Omega_k} (f(x) - \bar{f})^2 \, \mathrm{d}x \\ &= \sigma^2. \end{aligned} \tag{4.29}$$

Stratification can be phrased more generally as follows. Split the integration region Ω into M pieces Ω_k with

$$\Omega = \cup_{k=1}^{M} \Omega_k. \tag{4.30}$$

Take N_k random variables in each piece Ω_k with

$$\sum_{k=1}^{M} N_k = N. \tag{4.31}$$

In each subset Ω_k choose points $x_n^{(k)}$ distributed with density $p^{(k)}(x)$ in which

$$\begin{aligned} p^{(k)}(x) &= p(x)/\bar{p}_k, \\ \bar{p}_k &= \int_{\Omega_k} p(x)\,\mathrm{d}x. \end{aligned} \tag{4.32}$$

The stratified quadrature formula is the following sum over k:

$$I_N[f] = \sum_{k=1}^{M} \frac{\bar{p}_k}{N_k} \sum_{n=1}^{N_k} f\left(x_n^{(k)}\right). \tag{4.33}$$

The resulting integration error is

$$\begin{aligned} \epsilon_N[f] &= I[f] - I_N[f] \\ &= \sum_{k=1}^{M} \epsilon_{N_k}^{(k)}[f]. \end{aligned} \tag{4.34}$$

The components of this error are

$$\begin{aligned} \epsilon_{N_k}^{(k)}[f] &\approx N_k^{-1/2}\bar{p}_k \left(\int_{\Omega_k} (f(x) - \bar{f}_k)^2 p^{(k)}(x)\,\mathrm{d}x \right)^{1/2} \\ &= (\bar{p}_k/N_k)^{1/2}\sigma^{(k)}, \end{aligned} \tag{4.35}$$

in which the variances are

$$\begin{aligned} \sigma^{(k)} &= (\bar{p}_k)^{1/2} \left(\int_{\Omega_k} (f(x) - \bar{f}_k)^2 p^{(k)}(x)\,\mathrm{d}x \right)^{1/2} \\ &= \left(\int_{\Omega_k} (f(x) - \bar{f}_k)^2 p(x)\,\mathrm{d}x \right)^{1/2}, \end{aligned} \tag{4.36}$$

and the averages are

$$\bar{f}_k = \int_{\Omega_k} f(x)p(x)\,\mathrm{d}x/\bar{p}_k. \tag{4.37}$$

Stratification always lowers the integration error if the distribution of points is balanced. The balance condition is that, for all k,

$$\bar{p}_k/N_k = 1/N, \tag{4.38}$$

that is, the number of points in set Ω_k is proportional to its weighted size $\bar{p}_k$. The resulting error for stratified quadrature is

$$\epsilon_N \approx N^{-1/2}\sigma_s, \qquad \sigma_s^2 = \sum_{k=1}^{M} \sigma^{(k)2}. \tag{4.39}$$

Since the variance over a subset is always less than the variance over the whole set, that is,

$$\sigma_s \le \sigma, \tag{4.40}$$

then stratification always lowers the integration error. Actually, a better choice than the balance condition may be to put more points where f has largest variation. An adaptive method for stratification was formulated by Lepage (1978) and is described by Press et al. (1992).

4.5. *Importance sampling*

Importance sampling is probably the most widely used Monte Carlo variance reduction method. Consider the simple integral $\int f(x)\,\mathrm{d}x$ and rewrite it by introducing a density p as

$$\begin{aligned} I[f] &= \int f(x)\,\mathrm{d}x \\ &= \int \frac{f(x)}{p(x)} p(x)\,\mathrm{d}x. \end{aligned} \tag{4.41}$$

Now think of this as an integral with density function p. Sample points x_n from the density $p(x)$ and form the Monte Carlo estimate

$$I_N[f] = \frac{1}{N}\sum_{n=1}^{N} \frac{f(x_n)}{p(x_n)}. \tag{4.42}$$

The resulting error $\epsilon_N[f] = I[f] - I_N[f]$ has size

$$\epsilon_N[f] \approx \sigma_p N^{-1/2}, \tag{4.43}$$

in which the variance σ_p is

$$\sigma_p = \int \left(\frac{f(x)}{p(x)} - I\right)^2 p(x)\,\mathrm{d}x. \tag{4.44}$$

So importance sampling will reduce the Monte Carlo quadrature error, if

$$\sigma_p \ll \sigma. \tag{4.45}$$

Importance sampling is an effective method when f/p is nearly constant, so that σ_p is small. One difficulty in this method is that sampling from the density p is required, but this can be performed using acceptance–rejection, if necessary. One use of importance sampling is to emphasize rare but important events, *i.e.*, small regions of space in which the integrand f is large.

4.6. *Russian roulette*

Some Monte Carlo computations involve infinite sums, for instance, due to iteration of an integral equation. The Russian roulette method converts an infinite sum into a sum that is of finite length for each sample. Consider the sum

$$S = \sum_{n=0}^{\infty} a_n \tag{4.46}$$

and suppose that the terms a_n are exponentially decreasing, that is,

$$|a_n| \leq c\lambda^n, \tag{4.47}$$

in which $0 < \lambda < 1$. Choose $\lambda < \kappa < 1$, and let M be chosen according to a discrete exponential distribution, so that

$$\text{Prob}(M \geq n) = \kappa^n. \tag{4.48}$$

Define the random sum

$$\tilde{S} = \sum_{n=0}^{M} \kappa^{-n} a_n. \tag{4.49}$$

Since $|\kappa^{-n} a_n| < (\lambda/\kappa)^n$ and $|\lambda/\kappa| < 1$, then this sum is uniformly bounded for all M. Then

$$\begin{aligned} E[\tilde{S}] &= \sum_{n=0}^{\infty} \text{Prob}(M \geq n)\kappa^{-n} a_n \\ &= S. \end{aligned} \tag{4.50}$$

This formula leads to the following Monte Carlo method for computation of the infinite sum (4.46):

$$S_N = N^{-1} \sum_{i=0}^{N} \sum_{n=0}^{M_i} \kappa^{-n} a_n, \tag{4.51}$$

in which the values M_i are chosen according to the probability distribution (4.48). In this method, the terms a_n could also be computed by a Monte Carlo integral.

4.7. Example of variance reduction

As an example of the use of variance reduction consider the three-dimensional Gaussian integral

$$I[u] = (2\pi)^{-3/2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u e^{-(x_1^2+x_2^2+x_3^2)/2}\, dx_1\, dx_2\, dx_3, \tag{4.52}$$

in which the integrand u is defined by

$$\begin{aligned} u &= (1+r_0)^{-1}(1+r_1)^{-1}(1+r_2)^{-1}(1+r_3)^{-1}, \\ r_1 &= r_0 e^{\sigma x_1 - \sigma^2/2}, \\ r_2 &= r_1 e^{\sigma x_2 - \sigma^2/2}, \\ r_3 &= r_2 e^{\sigma x_3 - \sigma^2/2}. \end{aligned} \tag{4.53}$$

This integral can be interpreted as the present value of a payment of \$1 four years from now, for an interest rate of r_i in year i. The interest rates are allowed to evolve according to a lognormal model with variance σ, that is,

$$r_i = r_{i-1}\mathrm{e}^{\sigma x_i - \sigma^2/2}, \tag{4.54}$$

in which the $x_i's$ are independent $N(0,1)$ Gaussian random variables. The factors $(1+r_i)^{-1}$ are the discount factors. For the computations below, we take $r_0 = 0.10$ and $\sigma = 0.1$.

We evaluate $I[u]$ by sampling x_i from an $N(0,1)$ distribution using the transformation method, that is, by direct inversion of the cumulative distribution function for a normal random variable. Then we apply antithetic variables and control variates to this problem.

Approximate $(1+r_i)^{-1}$ by $1-r_i$ for $i \geq 1$. Then form the control variate as

$$g = (1+r_0)^{-1}(1-r_1)(1-r_2)(1-r_3). \tag{4.55}$$

Since g consists of a sum of linear exponentials, its integral can be performed exactly, for instance

$$\begin{aligned}\int_{-\infty}^{\infty} \mathrm{e}^{\lambda x}\mathrm{e}^{-x^2/2}\,\mathrm{d}x &= \mathrm{e}^{\lambda^2/2}\int_{-\infty}^{\infty} \mathrm{e}^{-(x-\lambda)^2/2}\,\mathrm{d}x \\ &= \mathrm{e}^{\lambda^2/2}\int_{-\infty}^{\infty} \mathrm{e}^{-y^2/2}\,\mathrm{d}y \\ &= \sqrt{2\pi}\mathrm{e}^{\lambda^2/2}. \end{aligned}\tag{4.56}$$

Numerical results are presented in Figure 2. This compares the quadrature error from standard Monte Carlo, antithetic variables and control variates, using both standard pseudo-random points and the quasi-random (Sobol') points that will be discussed in the next section. In order to make a meaningful comparison, we have performed an average (root mean square) of the error over 20 runs for each value of N. The computations are all independent, that is, the same random number is never used twice. The figure also shows the results from a single run without averaging or variance reduction. The results show the following points.

- Both control variates and antithetic variables provide significant error reduction.
- Control variates and antithetic variables together perform better than either one alone. This shows that the error reduction from the two methods are different.
- Quasi-Monte Carlo, which will be discussed in Section 5, has smaller error and a faster rate of convergence.
- Control variates and antithetic variables, both separately and together, provide further error reduction when used with quasi-Monte Carlo.

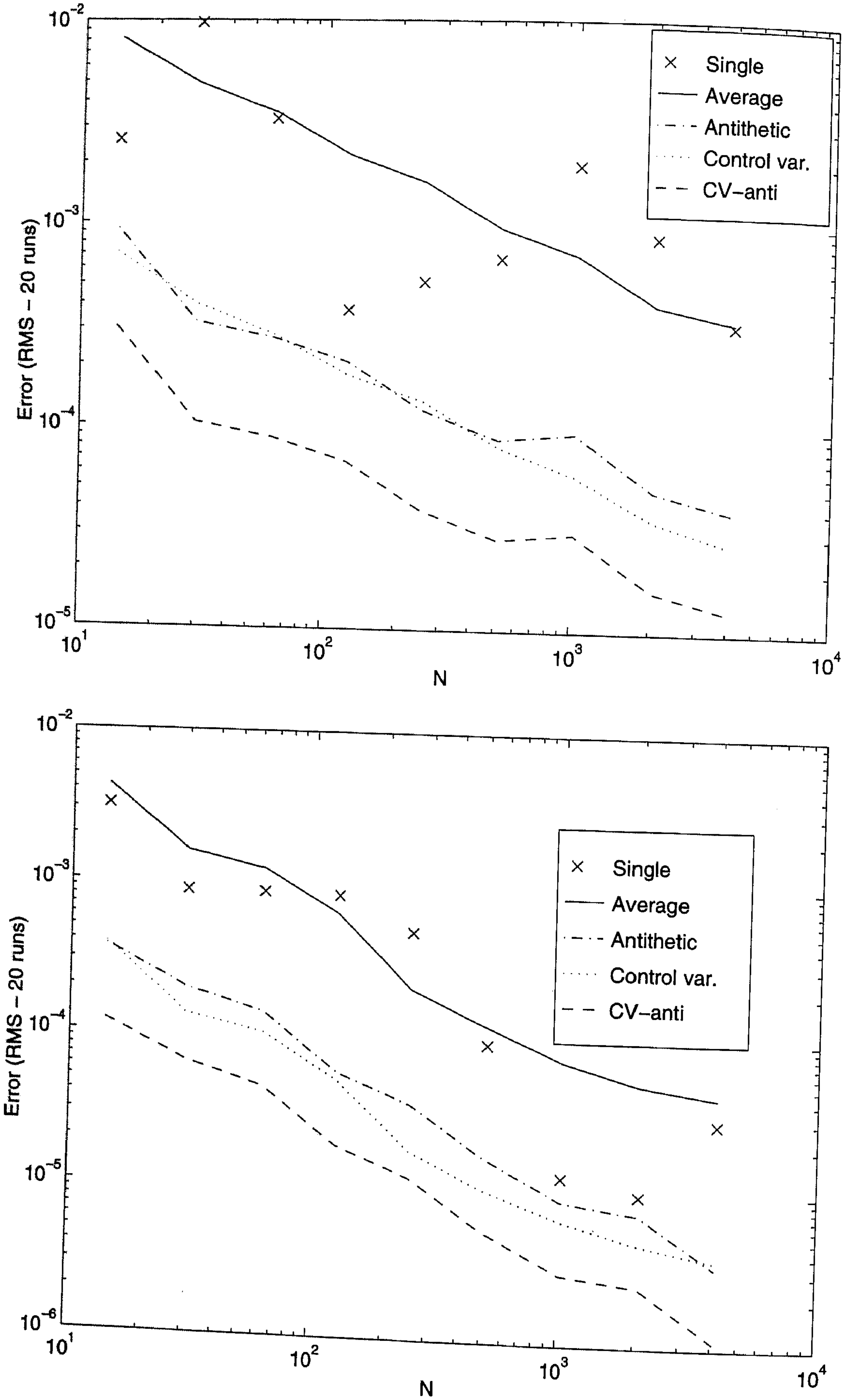


Fig. 2. Discounted cashflow, Monte Carlo (top) and quasi-Monte Carlo (bottom)

5. Quasi-random numbers

Quasi-random sequences are a deterministic alternative to random or pseudo-random sequences (Kuipers and Niederreiter 1974, Hua and Wang 1981, Niederreiter 1992, Zaremba 1968). In contrast to pseudo-random sequences, which try to mimic the properties of random sequences, quasi-random sequences are designed to provide better uniformity than a random sequence, and hence faster convergence for quadrature formulas. Uniformity of a sequence is measured in terms of its *discrepancy*, which is defined in Section 5.2 below, and for this reason quasi-random sequences are also called low-discrepancy sequences. Some have objected to the name quasi-random, since these sequences are intentionally not random. We continue to use this name, however, because of the unpredictability of quasi-random sequences.

5.1. Monte Carlo versus quasi-Monte Carlo

The difference between standard Monte Carlo and quasi-Monte Carlo is best illustrated by Figure 3, which compares a pseudo-random sequence with a quasi-random sequence (a Sobol' sequence) in two dimensions. Standard Monte Carlo methods use pseudo-random sequences and provide a convergence rate of $O(N^{-1/2})$ for Monte Carlo quadrature using N samples. In addition to integration problems, standard Monte Carlo is applicable to optimization and simulation problems.

The limiting factor in accuracy of standard Monte Carlo is the clumping that occurs in the points of a random or pseudo-random sequence. Clumping of points, as well as spaces that have no points, is clearly seen in the pseudo-random sequence in Figure 3. The reason for this clumping is that the points are independent. Since different points know nothing about each other, there is some small chance that they will lie very close together. A simple argument shows that about $\sqrt{N}$ out of N points lie in clumps.

Quasi-Monte Carlo methods use quasi-random sequences, which are deterministic, with correlations between the points to eliminate clumping. The resulting convergence rate is $O((\log N)^k N^{-1})$. Because of the correlations, quasi-random sequences are less versatile than random or pseudo-random sequences. They are designed for integration, rather than simulation or optimization. On the other hand, the desired result of a simulation can often be written as an expectation, which is an integral, so that quasi-Monte Carlo is then applicable. As discussed below, this often introduces high dimensionality, which can limit the effectiveness of quasi-Monte Carlo sequences.

5.2. Discrepancy

Quasi-random sequences were invented by number theorists who were interested in the uniformity properties of numerical sequences (Kuipers and Niederreiter 1974, Hua and Wang 1981). An important first step is formu-

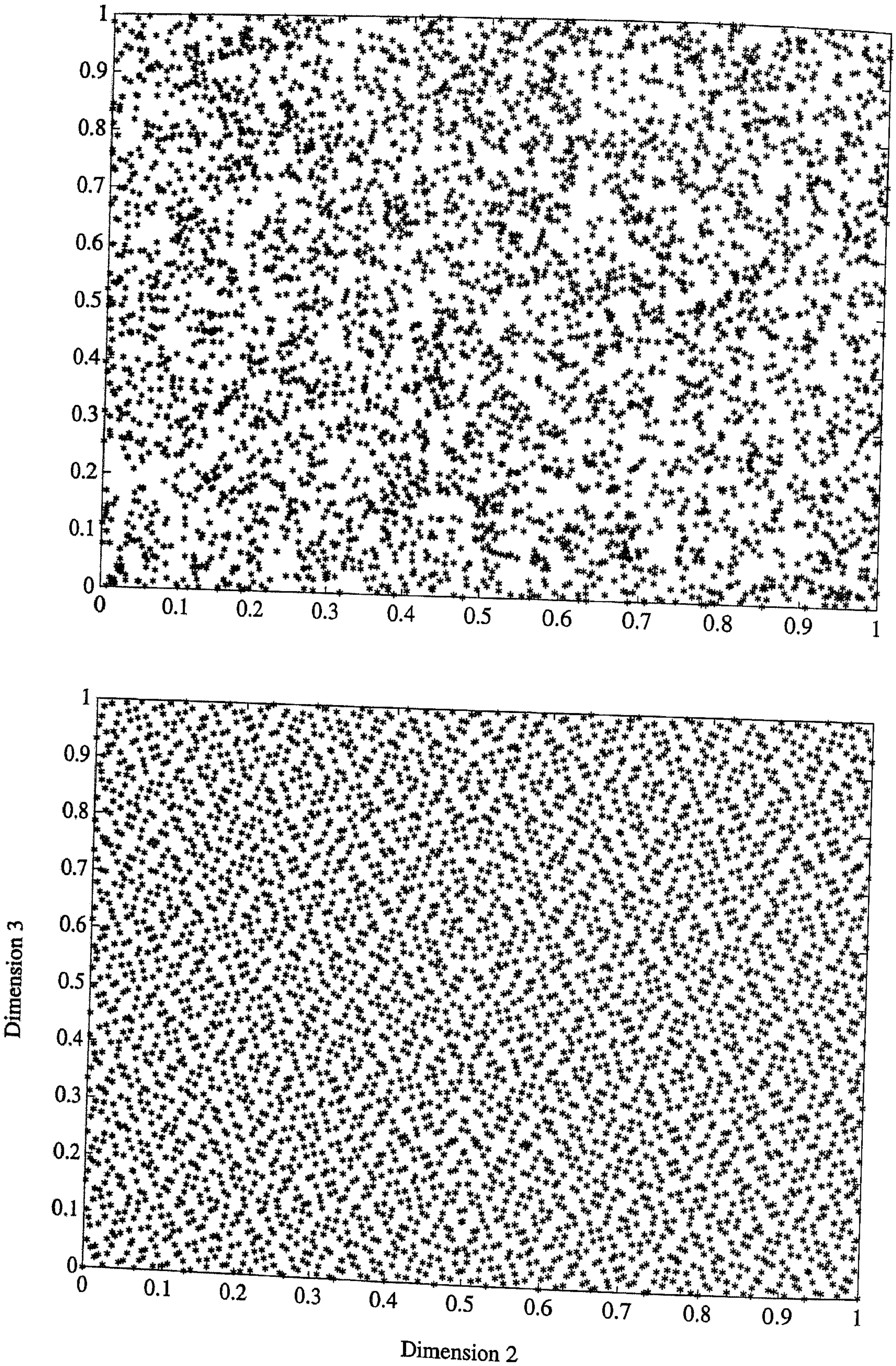


Fig. 3. Two-dimensional projection of a pseudo-random sequence (top) and a Sobol' sequence (bottom)

lating a quantitative measure of uniformity. Uniformity of a sequence of points is measured in terms of its *discrepancy.*

For a sequence of N points $\{\boldsymbol{x}_n\}$ in the unit cube I^d, define

$$R_N(J) = \frac{1}{N}\#\{\boldsymbol{x}_n \in J\} - m(J) \tag{5.1}$$

for any subset J of I^d. $R_N(J)$ is just the Monte Carlo quadrature error in measuring the volume of J. The discrepancy is then defined by some norm applied to $R_N(J)$. First, restrict J to be a rectangular set, and define the set of all such subsets to be E. Since a rectangular set can be defined by two antipodal vertices, a rectangle J can be identified as $J = J(\boldsymbol{x}, \boldsymbol{y})$ in which the points $\boldsymbol{x}$ and $\boldsymbol{y}$ are antipodal vertices. Now define two discrepancies: D_N, which is an L^∞ norm on J, and T_N, which is an L^2 norm, defined by

$$\begin{aligned} D_N &= \sup_{J\in E} |R_N(J)| \\ T_N &= \left[\int_{(\boldsymbol{x},\boldsymbol{y})\in I^{2d}, x_i<y_i} R_N(J(\boldsymbol{x},\boldsymbol{y}))^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y}\right]^{\frac{1}{2}} . \end{aligned} \tag{5.2}$$

It is useful to also define the discrepancy over rectangular sets with one vertex at 0, as

$$\begin{aligned} D_N^* &= \sup_{J\in E^*} |R_N(J)| \\ T_N^* &= \left[\int_{I^d} R_N(J(0,\boldsymbol{x}))^2 \,\mathrm{d}\boldsymbol{x}\right]^{\frac{1}{2}} \end{aligned} \tag{5.3}$$

in which E^* is the set $\{J(0, \boldsymbol{x})\}$. Other definitions include terms from lower-dimensional projections of the sequence (Hickernell 1998).

5.3. *Koksma–Hlawka inequality*

Quasi-random sequences are useful for integration because they lead to much smaller error than standard Monte Carlo quadrature. The basis for analysing quasi-Monte Carlo quadrature error is the Koksma–Hlawka inequality.

Consider the integral

$$I[f] = \int_{I^d} f(x)\,\mathrm{d}x \tag{5.4}$$

and the Monte Carlo approximation

$$I_N[f] = \frac{1}{N}\sum_{n=1}^{N} f(x_n), \tag{5.5}$$

and define the quadrature error

$$\varepsilon[f] = |I[f] - I_N[f]| . \tag{5.6}$$

Define the variation (in the Hardy–Krause sense) of f, a function of a single variable, as

$$V[f] = \int_0^1 \left| \frac{\mathrm{d}f}{\mathrm{d}t} \right| \mathrm{d}t. \tag{5.7}$$

In d dimensions, the variation is defined as

$$V[f] = \int_{I^d} \left| \frac{\partial^d f}{\partial t_1 \cdots \partial t_d} \right| \mathrm{d}t_1 \cdots \mathrm{d}t_d + \sum_{i=1}^{d} V[f_1^{(i)}] \tag{5.8}$$

in which $f_1^{(i)}$ is the restriction of the function f to the boundary $x_i = 1$. Since these restrictions are functions of $d-1$ variables, this definition is recursive.

Theorem 5.1 (Koksma–Hlawka theorem) For any sequence $\{x_n\}$ and any function f with bounded variation, the integration error ε is bounded as

$$\varepsilon[f] \leq V[f]\, D_N^*. \tag{5.9}$$

It is instructive to compare the Koksma–Hlawka result to the root mean square error (RMSE) of standard Monte Carlo integration using a random sequence, that is,

$$E[\varepsilon[f]^2]^{1/2} = \sigma[f] N^{-1/2}, \tag{5.10}$$

in which

$$\begin{aligned} \sigma[f] &= \left(\int_{I^d} (f(x) - I[f])^2 \, \mathrm{d}x \right)^{1/2}, \\ \varepsilon[f] &= \int_{I^d} f(x) \, \mathrm{d}x - \frac{1}{N} \sum_{n=1}^{N} f(x_n). \end{aligned} \tag{5.11}$$

In comparing the Koksma–Hlawka inequality (5.9) and the RMSE for standard Monte Carlo integration (5.10), note the following points.

- Both error bounds are a product of one factor that depends on the sequence (*i.e.*, D_N for Koksma–Hlawka and $N^{-1/2}$ for RMSE) and a factor that depends on the function f (*i.e.*, $V[f]$ for Koksma–Hlawka and $\sigma[f]$ for RMSE).
- The Koksma–Hlawka inequality is a worst-case bound, whereas the RMSE (5.10) is a probabilistic bound.
- The RMSE (5.10) is an equality, and so it is tight, whereas Koksma–Hlawka is an upper bound.
- Our experience is that $V[f]$ in Koksma–Hlawka is usually a gross overestimate, while the discrepancy is indicative of actual performance.

A further, more subtle point is that for standard Monte Carlo each point is an estimate of the entire integral. A typical quasi-random sequence, on the other hand, has a hierarchical structure, so that the initial points sample the integrand on a coarse scale and later points sample it on a finer scale. This is the source of the $\log N$ terms in the discrepancy bounds cited below.

5.4. Proof of the Koksma–Hlawka inequality

First consider functions f that vanish on the boundary of the unit cube. Note that

$$R(\boldsymbol{x}) = \left\{ \frac{1}{N} \sum_{n=1}^{N} \delta(\boldsymbol{x} - \boldsymbol{x}_n) - 1 \right\} \mathrm{d}\boldsymbol{x}, \tag{5.12}$$

in which $R(\boldsymbol{x}) = R_N(J(0, \boldsymbol{x}))$ as defined in Section 5.2. Since there are no boundary terms, then

$$\begin{aligned} \varepsilon[f] &= \left| \int_{I^d} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} - \frac{1}{N} \sum_{n=1}^{N} f(\boldsymbol{x}_n) \right| \\ &= \left| \int_{I^d} \left\{ 1 - \frac{1}{N} \sum_{n=1}^{N} \delta(\boldsymbol{x} - \boldsymbol{x}_n) \right\} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \right| \\ &= \left| \int_{I^d} R(\boldsymbol{x}) \, \mathrm{d}f(\boldsymbol{x}) \right| \\ &\leq \left(\sup_{\boldsymbol{x}} R(\boldsymbol{x}) \right) \int_{I^d} | \mathrm{d}f(\boldsymbol{x}) | \\ &= D_N^* V[f]. \end{aligned} \tag{5.13}$$

For f that is nonzero on the boundary of the unit cube, the terms from integration by parts are bounded by the boundary terms in $V[f]$.

5.5. Average integration error

Most computational experience confirms that discrepancy is a good indicator of quasi-Monte Carlo performance, while variation is not a typical indicator. In 1990, Woźniakowski (Woźniakowski 1991) proved the following surprising result.

Theorem 5.2 We have

$$E[\varepsilon_N[f]^2]^{1/2} = T_N^*, \tag{5.14}$$

in which the expectation E is taken with respect to functions f distributed according to the Brownian sheet measure.

The Brownian sheet measure is a measure on function space. It is a natural generalization of simple Brownian motion $b(x)$ to multi-dimensional 'time' $\boldsymbol{x}$. With probability one, a function $f(\boldsymbol{x})$, chosen from the Brownian sheet measure, is approximately 'half-differentiable'. In particular, its variation is infinite. In this context, we can definitely say that variation is not a good indicator of integration error. Since Woźniakowski's identity (5.14) is an equality, it follows that the L^2 discrepancy T_N^* does agree with the typical size of quasi-Monte Carlo integration error. Our computational experience is that T_N^* and D_N are usually of the same size.

Denote $\boldsymbol{x}' = (x_i')_{i=1}^d$ in which $x_i' = 1 - x_i$; also denote the finite difference operator $D_i f(\boldsymbol{x}') = f(\boldsymbol{x}' + \Delta_i \hat{e}_i') - f(\boldsymbol{x}')$, in which $\hat{e}_i'$ is the ith coordinate vector. The Brownian sheet is based at the point $\boldsymbol{x}' = 0$, that is, $\boldsymbol{x} = (1, \ldots, 1)$, and has $f(\boldsymbol{x}' = 0) = f(1, \ldots, 1) = 0$. For any point $\boldsymbol{x}'$ in $I^d$and any set of positive lengths Δ_i (with $x_i' + \Delta_i < 1$), the multi-dimensional difference $D_1 \ldots D_d f(\boldsymbol{x})$ is a normally distributed random variable with mean zero and variance

$$E[(D_1 \ldots D_d f(\boldsymbol{x}))^2] = \Delta_1 \ldots \Delta_d. \tag{5.15}$$

This implies that

$$E[\mathrm{d}f(\boldsymbol{x})\,\mathrm{d}f(\boldsymbol{y})] = \delta(\boldsymbol{x} - \boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y}, \tag{5.16}$$

in which $\mathrm{d}f$ is understood in the sense of the Itô calculus (Karatzas and Shreve 1991). Moreover $f(\boldsymbol{x}') = 0$ if $x_i' = 0$ for any i. For any $\boldsymbol{x}$ in I^d, $f(\boldsymbol{x})$ is normally distributed with mean zero and variance

$$E[f(\boldsymbol{x})^2] = \prod_{i=1}^d x_i'. \tag{5.17}$$

Woźniakowski's proof of (5.14) was a straightforward calculation of both sides of the equality. The following proof, derived by Morokoff and Caflisch (1994), is based on the properties of the Brownian sheet measure and follows the same lines as the proof of the Koksma–Hlawka inequality.

Proof of Woźniakowski's identity. First rewrite the integration error $E[f]$ using integration by parts, following the steps of the proof of the Koksma–Hlawka inequality. Note that

$$R(\boldsymbol{x}) = \left\{ \frac{1}{N} \sum_{n=1}^N \delta(\boldsymbol{x} - \boldsymbol{x}_n) - 1 \right\} \mathrm{d}\boldsymbol{x}, \tag{5.18}$$

in which $R(\boldsymbol{x}) = R_N(J(0, \boldsymbol{x}))$ as defined in Section 5.2. Also, $R(\boldsymbol{x}) = 0$ if $x_i = 0$, and $f(\boldsymbol{x}) = 0$ if $x_i = 1$, for any i. This implies that the boundary terms all disappear in the following integration by parts:

$$\begin{aligned}
\varepsilon[f] &= \left| \int_{I^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} - \frac{1}{N}\sum_{n=1}^{N} f(\boldsymbol{x}_n) \right| \\
&= \left| \int_{I^d} \left\{ 1 - \frac{1}{N}\sum_{n=1}^{N} \delta(\boldsymbol{x}-\boldsymbol{x}_n) \right\} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \right| \\
&= \left| \int_{I^d} R(\boldsymbol{x})\,\mathrm{d}f(\boldsymbol{x}) \right| .
\end{aligned}$$

The quantity $\mathrm{d}f$ in this identity is defined here through the Itô calculus, even though $V[f] = \infty$ with probability one.

It follows from (5.16) that the average square error is

$$\begin{aligned}
E[\varepsilon[f]^2] &= E\left[\left(\int_{I^d} R(\boldsymbol{x})\,\mathrm{d}f(\boldsymbol{x}) \right)^2 \right] \\
&= \int_{I^d \times I^d} R(\boldsymbol{x})R(\boldsymbol{y})E[\,\mathrm{d}f(\boldsymbol{x})\,\mathrm{d}f(\boldsymbol{y})] \\
&= \int_{I^d} R(\boldsymbol{x})^2\,\mathrm{d}\boldsymbol{x} \\
&= (T_N^*)^2. \qquad (5.19)
\end{aligned}$$

5.6. *Quasi-random number generators*

An infinite sequence $\{\boldsymbol{x}_n\}$ is *uniformly distributed* if

$$\lim_{N\to\infty} D_N = 0. \qquad (5.20)$$

The sequence is *quasi-random* if

$$D_N \leq c(\log N)^k N^{-1}, \qquad (5.21)$$

in which c and k are constants that are independent of N, but may depend on the dimension d. In particular, it is possible to construct sequences for which $k = d$. It is also common to say that a sequence is quasi-random only if the exponent k is equal to the dimension d of the sequence.

The simplest example of a quasi-random sequence is the Van der Corput sequence in one dimension ($d = 1$) (Niederreiter 1992). It is described most simply as follows. Write out n in base 2. Then obtain the nth point x_n by reversion of the bits of n around the decimal point, that is,

$$\begin{aligned}
n &= a_m a_{m-1} \ldots a_1 a_0 \quad \text{(base 2)} \\
x_n &= 0.a_0 a_1 \ldots a_{m-1} a_m \quad \text{(base 2)}. \qquad (5.22)
\end{aligned}$$

Halton sequences (Halton 1960) are a generalization of this procedure. In the kth dimension, expand n in base p_k, the kth prime, and form the kth component by reversion of this expansion around the decimal point. The discrepancy of a Halton sequence is bounded by

$$D_N(\text{Halton}) \leq c_d(\log N)^d N^{-1} \tag{5.23}$$

in which c_d is a constant depending on d. On the other hand, the average discrepancy of a random sequence is

$$E[T_N^2(\text{random})]^{1/2} = c_d N^{-1/2}. \tag{5.24}$$

Additional sequences go by the names of Haselgrove (Haselgrove 1961), Faure (Faure 1982), Sobol' (Sobol' 1967, 1976), Niederreiter (Niederreiter 1992, Xing and Niederreiter 1995), and Owen (Owen 1995, 1997, 1998, Hickernell 1996). Niederreiter has formulated a general theory for quasi-random sequences in terms of (t, s) nets (Niederreiter 1992). All of these sequences satisfy bounds like (5.23), but the more recent constructions have much better constants c_d.

An algorithm for construction of a Sobol' sequence is found in Press et al. (1992) and for Niederreiter sequences in Bratley, Fox and Niederreiter (1994). Various quasi-random number generators are found in the software collection ACM-Toms at `http://www.netlib.org/toms/index.html`.

For quasi-random sequences with discrepancy of size $O((\log N)^d N^{-1})$, the Koksma–Hlawka inequality implies that the integration error is of this same size, that is,

$$\varepsilon[f] \leq cV[f](\log N)^d N^{-1}. \tag{5.25}$$

Note that the Koksma–Hlawka inequality applies for any sequence. For quasi-random sequences, the discrepancy is small, so that the integration error is also small.

5.7. *Limitations: smoothness and dimension*

The Koksma–Hlawka inequality and discrepancy bounds for a quasi-random sequence together imply that quasi-Monte Carlo quadrature converges much faster than standard Monte Carlo quadrature. On the other hand, there are several distinct limitations to the effectiveness of quasi-Monte Carlo methods (Morokoff and Caflisch 1994, 1995).

First, there is no theoretical basis for empirical estimates of accuracy of quasi-Monte Carlo methods. Recall from Section 2 that the Central Limit Theorem can be used to test the accuracy of standard Monte Carlo quadrature as the computation proceeds, and then to predict the required number of samples N for a given accuracy level. Since there is no Central Limit Theorem for quasi-random sequences, there is no corresponding empirical error estimate for quasi-Monte Carlo. On the other hand, confidence in the accuracy of quasi-Monte Carlo integration comes from refining a set of computations.

Second, quasi-Monte Carlo methods are designed for integration and are not directly applicable to simulation. This is because of the correlations between the points of a quasi-random sequence. However, in many simulations the desired result is an expectation of some quantity, which can then be written as a high-dimensional integral on which quasi-Monte Carlo can be used. In fact, we can think of the different dimensions of a quasi-random point as independent, so that quasi-random sequences can represent a simulation by allocating one dimension per time-step or decision. This approach often requires a high-dimensional quasi-random sequence.

Third, quasi-Monte Carlo integration is found to lose its effectiveness when the dimension of the integral becomes large. This can be anticipated from the bound $(\log N)^d N^{-1}$ on discrepancy. For large dimension d, this bound is dominated by the $(\log N)^d$ term unless

$$N > 2^d. \tag{5.26}$$

Fourth, quasi-Monte Carlo integration is found to lose its effectiveness if the integrand f is not smooth. The factor $V[f]$ in the Koksma–Hlawka inequality (5.9) is an indicator of this dependence, although we actually find that a much smaller amount of smoothness, somewhere between continuity and differentiability, is usually enough. The limitation on smoothness is significant, because many Monte Carlo methods involve decisions of some sort, which usually correspond to multiplication by 0 or 1.

In the next subsection, some computational evidence for limitations on dimension and smoothness will be presented. Techniques for overcoming these limitations will be discussed in Section 6.

An important lesson from our computational experience is that quasi-Monte Carlo integration is almost always as accurate as standard Monte Carlo integration. So the 'loss of effectiveness' cited above means that quasi-Monte Carlo performs no better than standard Monte Carlo. The reasons for this are not clear, but it is consistent with the discrepancy computations presented in the next subsection.

5.8. Dependence on dimension

To demonstrate how quasi-random sequences depend on dimension, we will present results from two computational tests. The first is a computation of L^2 discrepancy for a quasi-random sequence. The second is an examination of the one- and two-dimensional projections of quasi-random sequences.

The L^2 discrepancy T_N can be computed directly (Morokoff and Caflisch 1994) by an explicit formula. Figure 4 shows computation of T_N for a Sobol' sequence for dimension 4 and 16. In addition to T_N, each graph shows a plot of $N^{-1/2}$ with a constant coming from the expected discrepancy of a random sequence. Each graph also shows a plot of N^{-1} (with a constant of

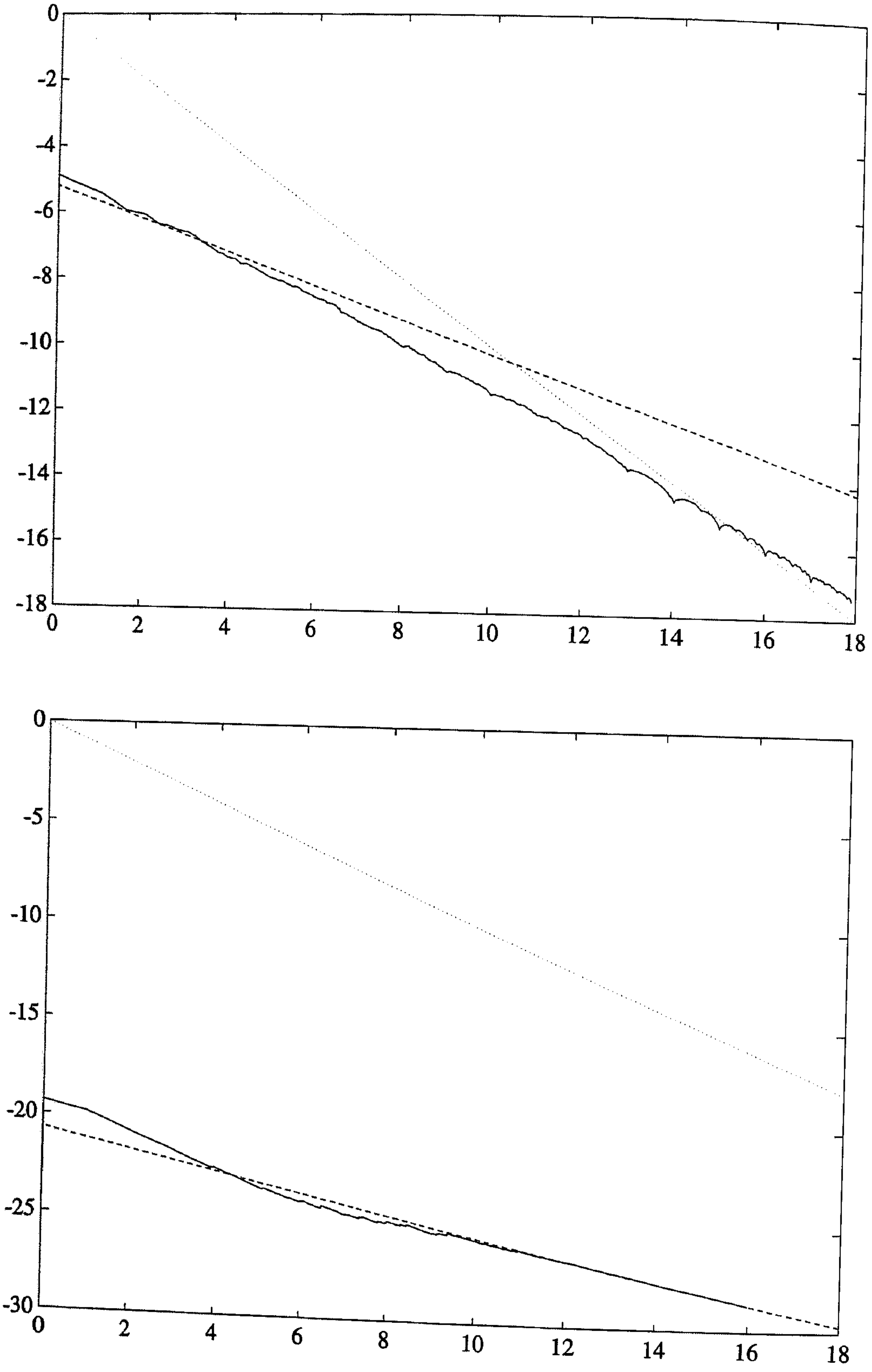


Fig. 4. Log-log (base 10) plot of discrepancy T_N for a Sobol' sequence in 4 dimensions (top) and for 16 dimensions (bottom)

no significance). Since the plots are on a log-log scale, these reference curves are lines of slope $-1/2$ and -1.

The figure for small dimension (4) and for large N, shows that $T_N \approx O(N^{-1})$. We have not tried to match the $\log N$ terms, since we do not seem to be able to detect them reliably. On the other hand, the graph for large dimension (16), shows that for moderate sized N, $T_N \approx O(N^{-1/2})$, with a constant that agrees with the discrepancy of a random sequence. Eventually, as N gets very large, T_N must be nearly of size $O(N^{-1})$. The agreement between discrepancy of quasi-random and random sequences for large d and moderate N has not been explained.

Next we consider one- and two-dimensional projections of a quasi-random sequence. Many, but not all, quasi-random sequences have the following special property. All one-dimensional projections of the sequence (*i.e.*, all single components) are equally well distributed. The Sobol' sequence, for example, has this property. This implies that any function f that consists of a sum of one-dimensional functions will be well integrated by quasi-Monte Carlo, in other words, functions of the form

$$f(\boldsymbol{x}) = \sum_{n=1}^{d} f_n(x_n). \tag{5.27}$$

In particular, quasi-Monte Carlo performs very well for linear functions

$$f(\boldsymbol{x}) \approx \sum_{i=1}^{d} a_i x_i. \tag{5.28}$$

Now consider two-dimensional projections. Figure 5 shows a 'good' and a 'bad' two-dimensional projection of a Sobol' sequence. The top plot of the 'good' projection is very uniform, while the bottom plot of the 'bad' projection shows gaps and a repetitive structure to the points. For this projection, the holes would be filled in as the further points of the sequence are laid down. At certain special values of N, the sequence would be quite uniform. We have observed patterns of this sort in a variety of quasi-random sequences (Morokoff and Caflisch 1994), but they are not expected to occur in the randomized quasi-random points of Owen (1995, 1997).

6. Quasi-Monte Carlo techniques

In this section, two techniques are described for overcoming the smoothness and high-dimension limitations of quasi-Monte Carlo. The first method is a smoothing method and will be demonstrated on the acceptance–rejection method, which involves a characteristic function in its standard form, coming from the decision to accept or reject. The second technique is a rearrangement of dimension, so as to put most of the weight of an integral into the leading dimensions of the quasi-random sequence.

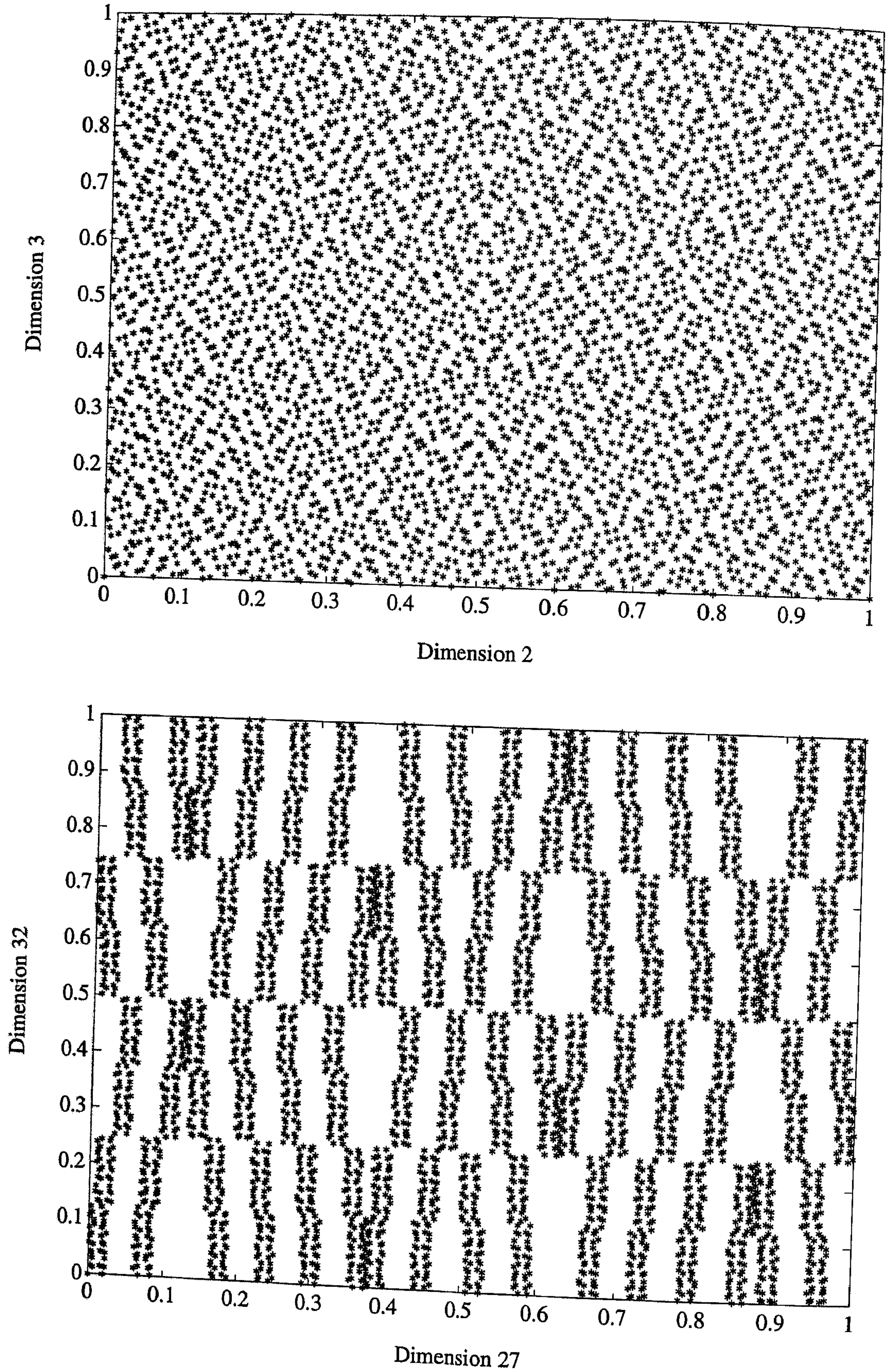


Fig. 5. Good and bad two-dimensional projections of a Sobol' sequence

6.1. Smoothing

The effectiveness of quasi-Monte Carlo is often lost on problems involving discontinuities, such as decisions in the acceptance–rejection method. One way to overcome this difficulty is to smooth out the integrand without changing the value of the integral. Here we describe a smoothed version of the acceptance–rejection method and compare it with standard acceptance–rejection, when used with a quasi-random sequence.

Standard acceptance–rejection can be expressed in an integration problem in the following way. Consider the integral of a function f multiplying a function p on the unit interval, with $0 \leq p(x) \leq 1$. Rewrite the integral as

$$\int_0^1 f(x)p(x)\,\mathrm{d}x = \int_0^1 \int_0^1 f(x)\chi(y < p(x))\,\mathrm{d}y\,\mathrm{d}x. \tag{6.1}$$

The acceptance–rejection method is a slight modification of the usual Monte Carlo formula for this integral, that is,

- sample (x, y) uniformly
- weight $w = \chi(y < p(x)) = 1$ (accept), if $y < p(x)$
- weight $w = \chi(y < p(x)) = 0$ (reject), if $y > p(x)$
- repeat until number of accepted points is N.

The Monte Carlo quadrature formula is then

$$I_N = N^{-1} \sum_{i=1}^{M} w_i f(x_i), \tag{6.2}$$

in which

$$N \approx \sum_{i=1}^{M} w_i. \tag{6.3}$$

The smoothed acceptance–rejection method (Spanier and Maize 1994, Moskowitz and Caflisch 1996) is formulated by replacing the discontinuous function $\chi(y < p(x))$ by a smooth function $q(x, y)$ satisfying

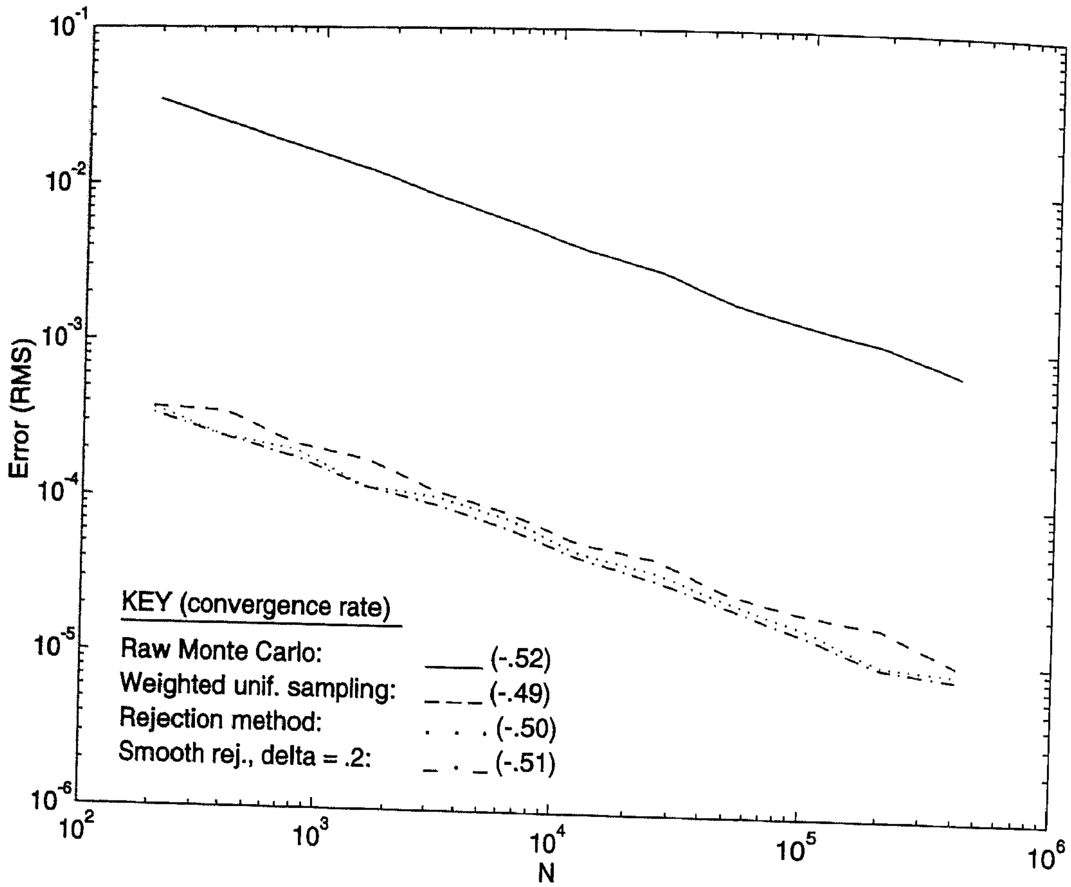
$$\int_0^1 q(x, y)\,\mathrm{d}y = p(x). \tag{6.4}$$

For example, one choice of q is a piecewise linear function:

$$q(x, y) = \begin{cases} 1 & \text{if } y < p(x) - \epsilon, \\ 0 & \text{if } y > p(x) + \epsilon, \\ \text{linear} & p(x) - \epsilon < y < p(x) + \epsilon. \end{cases} \tag{6.5}$$

The integral can then be rewritten as

$$\int_0^1 f(x)p(x)\,\mathrm{d}x = \int_0^1 \int_0^1 f(x)q(x, y)\,\mathrm{d}y\,\mathrm{d}x, \tag{6.6}$$



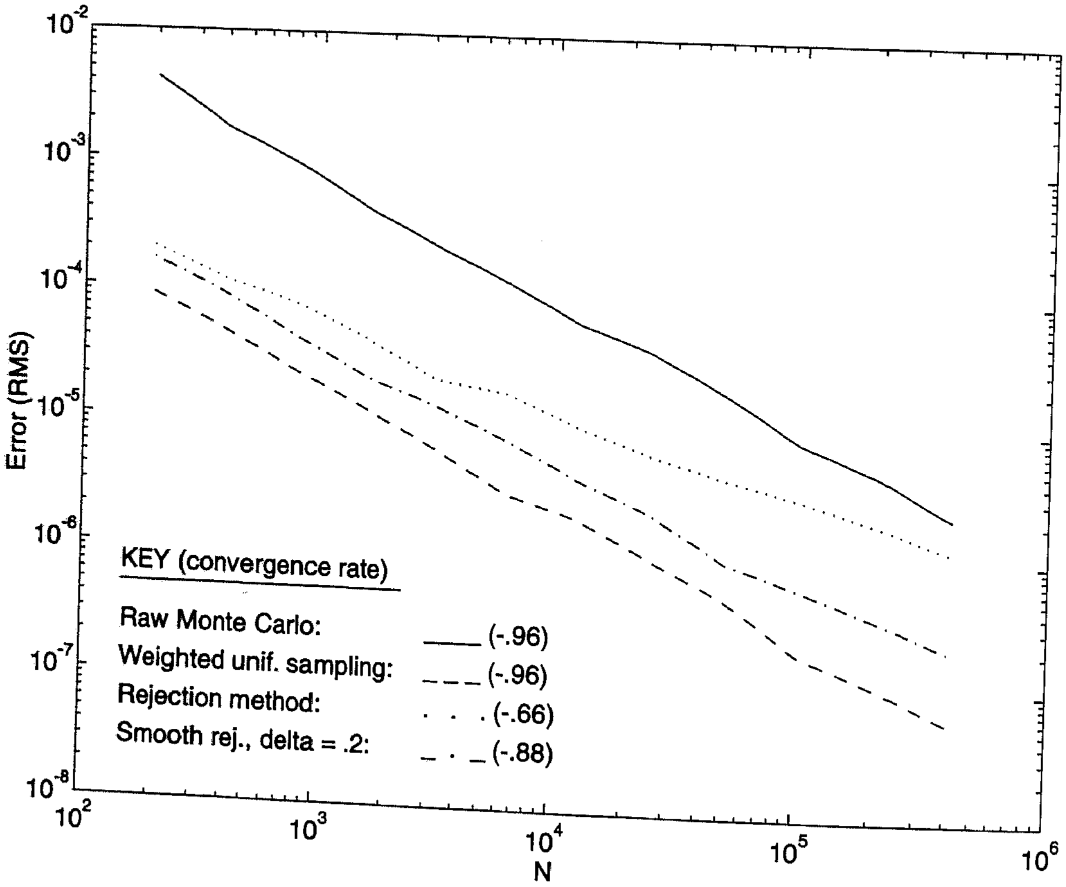


Fig. 6. Smoothed acceptance–rejection for Monte Carlo (top) and quasi-Monte Carlo (bottom)

and the sampling procedure is the following:

- sample (x, y) uniformly
- set the weight $w = q(x, y)$
- repeat until

$$\sum_{i=1}^{M} q(x_i, y_i) \approx N. \tag{6.7}$$

The quadrature formula, using smoothed acceptance–rejection, is then

$$J_N = N^{-1} \sum_{i=1}^{M} q(x_i, y_i) f(x_i), \tag{6.8}$$

in which

$$\sum_{i=1}^{M} q(x_i, y_i) \approx N. \tag{6.9}$$

Figure 6 shows the results of standard and smoothed acceptance–rejection for Monte Carlo using both a pseudo-random sequence and a quasi-random sequence. The integral evaluated here (Moskowitz and Caflisch 1996) has the form

$$I = \int_{I^7} f(\boldsymbol{x}) p(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}, \tag{6.10}$$

in which $I^7 = [0,1]^7$ and

$$\begin{aligned} p(\boldsymbol{x}) &= \exp\left\{ -\left(\sin^2\left(\frac{\pi}{2} x_1\right) + \sin^2\left(\frac{\pi}{2} x_2\right) + \sin^2\left(\frac{\pi}{2} x_3\right) \right) \right\}, \\ f(\boldsymbol{x}) &= \mathrm{e} \arcsin\left(\sin(1) + \left[\frac{x_1 + \cdots + x_7}{200} \right] \right). \end{aligned} \tag{6.11}$$

The integral I is evaluated in several ways. First, by 'raw Monte Carlo', by which we mean evaluation of the product $f(x)p(x)$ at points from a uniformly distributed sequence. Second, points are sampled from the (unnormalized) density function p using acceptance–rejection, then f is evaluated at these points. Third, standard acceptance–rejection is replaced by smoothed acceptance–rejection. Finally, all three of these methods are performed both with pseudo-random and quasi-random points. These numerical results show the following.

- Importance sampling, using acceptance–rejection, reduces variance and leads to smaller errors for both Monte Carlo and quasi-Monte Carlo.
- For Monte Carlo, smoothing has little effect on errors, since it does not change the variance.
- Without importance sampling, the quasi-Monte Carlo method converges at a rapid rate, but with a constant that is larger than for the method with importance sampling.

- For quasi-Monte Carlo, the error for unsmoothed acceptance–rejection decreases at a slower rate, because of discontinuities involved in the acceptance–rejection decision.
- For quasi-Monte Carlo, the rapid convergence rate is regained using smoothing.

Although smoothed acceptance–rejection regains much of the effectiveness of quasi-Monte Carlo, it entails a loss of efficiency. Since fractional weights are involved, more accepted samples are required to get total weight N. Moreover, we do not have an effective quasi-Monte Carlo method for the Metropolis algorithm (Kalos and Whitlock 1986), which is a stochastic process involving acceptance–rejection.

6.2. Dimension reduction: Brownian bridge method

For problems in a dimension of moderate size, quasi-Monte Carlo provides a significant speed-up over standard Monte Carlo. Many of the most important applications of Monte Carlo, however, involve integrals of a very high dimension. One class of examples comprises path integrals involving Brownian motion $b(t)$. A typical example is a Feynman–Kac integral (Karatzas and Shreve 1991) of the form

$$I = E\left[\int_0^T f(b(t)) \exp\left\{\int_0^t \lambda(b(s), s)\, \mathrm{d}s\right\} \mathrm{d}t\right]. \tag{6.12}$$

In order to evaluate this expectation by Monte Carlo, we need to discretize the time in the Brownian motion to obtain a random walk with M steps. Then the integral becomes an integral over the M steps of a random walk, each of which is distributed by a normal distribution. An accurate representation of the integral requires a large number M of time-steps, which results in an integral of large-dimension M. Because of the high dimension, we find that quasi-Monte Carlo loses much of its effectiveness on such a problem. The main point of this section is to show how a rearrangement of the dimensions can regain the effectiveness of quasi-Monte Carlo for this problem.

The standard discretization of a random walk is to represent the position $b(t+\Delta t)$ in terms of the previous position $b(t)$ by the formula

$$b(t+\Delta t) = b(t) + \sqrt{\Delta t}\, \nu, \tag{6.13}$$

in which ν is an $N(0,1)$ random variable. Using a sequence of independent samples of ν, we can generate the random walk sequentially by

$$y_0 = 0,\ y_1 = b(\Delta t),\ y_2 = b(2\Delta t), \dots. \tag{6.14}$$

This representation leads to the M-dimensional integral discussed above.

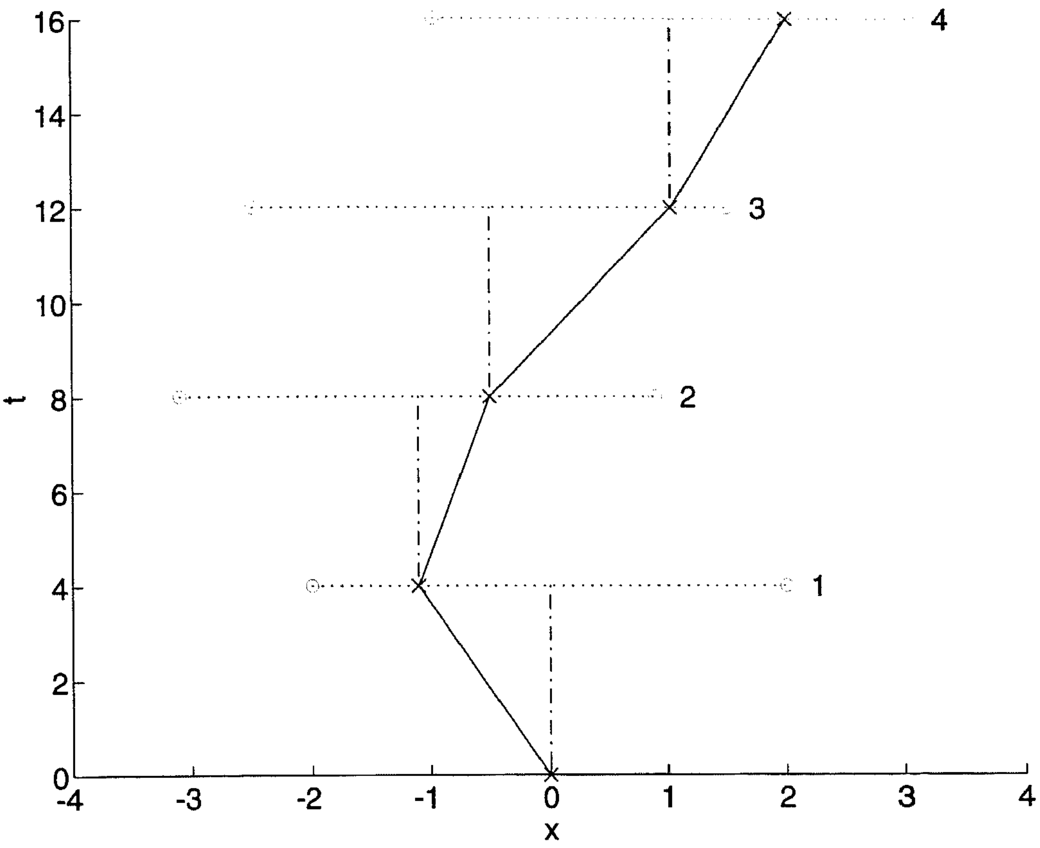


Fig. 7. Standard discretization of random walk

The effectiveness of quasi-Monte Carlo will be regained using an alternative representation of the random walk, the Brownian bridge discretization, which was first introduced as a quasi-Monte Carlo technique by Moskowitz and Caflisch (1996). This representation relies on the following Brownian bridge formula (Karatzas and Shreve 1991) for $b(t+\Delta t_1)$, given $b(t)$ and $b(T=t+\Delta t_1+\Delta t_2)$:

$$b(t+\Delta t_1) = ab(t) + (1-a)b(T) + c\nu, \tag{6.15}$$

in which

$$a = \Delta t_2/(\Delta t_1+\Delta t_2), \qquad c = \sqrt{a\Delta t_1}. \tag{6.16}$$

Using this representation, the random walk can be generated by successive subdivision. Suppose for simplicity that M is a power of 2. Then generate the random walk in the following order:

$$y_0 = 0,\ y_M,\ y_{M/2},\ y_{M/4},\ y_{3M/4},\ \dots . \tag{6.17}$$

The standard discretization and Brownian bridge discretization are represented schematically in Figures 7 and 8.

The significance of this representation is that it first chooses the large time-steps over which the changes in $b(t)$ are large. Then it fills in the small time-steps in between, in which the changes in $b(t)$ are quite small. The

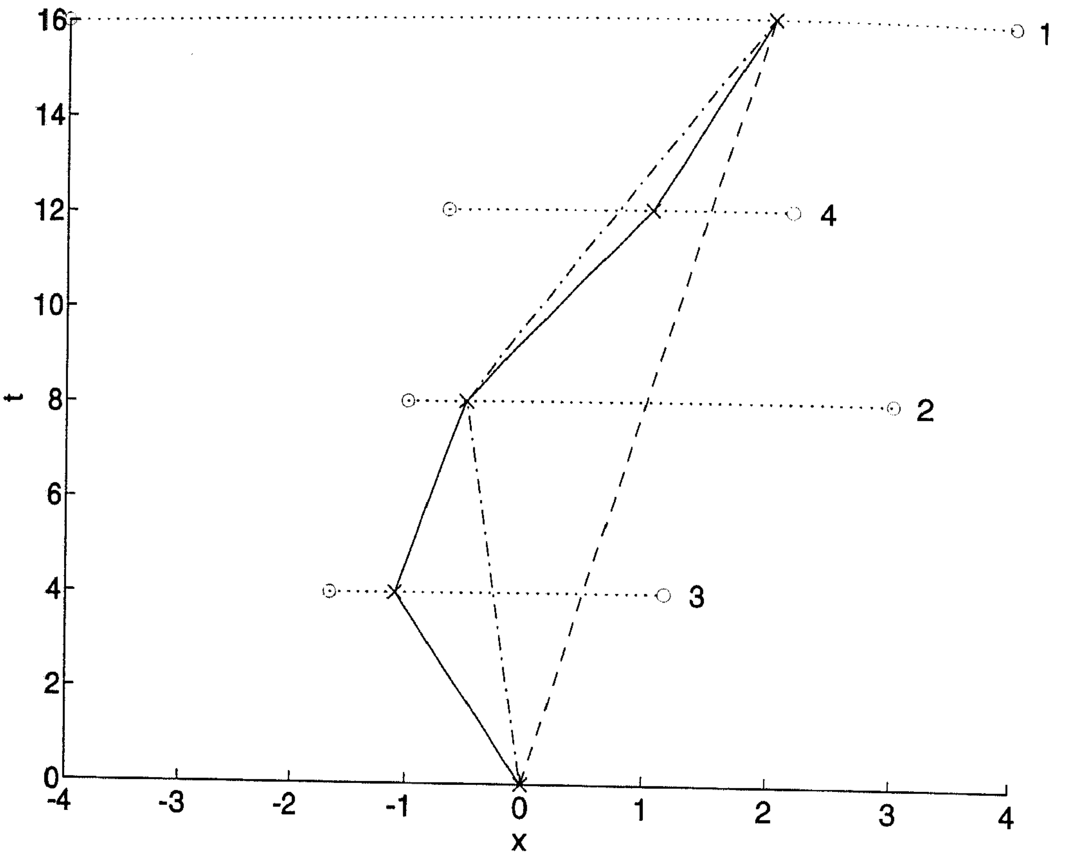


Fig. 8. Brownian bridge discretization of random walk

advantage of this representation is that it concentrates the variance into the early, large time-steps. This is much like a principal component analysis (Acworth, Broadie and Glasserman 1997).

Although the actual dimension of the problem is not changed, in some sense the effective dimension of the problem is lowered, so that quasi-Monte Carlo retains its effectiveness. To make this statement quantitative, suppose that, at some given value of N, the discrepancy is of size N^{-1} for dimension d (omitting logarithmic terms for simplicity), but is of size $N^{-1/2}$ for the remaining dimensions. We expect that the integration error is roughly of the size of the variance times the discrepancy. Using the Brownian bridge discretization, the variance over the first d dimensions is σ_0, which is about the same size as the original value of σ, whereas the variance over the remaining $M-d$ dimensions is σ_1, which is much smaller, that is,

$$\sigma_1 \ll \sigma_0 \approx \sigma. \tag{6.18}$$

Denote ε_s and ε_{bb} to be the errors for quasi-Monte Carlo using the standard discretization and the Brownian bridge discretization, respectively. Then, approximately,

$$\varepsilon_s = \sigma N^{-1/2}, \qquad \varepsilon_{bb} = \sigma_0 N^{-1} + \sigma_1 N^{-1/2}. \tag{6.19}$$

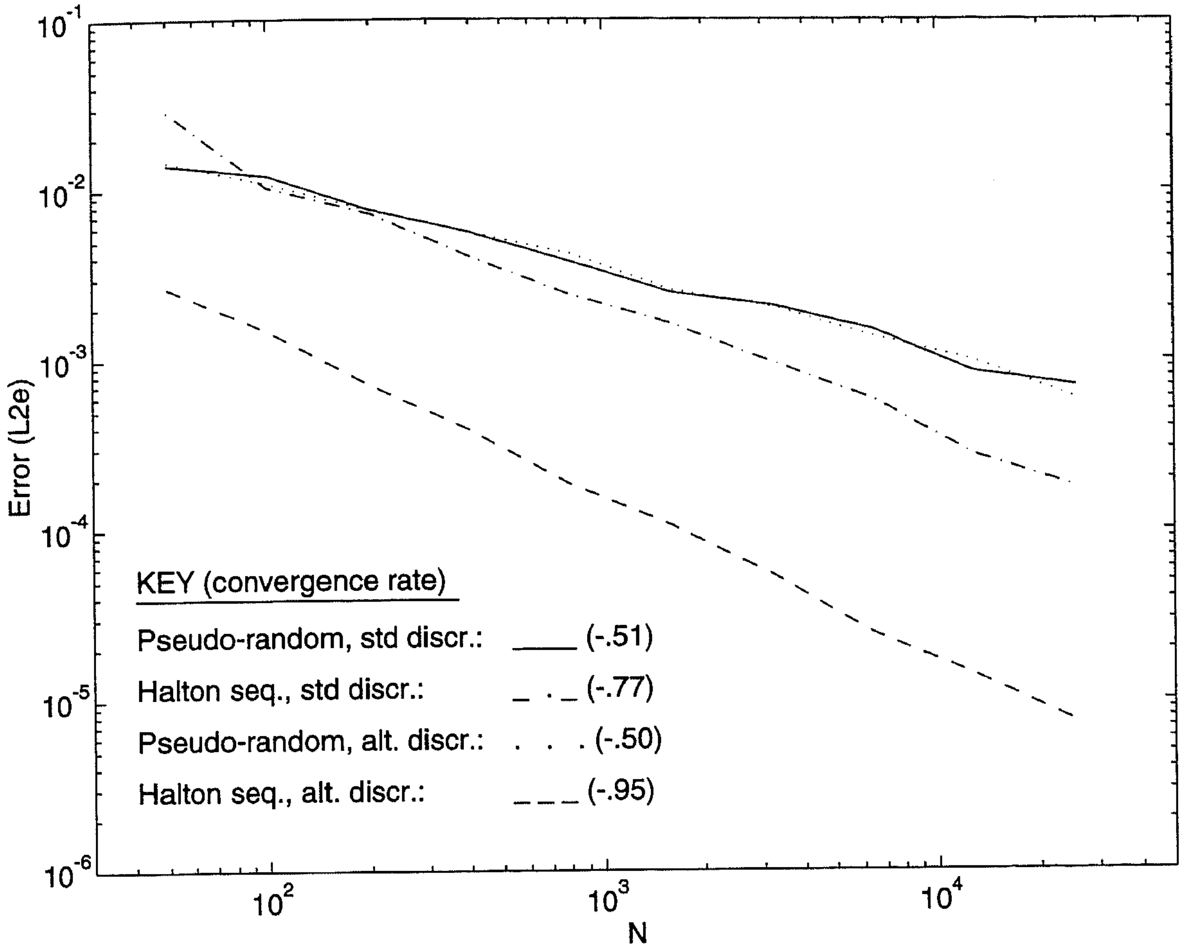


Fig. 9. Log-log plot for Feynman–Kac integral, 75 runs, $T = 0.08$, $m = 32$

The ordering (6.18) then implies that

$$\varepsilon_{bb} \ll \varepsilon_s. \tag{6.20}$$

This shows that the Brownian bridge discretization can provide a significant improvement in quasi-Monte Carlo integration for path integral problems.

As an example, consider an integral of the form (6.12), in which

$$\begin{aligned} \lambda(x,t) &= \left(\frac{1}{t+1} + \frac{1}{x^2+1} - \frac{4x^2}{(x^2+1)^2}\right), \\ f(x) &= \frac{1}{x^2+1}. \end{aligned} \tag{6.21}$$

According to the Feynman–Kac formula (Karatzas and Shreve 1991), the integral $I = F(x,t)$, in which x is the starting point of the Brownian motion, solves the following linear parabolic differential equation:

$$\frac{\partial F}{\partial t}(x,t) = \frac{1}{2}\frac{\partial^2 F}{\partial x^2}(x,t) + \lambda(x,t)\frac{\partial F}{\partial x}(x,t), \\ \text{with } F(x,0) = f(x). \tag{6.22}$$

The exact solution is $F(x,t) = (t+1)(x^2+1)^{-1}$.

Computational results for this problem of Moskowitz and Caflisch (1996) are presented in Figure 9, at time $T = 0.08$ with $M = 32$. The results show the following.

- For Monte Carlo using pseudo-random numbers, the error is the same for the standard and Brownian bridge discretizations. This is because the variance of the two is the same.
- For quasi-Monte Carlo with the standard discretization, the error is only a little less than for standard Monte Carlo; *i.e.*, the effectiveness of quasi-Monte Carlo has been lost for this high-dimension ($M = 32$).
- With the Brownian bridge discretization, the error for quasi-Monte Carlo is substantially reduced, in terms of both the rate of convergence and the constant; *i.e.*, the effectiveness of quasi-Monte Carlo has been regained.

It is necessary to use the transformation method (Section 3.3) rather than the Box–Muller method (Section 3.4), since the latter method has large gradients (Morokoff and Caflisch 1993). Similar results have been obtained on problems from computational finance by Caflisch, Morokoff and Owen (1997*b*).

7. Monte Carlo methods for rarefied gas dynamics

Computations for rarefied gas dynamics, as occur in the outer parts of the atmosphere, are difficult since the gas is represented by a distribution function over space, time and molecular velocity. For general problems, the only effective method has been a Monte Carlo method, as described below. One reason for including this application of Monte Carlo in this survey is that formulation of an effective Monte Carlo method in the fluid dynamic limit is an open problem.

7.1. The Boltzmann equation

The kinetic theory of gases describes the behaviour of a gas in which the density is not too large, so that the only interactions between gas particles are binary collisions. The resulting nonlinear Boltzmann equation,

$$\frac{\partial}{\partial t}F + \boldsymbol{\xi}\cdot\frac{\partial}{\partial \mathbf{x}}F = \varepsilon^{-1}Q(F,F), \tag{7.1}$$

for the molecular distribution function $F(\mathbf{x},\boldsymbol{\xi},t)$, is a basic equation of nonequilibrium statistical mechanics (Cercignani 1988, Chapman and Cowling 1970). The collision operator is

$$Q(F,F)(\boldsymbol{\xi}) = \int (F(\boldsymbol{\xi}_1')F(\boldsymbol{\xi}') - F(\boldsymbol{\xi}_1)F(\boldsymbol{\xi}))B(\omega, |\boldsymbol{\xi}_1 - \boldsymbol{\xi}|)\,\mathrm{d}\omega\,\mathrm{d}\boldsymbol{\xi}_1, \tag{7.2}$$

in which $\boldsymbol{\xi}, \boldsymbol{\xi}_1$ represent velocities before a collision, $\boldsymbol{\xi}', \boldsymbol{\xi}_1'$ represent velocities after a collision, and the collision parameters are represented by $\omega \in S^2$, where S^2 is the unit sphere in $\mathbb{R}^3$. The parameter ε is a measure of the 'mean free time', which is defined as the characteristic time between collisions of a particle.

For a molecular distribution F, the macroscopic (or fluid dynamic) variables are the density ρ, velocity $\boldsymbol{u}$ and temperature T defined by

$$\begin{aligned} \rho &= \int F \,\mathrm{d}\boldsymbol{\xi}, \\ \boldsymbol{u} &= \rho^{-1} \int \boldsymbol{\xi} F \,\mathrm{d}\boldsymbol{\xi}, \\ T &= (3\rho)^{-1} \int |\boldsymbol{\xi} - \boldsymbol{u}|^2 F \,\mathrm{d}\boldsymbol{\xi}. \end{aligned} \tag{7.3}$$

The importance of these moments of F is that they correspond to conserved quantities, namely, mass, momentum and energy, for the collisional process, which is expressed as follows:

$$\begin{aligned} \int Q(F,F)\,\mathrm{d}\boldsymbol{\xi} &= 0, \\ \int \boldsymbol{\xi} Q(F,F)\,\mathrm{d}\boldsymbol{\xi} &= 0, \\ \int |\boldsymbol{\xi}|^2 Q(F,F)\,\mathrm{d}\boldsymbol{\xi} &= 0. \end{aligned} \tag{7.4}$$

These also provide the degrees of freedom for the equilibrium case, the Maxwellian distributions

$$M(\boldsymbol{\xi},\ \rho,\ \boldsymbol{u},\ T) = \rho(2\pi T)^{-3/2} \exp\left(-|\boldsymbol{\xi} - \boldsymbol{u}|^2/2T\right). \tag{7.5}$$

In the limit $\varepsilon \to 0$, the solution F of (7.1) is given by the Hilbert (or Chapman–Enskog) expansion, in which the leading term is a Maxwellian distribution of the form (7.5), in which ρ, $\boldsymbol{u}$, T satisfy the compressible Euler (or Navier–Stokes) equations of fluid dynamics (Chapman and Cowling 1970). The Euler equations are

$$\begin{aligned} \rho_t + \boldsymbol{\nabla} \cdot (\boldsymbol{u}\rho) &= 0, \\ (\rho \boldsymbol{u})_t + \boldsymbol{\nabla} \cdot (\boldsymbol{u}\boldsymbol{u}\rho) + \boldsymbol{\nabla}(\rho T) &= 0, \\ \left(\frac{1}{2}\rho|\boldsymbol{u}|^2 + \frac{3}{2}\rho T\right)_t + \boldsymbol{\nabla} \cdot \left(\boldsymbol{u}\left(\frac{1}{2}\rho|\boldsymbol{u}|^2 + \frac{5}{2}\rho T\right)\right) &= 0. \end{aligned} \tag{7.6}$$

This expansion is not valid in layers around shocks, boundaries and nonequilibrium initial data, where special boundary, shock or initial layer expansions can be constructed (Caflisch 1983). By varying this limit, taking the velocity $\boldsymbol{u}$ to be size ε as well, the compressible equations are replaced by the incompressible fluid equations (Bardos, Golse and Levermore 1991, 1993, De Masi, Esposito and Lebowitz 1989).

7.2. Particle methods

In particle methods for transport theory, the distribution $F(\mathbf{x},\boldsymbol{\xi},t)$ is represented as the sum

$$F_N(\mathbf{x},\boldsymbol{\xi},t)=\sum_{n=1}^{N}\delta(\boldsymbol{\xi}-\boldsymbol{\xi}_n(t))\delta(\mathbf{x}-\mathbf{x}_n(t)), \tag{7.7}$$

and the positions $\mathbf{x}_n(t)$ and velocities $\boldsymbol{\xi}_n(t)$ are evolved in time to simulate the effects of convection and collisions. The most common particle methods use random collisions between a reasonable number of particles (*e.g.*, 10^3-10^6) to simulate the dynamics of many particles (*e.g.*, 10^{23}).

In the direct simulation Monte Carlo (DSMC) method pioneered by Bird (1976, 1978), the numerical method is designed to simulate the physical processes as closely as possible. This makes it easy to understand and to insert new physics; it is also numerically robust. First, space and time are discretized into spatial cells of size Δx^3 and time-steps of duration Δt. In each time-step the evolution is divided into two steps: transport and collisions. For the transport step each particle is moved from position $\mathbf{x}_n(t)$ to $\mathbf{x}_n(t+\Delta t)=\mathbf{x}_n(t)+\Delta t\,\boldsymbol{\xi}_n(t)$.

In the collision step, random collisions are performed between particles within each spatial bin. In each collision, particles $\boldsymbol{\xi}_n$ and $\boldsymbol{\xi}_m$ are chosen randomly from the full set of particles in a bin, with probability p_{mn} given by

$$p_{mn}=\frac{S(|\boldsymbol{\xi}_m-\boldsymbol{\xi}_n|)}{\sum_{1\le i\le j\le N_0}S(|\boldsymbol{\xi}_i-\boldsymbol{\xi}_j|)}, \tag{7.8}$$

in which N_0 is the number of particles in the spatial bin, and

$$S(|\boldsymbol{\xi}_i-\boldsymbol{\xi}_j|)=\int_{S^2}B(\omega,|\boldsymbol{\xi}_m-\boldsymbol{\xi}_n|)\,\mathrm{d}\omega \tag{7.9}$$

is the total collision rate between particles of velocity $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$. As written, this choice requires $O(N^2)$ operations to evaluate the sum in the denominator in (7.8). By a standard acceptance–rejection scheme, however, each choice can be made in $O(1)$ steps, so that the total method requires only $O(N)$ steps.

Next the collision parameters ω are randomly chosen from a uniform distribution on S^2. The outcome of the collision is two new velocities $\boldsymbol{\xi}_n'$ and $\boldsymbol{\xi}_m'$ which replace the old velocities $\boldsymbol{\xi}_n$ and $\boldsymbol{\xi}_m$.

The number of collisions performed in each time-step has been determined by several methods. In the original 'time-counter' (TC) method, a collision time Δt_c is determined. It is equal to one over the frequency for that collision type, that is,

$$\Delta t_c = 2(nN_0S(|\xi_m - \xi_n|))^{-1}, \tag{7.10}$$

in which n is the number density of particles. This is added to the time-counter $t_c = \sum \Delta t_c$. In the time interval of length Δt beginning at t, collisions are continued until t_c exceeds the final time, that is, until $t_c \geq t + \Delta t$. For N particles this method has operation count $O(N)$.

The unlikely possibility of choosing a collision with very small frequency can result in a large collisional time-step that may cause relatively large errors. To remove such errors, Bird (1976) has developed a 'no time-counter' (NTC) method. This method uses a maximum collision probability $S_{\max}$. The number of collisions to be performed in time-step dt is chosen as if the collision frequency were exactly $S_{\max}$ for all collision pairs. For each selection of a collision pair, the collision between ξ_m and ξ_n is then performed with probability $S(|\xi_m - \xi_n|)/S_{\max}$, as in an acceptance–rejection scheme. The DSMC method with the time-counter or no-time-counter algorithm has been enormously successful. A rigorous convergence result for DSMC was proved by Wagner (1992).

Several related methods have been developed by other researchers, including Koura (1986), Nanbu (1986), and Goldstein, Sturtevant and Broadwell (1988). Additional modifications of Nanbu's method, including use of quasi-random sequences, have been developed by Babovsky, Gropengiesser, Neunzert, Struckmeier and Wiesen (1990).

7.3. Methods with the correct diffusion limit

One of the most difficult problems of transport theory is that there may be wide variation of mean free paths within a single problem. In regions where the mean free path is large, there are few collisions, so that large numerical time-steps may be taken, whereas in regions where the mean free path is small, the time- and space-steps must be small. Thus the highly collisional regions determine the numerical time-step, which may make computations impractically slow. On the other hand, much of the extra effort in the collisional regions seems wasted, since in those regions the gas will be nearly in fluid dynamic equilibrium, so that a fluid dynamic description (7.6) should be valid.

A partial remedy to this problem is to use a numerical method that converts to a numerical method for the correct fluid equations in regions where the mean free time is small. This allows large, fluid dynamic time-steps in the collisional region and large collisional time-steps in the large mean free time region.

Consider a discretization of the Boltzmann equation (7.1) with discrete space and time scales Δx and Δt. If Δx, Δt are much smaller than ε, then the Boltzmann solution is well resolved. On the other hand, in regions of small mean free path, we want to let the discretization scale be much larger than the collision scale ε. So we consider the limit in which the numerical parameters Δx, Δt are held fixed, while $\varepsilon \to 0$. We say that the discretized equation has the correct diffusion limit, if in this limit the solution of the discrete equations for (7.1) goes to the solution for a discretization of the fluid equation (7.6).

In the context of linear neutron transport, Larsen and co-workers (Larsen, Morel and Miller 1987, Borgers, Larsen and Adams 1992) have investigated the diffusion limit for a variety of difference schemes and have found that many of them have the correct diffusion limit only in special regimes. They have constructed some alternative methods that always have the correct diffusion limit. Jin and Levermore (1993) have applied a similar procedure to an interface problem to get a constraint on the quadrature set for the discretization of a scattering integral. Another class of methods that uses information from the diffusion limit to improve a numerical transport methods is the family of 'diffusion synthetic acceleration methods' (Larsen 1984). For the Broadwell model of the Boltzmann equation, finite difference numerical methods with the correct diffusion limit have been developed (Caflisch, Jin and Russo 1997*a*, Jin, Pareschi and Toscani 1998). These use implicit differencing for the collision step, because it is a stiff problem with rate ε^{-1}.

For the nonlinear Boltzmann equation, on the other hand, Monte Carlo methods are the most practical because of the large number of degrees of freedom. Application of the methods would require some kind of implicit step to handle the collisions, but no such method has been formulated so far. We believe that a method that uses fluid dynamic information to improve the collisional step for rarefied gas dynamic computations would have great impact. Some important partial steps in this direction have been taken (Gabetta, Pareschi and Toscani 1997), but they have not yet been extended to particle methods such as DSMC.

REFERENCES

P. Acworth, M. Broadie and P. Glasserman (1997), A comparison of some Monte Carlo and quasi Monte Carlo techniques for option pricing, in *Monte Carlo and Quasi-Monte Carlo Methods 1996* (G. Larcher, H. Niederreiter, P. Hellekalek and P. Zinterhof, eds), Springer.

H. Babovsky, F. Gropengiesser, H. Neunzert, J. Struckmeier and J. Wiesen (1990), 'Application of well-distributed sequences to the numerical simulation of the Boltzmann equation', *J. Comput. Appl. Math.* **31**, 15–22.

C. Bardos, F. Golse and C. D. Levermore (1991), 'Fluid dynamic limits of kinetic equations: I. Formal derivations', *J. Statist. Phys.* **63**, 323–344.

C. Bardos, F. Golse and C. D. Levermore (1993), 'Fluid dynamic limits of kinetic equations: II. Convergence proofs for the Boltzmann equation', *Comm. Pure Appl. Math.* **46**, 667–753.

G. A. Bird (1976), *Molecular Gas Dynamics*, Oxford University Press.

G. A. Bird (1978), 'Monte Carlo simulation of gas flows', *Ann. Rev. Fluid Mech.* **10**, 11–31.

C. Borgers, E. W. Larsen and M. L. Adams (1992), 'The asymptotic diffusion limit of a linear discontinuous discretization of a two-dimensional linear transport equation', *J. Comput. Phys.* **98**, 285–300.

P. Bratley, B. L. Fox and H. Niederreiter (1994), 'Algorithm 738 - Programs to generate Niederreiter's discrepancy sequences', *ACM Trans. Math. Software* **20**, 494–495.

R. E. Caflisch (1983), Fluid dynamics and the Boltzmann equation, in *Nonequilibrium Phenomena I: The Boltzmann equation* (E. W. Montroll and J. L. Lebowitz, eds), Vol. 10 of *Studies in Statistical Mechanics*, North Holland, pp. 193–223.

R. E. Caflisch, S. Jin and G. Russo (1997*a*), 'Uniformly accurate schemes for hyperbolic systems with relaxation', *SIAM J. Numer. Anal.* **34**, 246–281.

R. E. Caflisch, W. Morokoff and A. B. Owen (1997*b*), 'Valuation of mortgage-backed securities using Brownian bridges to reduce effective dimension', *J. Comput. Finance* **1**, 27–46.

C. Cercignani (1988), *The Boltzmann Equation and its Applications*, Springer.

S. Chapman and T. G. Cowling (1970), *The Mathematical Theory of Non-Uniform Gases*, Cambridge University Press.

A. De Masi, R. Esposito and J. L. Lebowitz (1989), 'Incompressible Navier–Stokes and Euler limits of the Boltzmann equation', *Comm. Pure Appl. Math.* **42**, 1189–1214.

H. Faure (1982), 'Discrépance de suites associées à un système de numération (en dimension *s*)', *Acta Arithmetica* **41**, 337–351.

W. Feller (1971), *An Introduction to Probability Theory and its Applications: Vol. I*, Wiley.

E. Gabetta, L. Pareschi and G. Toscani (1997), 'Relaxation schemes for nonlinear kinetic equations', *SIAM J. Numer. Anal.* **34**, 2168–2194.

D. Goldstein, B. Sturtevant and J. E. Broadwell (1988), Investigations of the motion of discrete-velocity gases, in *Rarefied Gas Dynamics: Theoretical and Computational Techniques* (E. P. Muntz, D. P. Weaver and D. H. Campbell, eds), Vol. 118 of *Progress in Aeronautics and Astronautics*, Proceedings of the 16th International Symposium on Rarefied Gas Dynamics, pp. 100–117.

J. H. Halton (1960), 'On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals', *Numer. Math.* **2**, 84–90.

J. M. Hammersley and D. C. Handscomb (1965), *Monte Carlo Methods*, Wiley, New York.

C. Haselgrove (1961), 'A method for numerical integration', *Math. Comput.* **15**, 323–337.

F. J. Hickernell (1996), 'The mean square discrepancy of randomized nets', *ACM Trans. Model. Comput. Simul.* **6**, 274–296.

F. J. Hickernell (1998), 'A generalized discrepancy and quadrature error bound', *Math. Comput.* **67**, 299–322.

R. V. Hogg and A. T. Craig (1995), *Introduction to Mathematical Statistics*, Prentice Hall.

L. K. Hua and Y. Wang (1981), *Applications of Number Theory to Numerical Analysis*, Springer, Berlin/New York.

S. Jin and C. D. Levermore (1993), 'Fully-discrete numerical transfer in diffusive regimes', *Transp. Theory Statist. Phys.* **22**, 739–791.

S. Jin, L. Pareschi and G. Toscani (1998), 'Diffusive relaxation schemes for discrete-velocity kinetic equations'. Preprint.

M. H. Kalos and P. A. Whitlock (1986), *Monte Carlo Methods, Vol. I: Basics*, Wiley, New York.

I. Karatzas and S. E. Shreve (1991), *Brownian Motion and Stochastic Calculus*, Springer, New York.

W. J. Kennedy and J. E. Gentle (1980), *Statistical Computing*, Dekker, New York.

K. Koura (1986), 'Null-collision technique in the direct-simulation Monte Carlo method', *Phys. Fluids* **29**, 3509–3511.

L. Kuipers and H. Niederreiter (1974), *Uniform Distribution of Sequences*, Wiley, New York.

E. W. Larsen (1984), 'Diffusion-synthetic acceleration methods for discrete-ordinates problems', *Transp. Theory Statist. Phys.* **13**, 107–126.

E. W. Larsen, J. E. Morel and W. F. Miller (1987), 'Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes', *J. Comput. Phys.* **69**, 283–324.

G. Lepage (1978), 'A new algorithm for adaptive multidimensional integration', *J. Comput. Phys.* **27**, 192–203.

G. Marsaglia (1991), 'Normal (Gaussian) random variables for supercomputers', *J. Supercomputing* **5**, 49–55.

W. Morokoff and R. E. Caflisch (1993), 'A quasi-Monte Carlo approach to particle simulation of the heat equation', *SIAM J. Numer. Anal.* **30**, 1558–1573.

W. Morokoff and R. E. Caflisch (1994), 'Quasi-random sequences and their discrepancies', *SIAM J. Sci. Statist. Comput.* **15**, 1251–1279.

W. Morokoff and R. E. Caflisch (1995), 'Quasi-Monte Carlo integration', *J. Comput. Phys.* **122**, 218–230.

B. Moskowitz and R. E. Caflisch (1996), 'Smoothness and dimension reduction in quasi-Monte Carlo methods', *J. Math. Comput. Modeling* **23**, 37–54.

K. Nanbu (1986), Theoretical basis of the direct simulation Monte Carlo method, in *Proceedings of the 15th International Symposium on Rarefied Gas Dynamics* (V. Boffi and C. Cercignani, eds), Vol. I, pp. 369–383.

H. Niederreiter (1992), *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia, PA.

A. B. Owen (1995), Randomly permuted (t, m, s)-nets and (t, s)-sequences, in *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing* (H. Niederreiter and P. J.-S. Shiue, eds), Springer, New York, pp. 299–317.

A. B. Owen (1997), 'Monte Carlo variance of scrambled net quadrature', *SIAM J. Numer. Anal.* **34**, 1884–1910.

A. B. Owen (1998), 'Scrambled net variance for integrals of smooth functions', *Ann. Statist.* In press.

W. H. Press and S. A. Teukolsky (1992), 'Portable random number generators', *Comput. Phys.* **6**, 522–524.

W. H. Press, S. A. Teukolsky, W. T. Vettering and B. P. Flannery (1992), *Numerical Recipes in C: The Art of Scientific Computing*, 2nd edn, Cambridge University Press.

D. I. Pullin (1979), 'Generation of normal variates with given sample and mean variance', *J. Statist. Comput. Simul.* **9**, 303–309.

I. M. Sobol' (1967), 'The distribution of points in a cube and the accurate evaluation of integrals', *Zh. Vychisl. Mat. Mat. Fiz.* **7**, 784–802. In Russian.

I. M. Sobol' (1976), 'Uniformly distributed sequences with additional uniformity property', *USSR Comput. Math. Math. Phys.* **16**, 1332–1337.

J. Spanier and E. H. Maize (1994), 'Quasi-random methods for estimating integrals using relatively small samples', *SIAM Rev.* **36**, 18–44.

W. Wagner (1992), 'A convergence proof for Bird's Direct Simulation Monte Carlo method for the Boltzmann equation', *J. Statist. Phys.* **66**, 1011–1044.

H. Woźniakowski (1991), 'Average case complexity of multivariate integration', *Bull. Amer. Math. Soc.* **24**, 185–194.

C. P. Xing and H. Niederreiter (1995), 'A construction of low-discrepancy sequences using global function fields', *Acta Arithmetica* **73**, 87–102.

S. K. Zaremba (1968), 'The mathematical basis of Monte Carlo and quasi-Monte Carlo methods', *SIAM Rev.* **10**, 303–314.