S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*

DOI: 10.1016/S0927-0507(06)13012-1

Chapter 12

Quasi-Random Number Techniques

C. Lemieux

Department of Mathematics and Statistics, University of Calgary, Canada
E-mail: lemieux@math.ucalgary.ca

Abstract

Over the last decade, quasi-Monte Carlo methods have been used as an efficient estimation tool in various high-dimensional applications, particularly in the field of finance. These methods can be seen as a deterministic version of the Monte Carlo method for multidimensional integration, in which quasi-random numbers are used to construct highly-uniform point sets over which the integrand is sampled. This chapter discusses the use of these techniques in simulation.

1 Introduction

At the heart of any simulation study, a reliable source of (pseudo)random numbers – such as those discussed in Chapter 3 – is required to ensure that the statistical output analysis of interest is done correctly. It may thus appear dangerous to replace this pseudorandom source by a stream of *quasi-random* numbers, which, unlike pseudorandom numbers, are designed to produce a highly-uniform sampling in which correlations are allowed. Nevertheless, over the last decade there have been many success stories in which using these quasi-random numbers have produced estimates with a smaller error than their pseudorandom counterpart (Paskov and Traub, 1995; Spanier, 1995; Ninomiya and Tezuka, 1996; Acworth et al., 1997; Caflisch et al., 1997; Morokoff and Caflisch, 1997). These successful applications of quasi-random numbers (particularly those in the field of finance) have received a great deal of attention (Business Week, 1994; The Economist, 1995; New York Times, 1995). But maybe more importantly, they have generated considerable interest among many researchers, thereby resulting in numerous papers where improvements and theories explaining this success have been presented (see, e.g., Caflisch et al., 1997; Morokoff and Caflisch, 1997; Acworth et al., 1997; Owen, 1998a; Sloan and Woźniakowski, 1998; Hickernell, 1998c; Hickernell and Wang, 2001; Owen, 2002; Papageorgiou, 2003; Sobol' and Asostsky, 2003; Wang and Fang,

2003). The purpose of this chapter is to present the general tools and principles required to use quasi-random numbers in simulation. In the remainder of this introduction, we briefly outline the main ideas behind these methods.

To better understand why it makes sense to use quasi-random numbers for simulation, it is useful to formulate the goal of the simulation study in terms of an integral to be estimated. Indeed, typically simulation is used to estimate one or more quantities of the form

$$\mu = \int_{[0,1)^s} f(\mathbf{u})\, \mathrm{d}\mathbf{u}, \tag{1}$$

where f is a real-valued function and s is a positive integer. The function f can be interpreted as the mapping that transforms a set of s numbers between 0 and 1 into an observation of the output quantity of interest, and μ is the expectation of this quantity. In other words, s is the number of pseudorandom numbers that are required in each run of the simulation, and $\mathbf{u}$ is the vector that contains those uniform numbers. If an unbounded number of uniform numbers are required in each simulation run, then s can be considered as infinite. An example describing what this function f is for a simple queueing problem is given in Section 2.

In this context, using n independent simulation runs to estimate μ amounts to using the Monte Carlo (MC) method. In this case, μ is approximated by

$$\hat{\mu}_{\mathrm{MC}} = \frac{1}{n} \sum_{i=1}^{n} f(\mathbf{u}_i), \tag{2}$$

where $P_n = \{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ is a set of n independent points uniformly distributed over $[0, 1)^s$. (Of course, in practice these n points are obtained by using a pseudorandom number generator.)

From this point of view, it seems like better estimates of μ could be obtained by choosing these n sampling points more carefully. For example, it is well known that for small dimensions s (say, below 4 or 5), methods like Gaussian quadrature or Simpson's rule can achieve a smaller error than MC (Davis and Rabinowitz, 1984). The use of quasi-random numbers, in what has become known as *quasi-Monte Carlo (QMC) methods*, can also be shown to produce a smaller error than MC, at least asymptotically and for certain classes of functions (see, e.g., Niederreiter, 1992). These methods provide estimators of the same form as the MC estimator (2), but in which the random point set P_n is replaced by a deterministic, highly-uniform point set. They were proposed as an alternative to the MC method shortly after the latter was introduced at the end of the 1940's. One of these early proposals was from Richtmyer (1951), who suggested approximating multidimensional integrals by using the points produced by the sequence $\{n\boldsymbol{\alpha} \bmod 1, n = 1, 2, \ldots\}$, where $\boldsymbol{\alpha}$ is an s-dimensional vector of irrational numbers which, together with 1, are linearly independent over the rationals, and the mod 1 operation is taken coordinate-wise. Other constructions that produce highly-uniform point

sets were proposed later, including Korobov rules (Korobov, 1959), Halton sequences (Halton, 1960), Sobol' sequences (Sobol', 1967), Faure sequences (Faure, 1982), Niederreiter sequences (Niederreiter, 1987) and Niederreiter–Xing sequences (Niederreiter and Xing, 1996).

The uniformity of the point sets used by QMC methods is often measured by their "discrepancy", that is, by how far their induced empirical distribution departs from a truly uniform distribution. For this reason, another way to refer to quasi-random numbers is to talk about *low-discrepancy point sets or sequences*. In this chapter, we interchangeably use the terms low-discrepancy point sets, highly-uniform point sets, or quasi-random point sets. Typically, a sequence of numbers $\mathbf{u}_1, \mathbf{u}_2, \ldots$ in $[0,1)^s$ is considered to have low discrepancy if $D(P_n) \in \mathrm{O}(\log^s n/n)$, where $P_n = \{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ contains the first n points of the sequence, and $D(P_n)$ is the so-called *star discrepancy* of P_n, which is defined by

$$D(P_n) = \sup_{\mathbf{v}=(v_1,\ldots,v_s)\in[0,1)^s} \left| \prod_{j=1}^{s} v_j - \frac{1}{n} \left| P_n \cap \prod_{j=1}^{s} [0, v_j) \right| \right|.$$

That is, consider all rectangular boxes anchored at the origin (as determined by $\mathbf{v}$) and compute the difference between the volume of the box and the fraction of points in P_n that fall in the box. Then take the supremum of this difference over all boxes. For example, in one dimension the regular grid $\{0, 1/n, 2/n, \ldots, (n-1)/n\}$ has a discrepancy of $1/n$. For a dimension s larger than one however, the s-dimensional grid $\{(i_1/N, \ldots, i_s/N), 0 \leqslant i_j < N, 1 \leqslant j \leqslant s\}$ with $N = n^{1/s}$ has a discrepancy in $\mathrm{O}(1/N) = \mathrm{O}(n^{-1/s})$, so it is not a low-discrepancy point set. When $s > 1$, the points must be placed more carefully to get a low discrepancy, i.e., not by simply taking the Cartesian product of s one-dimensional low-discrepancy point sets. For example, one kind of construction called a *lattice rule* places the points on the vertices of a *lattice* that intersect the unit hypercube $[0,1)^s$. Figure 1 illustrates the difference between a lattice point set and a regular grid in two dimensions.

There is a large amount of literature on the concept of discrepancy (Niederreiter, 1992; Morokoff and Caflisch, 1994; Hickernell, 1998a; Matoušek, 1998), but we do not discuss it further here since its relevance for simulation is rather limited.

Because of the deterministic nature of the point sets P_n on which QMC methods are based, the error analysis for these methods is often done by trying to derive upper bounds on the deterministic error

$$E_n = |\hat{\mu}_{\mathrm{QMC}} - \mu|,$$

where

$$\hat{\mu}_{\mathrm{QMC}} = \frac{1}{n} \sum_{\mathbf{u}_i \in P_n} f(\mathbf{u}_i),$$

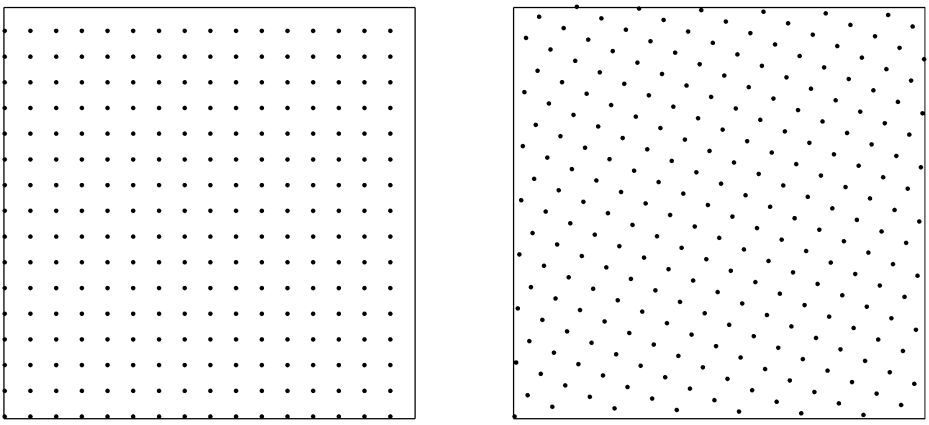

Fig. 1. *Left*: two-dimensional regular grid. *Right*: two-dimensional lattice point set.

and P_n is a highly-uniform point set. For example, one can use the *Koksma–Hlawka inequality*

$$E_n \leqslant D(P_n)V(f), \tag{3}$$

where $D(P_n)$ is the star discrepancy and $V(f)$ is the variation of f in the sense of Hardy and Krause (see Niederreiter, 1992, for more details). This inequality can be used to show that for any function with $V(f) < \infty$, the error of the approximation $\hat{\mu}_{\mathrm{QMC}}$ is in $\mathrm{O}(\log^s n/n)$, which for a fixed s is better than the probabilistic $\mathrm{O}(n^{-1/2})$ associated with MC.

What this means is that asymptotically and for a fairly limited class of functions, the error is smaller for approximations based on low-discrepancy point sets than it is for MC-based estimates. However, one of the problems with this type of analysis is that it does not explain the success of QMC methods on high-dimensional integrands (e.g., $s = 360$ in Paskov and Traub, 1995) since for s even as small as 10, n needs to be about 10^{39} in order to have $\log^s n/n \leqslant n^{-1/2}$. Also, inequalities such as (3) only provide bounds on the error and cannot be used to estimate E_n.

If we go back to the idea of using quasi-random numbers for simulation, it seems like it cannot work since in the simulation context, estimates of the error are important, unbiased estimators are typically preferred, and we often have to work with large values of s. Fortunately, these seemingly major hurdles can be removed if instead of using purely deterministic quasi-random numbers, one uses *randomized quasi-Monte Carlo* (RQMC) methods. The idea here is to randomize a highly-uniform point set P_n so that each of its points follows the uniform distribution over $[0, 1)^s$, while preserving the high uniformity of P_n. Many randomization techniques that achieve this have been proposed and used in practice; see Owen (1998a), L'Ecuyer and Lemieux (2002) and the references therein. Since they are better suited for simulation, we will be mostly discussing RQMC methods in this chapter.

Once we have a randomized highly-uniform point set $\widetilde{P}_n$ with the properties mentioned above, each of its points are used to drive one run of the simulation. More precisely, the ith run is driven by the s coordinates of $\mathbf{u}_i = (u_{i,1}, \ldots, u_{i,s})$. Since each $\mathbf{u}_i$ is uniformly distributed over $[0,1)^s$, the estimator

$$\hat{\mu}_{\mathrm{RQMC}} = \frac{1}{n} \sum_{\mathbf{u}_i \in \widetilde{P}_n} f(\mathbf{u}_i) \tag{4}$$

is unbiased, and its variance can be estimated by creating m copies of $\hat{\mu}_{\mathrm{RQMC}}$ based on m i.i.d. copies of the randomized point set $\widetilde{P}_n$. More details on this procedure are given in Sections 2 and 6.

As we just outlined, with RQMC methods it is possible to estimate the variance of the estimator $\hat{\mu}_{\mathrm{RQMC}}$. One can therefore verify empirically whether the RQMC estimator has a smaller variance than the corresponding MC estimator based on n points. This type of analysis has been done for different kinds of problems, using different quasi-random point sets and randomizations (see, e.g., Tan and Boyle, 2000; L'Ecuyer and Lemieux, 2000; Kollig and Keller, 2002; Lemieux and L'Ecuyer, 2003). In most (if not all) cases, it was observed that the RQMC estimator reduced the variance compared to the MC estimator. Moreover, it can be proved theoretically that in some cases, the variance of the RQMC estimator is smaller than the MC variance (Owen, 1997a, 1997b, 1998b). For this reason, one can think of RQMC methods as *variance reduction techniques* such as those discussed in Chapters 10 and 11. More precisely, because of the nature of the randomized point set $\widetilde{P}_n$, they can be seen as *correlation induction* methods.

The plan for the remainder of this chapter is as follows. We illustrate the main ideas of quasi-random number techniques in Section 2 by using a simple queueing example. In Section 3, we discuss the concept of effective dimension, which is important to understand the success of quasi-random number techniques in practice. The main tools used for designing highly-uniform point sets are presented in Section 4, where we also describe some of the constructions most commonly used in practice. Section 5 is devoted to recurrence-based point sets, which have been discussed in Chapter 3 and provide an especially useful type of highly-uniform point set in practice. In Section 6, we talk about randomization techniques and give some theoretical results supporting the use of RQMC methods as variance reduction techniques. The connections between these results and selection criteria for choosing good parameters for different constructions are also briefly mentioned. Issues that arise when quasi-random point sets are combined with other variance reduction techniques are discussed in Section 7. Finally, current and future avenues of investigation for the use of quasi-random point sets in simulation are presented in Section 8.

2 An example

Consider an $M/M/1$ queue with an arrival rate of $\lambda = 1$ customer per minute and a service rate of $\mu = 1.2$ customers per minute. Suppose the system starts empty, runs for 8 hours, and that we want to estimate the expected average waiting time in the queue for the customers that entered the system during that period. Formally, our goal is to estimate

$$\mu = \mathrm{E}\left(\frac{\sum_{i=1}^{N} w_i}{N}\right),$$

where N is the number of customers that have entered the system in 8 hours and w_i is the waiting time in the queue of the ith customer. The above ratio is defined to be 0 when $N = 0$. We have expressly chosen a finite-horizon measure of performance for this example, just so that the problem would fit more naturally in the general framework outlined in the Introduction.

Using the formulation (1), we see that here the dimension s is unbounded since it depends on the number N of customers entering the system, which is itself unbounded. To describe the function f in (1) that corresponds to this problem, we first use Lindley's equation

$$w_i = \max(0, w_{i-1} + s_{i-1} - a_i), \quad i \geqslant 1, \tag{5}$$

where s_i is the service time of the ith customer, a_i is the interarrival time between the $(i-1)$st and ith customer, and $w_0 = s_0 = 0$.

To write w_i as a function of $\mathbf{u} = (u_1, u_2, \ldots)$, we need to decide what nonuniform generation method (see Chapter 4) to use for the a_i's and the s_i's. With RQMC methods, *inversion* is the most natural choice because it often helps minimizing the number s of uniform numbers required for each simulation. Thus we let

$$a_i = -\frac{\ln(1-u_{2i-1})}{\lambda} \tag{6}$$

and

$$s_i = -\frac{\ln(1-u_{2i})}{\mu}. \tag{7}$$

Using (5) along with (6) and (7), it is easy to see that each w_i can be written as a function g_i of $u_1, \ldots, u_{2i-1}$. Similarly, the number N of customers entering the system can be written as

$$N = \sum_{i=1}^{\infty} \mathbf{I}_{a_1+\cdots+a_i<480},$$

where $\mathbf{I}_X$ is the indicator function for the event X. Since a_i is a function of u_{2i-1} for each i, N itself becomes a function of $\mathbf{u}$. We can then write

$$f(\mathbf{u}) = \frac{1}{N(\mathbf{u})} \sum_{i=1}^{N(\mathbf{u})} g_i(u_1, \ldots, u_{2i-1}).$$

A possibly more intuitive way of understanding how f takes the vector $\mathbf{u}$ as input and outputs an observation of the quantity of interest is to use pseudocode as follows:

OneSim$(u_1, u_2, \ldots)$
  TotWait = 0 // Total waiting time
  $w = 0$
  $a = -\ln(1 - u_1)/\lambda$
  $j = 2$ // Indexes the coordinate of $\mathbf{u}$ to be used next
  time = a // Current time
  NbCust = 1 // Number N of customers that entered so far
  while(time < 480) **do**
    $s = -\ln(1 - u_j)/\mu$
    $a = -\ln(1 - u_{j+1})/\lambda$
    NbCust = NbCust + 1
    time = time + a
    $w = \max(0, w + s - a)$
    if(time < 480) **then** TotWait = TotWait + w
    $j = j + 2$
  return(TotWait/NbCust)

This describes how this simple simulation problem fits the framework of multidimensional integration over $[0, 1)^s$, with $s = \infty$ here. Now we need to explain how quasi-random numbers can be used to estimate μ. Probably the simplest method to describe is the one based on a *randomly shifted Korobov point set*. This is a special case of a randomized recurrence-based point set where the underlying highly-uniform point set is obtained by taking the successive overlapping vectors produced by a linear congruential generator (LCG). The point set is then randomized by adding a random vector $\mathbf{v} = (v_1, v_2, \ldots)$ modulo 1 to each point.

In practice, using the LCG formulation to construct P_n is very useful (see Section 5). Here however, we describe P_n using the "Korobov formulation" (Korobov, 1959). The ith point in the randomized point set $\widetilde{P}_n$ is given by

$$\mathbf{u}_i = \left(\frac{i(1, a, a^2, \ldots) \bmod n}{n} + \mathbf{v} \right) \bmod 1,$$

where $a \in \{1, \ldots, n-1\}$ is the generator of the point set. Hence $u_{i,j} = ((i \times a^{j-1} \bmod n)/n + v_j) \bmod 1$ for each $i = 0, \ldots, n-1$, $j \geqslant 1$.

Putting everything together, the following algorithm estimates μ by using 10 i.i.d. copies of an RQMC estimator based on a randomly shifted Korobov point set. It returns an estimator for μ and its estimated standard deviation. We assume below that Rand01$(\cdot)$ returns a (pseudo)random uniform number between 0 and 1, and that ave$(\mathbf{x})$ and std$(\mathbf{x})$ return the average and sample standard deviation of the vector $\mathbf{x}$, respectively. In the pseudocode below, the definition of $\mathbf{v}$ suggests it is a vector of infinite length. In practice, it suffices to generate the coordinates only as they are needed. More details are given in Section 6.

QueueKorobov
  for $l = 1$ **to** 10 **do**
    sum $= 0$
    $\mathbf{v} = (v_1, v_2, \ldots)$ // where $v_j =$ Rand01$(\cdot)$
    for $i = 0$ **to** $n - 1$ **do**
      sum $=$ sum $+$ **OneSim**$(u_{i,1}, u_{i,2}, \ldots)$
        // where $u_{i,j} = ((1/n)(i \times a^{j-1} \bmod n) + v_j) \bmod 1$
    end for
    $x[l] =$ sum $/ n$
  end for
  return(ave$(\mathbf{x})$, std$(\mathbf{x})$) // $\mathbf{x} = (x[1], x[2], \ldots, x[10])$

In Table 1, we give the results obtained both with a randomly shifted Korobov point set and the MC method for this example. The generators used are $a = 76$ and $a = 1516$ for $n = 1021$ and $n = 4093$, respectively (L'Ecuyer and Lemieux, 2000). The first number in each entry for "Korobov" (MC) is the mean of 10 i.i.d. copies of $\hat{\mu}_{\mathrm{RQMC}}(\hat{\mu}_{\mathrm{MC}})$ based on n points, and the number in parentheses is the estimated standard deviation of $\hat{\mu}_{\mathrm{RQMC}}(\hat{\mu}_{\mathrm{MC}})$. As we can see there, the RQMC estimator has an estimated standard deviation less than half the size of its MC counterpart. In addition, the computation time required is slightly smaller for this RQMC estimator than for the MC estimator, so in terms of efficiency, the gains are even higher. Note that for this example, since the average number of clients entering the system in 8 hours is 480 and we need to generate two random variables per client, the average number of required components in $\mathbf{u}$ is 960.

Table 1.
Simulation results with a randomly shifted Korobov point set and MC for the $M/M/1$ problem: each entry gives the estimated mean (standard deviation)

n	Korobov	MC
1021	3.84 (0.04)	3.87 (0.10)
4093	3.85 (0.02)	3.85 (0.04)

3 A key concept: Effective dimension

We saw in the previous section an example where the RQMC estimator has an empirical standard deviation less than half the size of its MC counterpart on a problem where the dimension s is close to 1000. The "classical" analysis that we briefly outlined in the Introduction (e.g., the Koksma–Hlawka inequality (3)) cannot explain this kind of result. Moreover, when the dimension s is close to 1000, it seems impossible that a set of only 1021 points could have sampled the unit hypercube $[0, 1)^s$ very evenly. To explain the success of the RQMC estimator in this example, we must look at the function f that is integrated, and how it interacts with the point set $\widetilde{P}_n$ used. (Since the features of $\widetilde{P}_n$ that are relevant when studying this interaction are usually inherited from P_n, we consider the unrandomized point set P_n in what follows.)

Intuitively, it makes sense to think that in this example, the waiting time of a customer will probably not be affected by the waiting times of customers that were not close to him in the system, e.g., that arrived one hour before or after him. In terms of the function f, this translates into saying that the interaction of, say, two coordinates u_j and u_l where $|j - l|$ is large does not contribute very much to the variability of f. Going one step further, one could say that for this example, it is reasonable to assume that f can be well approximated by a sum of functions defined over subspaces of low dimensions. Consequently, as long as the set P_n is highly-uniform over these subspaces, the RQMC estimator based on P_n will be a good approximation for f. These ideas can be made more formal by using the *functional ANOVA decomposition* of a function (Hoeffding, 1948), which we now describe.

3.1 A functional ANOVA decomposition

This decomposition allows us to write any square-integrable function f defined over $[0, 1)^s$ as a sum

$$f(\mathbf{u}) = \sum_{I \subseteq \{1,\ldots,s\}} f_I(\mathbf{u}),$$

where f_I only depends on those variables u_j such that $j \in I$. This decomposition is such that $\int_{[0,1)^s} f_I(\mathbf{u})\,\mathrm{d}\mathbf{u} = 0$ for any nonempty I, and $f_\emptyset(\mathbf{u}) = \mu$. Also, it is orthogonal, i.e., for any $I \neq J$, we have that

$$\int_{[0,1)^s} f_I(\mathbf{u}) f_J(\mathbf{u})\,\mathrm{d}\mathbf{u} = 0.$$

We assume s is finite here, but ANOVA decompositions can also be used to analyze integrands f of infinite dimension (Owen, 1998a).

Each component is defined as follows (Owen, 1998a): $f_\emptyset(\mathbf{u}) = \mu$, and for $I \neq \emptyset$, $f_I(\mathbf{u}) = \int_{[0,1)^{s-|I|}} f(\mathbf{u})\,\mathrm{d}\mathbf{u}_{I^c} - \sum_{J \subset I} f_J(\mathbf{u})$, where I^c is the complement of I and $\subset$ denotes strict set inclusion. For example, if $f(u_1, u_2) = u_1 + 2u_1u_2^2 +$

u_2^3, then $f_\emptyset(u_1, u_2) = 13/12$ $f_{\{1\}}(u_1, u_2) = 5u_1/3 - 5/6$, $f_{\{2\}}(u_1, u_2) = u_2^2 + u_2^3 - 7/12$, and $f_{\{1,2\}}(u_1, u_2) = 2u_1u_2^2 - 2u_1/3 - u_2^2 + 1/3$.

For each nonempty subset I, the variance σ_I^2 of the ANOVA component f_I is given by

$$\sigma_I^2 = \mathrm{Var}\big(f_I(\mathbf{U})\big) = \int_{[0,1)^s} f_I^2(\mathbf{u})\,\mathrm{d}\mathbf{u}.$$

Because of the orthogonality of the f_I's, we have that $\sum_{I\subseteq\{1,\ldots,s\}} \sigma_I^2 = \sigma^2$, where $\sigma^2 = \mathrm{Var}(f(\mathbf{U}))$. Therefore we can view σ_I^2/σ^2 – these are called *sensitivity indices* in Sobol' (2001) – as a measure of the relative importance of the ANOVA component f_I. Note that the best mean-square approximation of f by a sum of d-dimensional (or less) functions is given by $\sum_{I:|I|\leqslant d} f_I$.

In many situations – for example, when evaluating financial products – the function f of interest is such that the components f_I with $|I|$ small are the most important. For instance, Lemieux and Owen (2001) estimate that the one- and two-dimensional components of the 30-dimensional function representing the price f of an Asian option (with specific parameters) contribute about 97% of the variance of f. In some sense, this suggests that for this example, the *effective dimension* of f is 2 rather than 30. In Paskov (1997), this concept of effective dimension was used to explain the success of QMC methods on a 360-dimensional integrand resulting from the evaluation of a Collateralized Mortgage Obligation (CMO). Other investigations of the effective dimension in finance problems are reported in, e.g., Wang and Fang (2003), Wang and Sloan (2003). Because of its crucial importance in our understanding of the success of QMC methods in practice, we now discuss this concept in more detail.

3.2 Effective dimension

Our treatment here closely follows Caflisch et al. (1997) and Hickernell (1998b), which provide two definitions of the effective dimension of f based on its ANOVA decomposition.

Definition 1. The effective dimension of f (in proportion p), in the superposition sense, is the smallest integer d_S such that $\sum_{I:|I|\leqslant d_S} \sigma_I^2 \geqslant p\sigma^2$.

In the 30-dimensional Asian option mentioned above, f has an effective dimension of 2 in the superposition sense, in proportion 0.97.

Definition 2. The effective dimension of f (in proportion p), in the truncation sense, is the smallest integer d_T such that $\sum_{I:I\subseteq\{1,\ldots,d_T\}} \sigma_I^2 \geqslant p\sigma^2$.

When a function f has an effective dimension of d in the superposition sense in proportion p, it means that it can be approximated by a sum $\tilde{f}$ of

d-dimensional (or less) functions, and that the approximation explains at least $100p\%$ of the variance. Now, if we have a point set P_n such that all projections $P_n(I)$ of the form

$$P_n(I) = \{(u_{i,i_1}, \ldots, u_{i,i_j})\colon \mathbf{u}_i = (u_{i,1}, \ldots, u_{i,s}) \in P_n, I = \{i_1, \ldots, i_j\}\}$$

with $|I| = j \leqslant d$ are highly-uniform, then the estimator $\hat{\mu}_{\mathrm{RQMC}}$ based on P_n should approximate the integral of $\tilde{f}$ very well. Furthermore, if p is close to 1 then $\tilde{f}$ is "close" to f (in a mean-square sense), and thus $\hat{\mu}_{\mathrm{RQMC}}$ should also be a very good estimator of μ. A more formal treatment of this kind of argument can be found in Wang and Fang (2003).

The definition of the effective dimension in the truncation sense can lead to similar arguments describing how the interaction between P_n and f can lead to a successful application of RQMC methods. An important difference with the definition in the superposition sense is that the ordering of the variables $u_1, \ldots, u_s$ matters here. The motivation behind this definition is that for some QMC constructions such as Sobol' and Halton sequences, it has been observed that the projections $P_n(I)$ deteriorate as the indices in I increase (see, e.g., Morokoff and Caflisch, 1994). In order for the estimator $\hat{\mu}_{\mathrm{RQMC}}$ to be good for such point sets, the function f to be integrated must be such that the components f_I with $|I|$ or $i_{\min I} = \min\{j\colon j \in I\}$ large should be unimportant. Not all constructions for highly-uniform point sets have this undesirable feature. For instance, recurrence-based point sets have projections $P_n(I)$ that do not deteriorate as $i_{\min I}$ increases (all things being equal elsewhere) because they are *dimension-stationary*. This means that the projections $P_n(I)$ depend only on the spacing between the indices in I, e.g., $P_n(\{1, 3, 4\}) = P_n(\{2, 4, 5\}) = P_n(\{10, 12, 13\})$.

Another possible definition for the effective dimension is given in L'Ecuyer and Lemieux (2000).

Definition 3. The effective dimension of f (in proportion p), in the successive-dimensions sense, is the smallest integer d_{SD} such that

$$\sum_{i=1}^{s-d_{\mathrm{SD}}+1} \sum_{I:I\subseteq\{i,\ldots,i+d_{\mathrm{SD}}-1\}} \sigma_I^2 \geqslant p\sigma^2.$$

As for d_T, the value of d_{SD} depends on the ordering of the variables u_j because of the restriction on the subsets I that the range $r_I = \max\{j\colon j \in I\} - \min\{j\colon j \in I\}$ be upper bounded by d_{SD}. The motivation behind this definition is that in some problems, especially in the simulation context, a variable u_j does not interact very strongly with variables u_l such that $|l - j|$ is large. We made this point (intuitively) at the beginning of this section, when discussing the $M/M/1$ example. In addition, when designing highly-uniform point sets in high dimensions, it is not possible in practice to make sure that all projections

$P_n(I)$ are good, even if we only consider those with $|I| \leqslant d$ for some $d < s$. However, if the point set is designed for classes of functions that are believed to have a small effective dimension in the above sense, then it is reasonable to decrease the number of projections considered by adding the corresponding restriction on the range of I. We refer the reader to Chapter 3 for more on selection criteria that use such restrictions.

3.3 Relevance in the simulation context

In the context of simulation, it often happens that the integrand f in (1) has a very large nominal dimension, but a small or moderate effective dimension. Intuitively, the reason for this is that although a large number of input variables are used in each simulation, each of them usually interacts strongly with only a few other variables. For this reason, RQMC methods can often be used successfully in the simulation context. For example, in Lemieux and L'Ecuyer (1999), RQMC methods are used to reduce the variance by factors between 2 and 5 for a ruin probability problem where the nominal dimension is 8000.

It is important to note that in the formulation (1), there are two things that can influence the size of the effective dimension of f: (i) which nonuniform generation methods are used; (ii) how the variables u_j are assigned to the input variables in the problem. As we mentioned in Section 2, with RQMC methods it is preferable to use inversion for generating nonuniform random variables, as it often helps minimizing the dimension (nominal, and thus effective) of f. Inversion also helps with the assignment mentioned in (ii), in the same way it helps achieve synchronization when common random numbers or antithetic variates are used (Law and Kelton, 2000, pp. 586–599). When using inversion, there is often a "natural" way of assigning the coordinates u_j to the input variables: one simply assigns the u_j's in the chronological order produced by the simulation. This is what we did in Section 2: u_1 was used to generate the first interarrival time, then u_2 was used for the service time of the first client, then u_3 for the next interarrival time, etc. Assignment done in this way typically contributes to reduce the effective dimension (in the successive-dimensions sense) since the u_j's whose indices are far apart are associated with events that are far apart in time and thus do not interact strongly together. This is especially true when the simulation has regenerative cycles (see Chapter 16). In this case, if all the variables u_j associated with one cycle have successive indices, then we can say that roughly, the effective dimension (in the successive-dimensions sense) is bounded above by the expected number of input variables associated with one regenerative cycle.

The assignment choice may not be that crucial in some cases and, as mentioned before, it does not affect the value of the effective dimension in the superposition sense. Also, if the point set P_n is such that for any fixed size d, each projection $P_n(I)$ with $|I| = d$ is of the same quality, then the assignment can be done arbitrarily. This is true when P_n is a set of i.i.d. uniform points as in the MC method, but typical highly-uniform point sets do not have this

property. Instead, for a fixed size $|I|$, the projections $P_n(I)$ with either a small minimal index $i_{\min I}$ or range r_I are usually more uniform. In this case, the assignment choice may make a difference, but what may really help with such point sets is to use *dimension reduction techniques*, which we now discuss.

3.4 Reducing the effective dimension

For some problems, it is possible to use techniques that can reduce the effective dimension. One kind of application where this is especially true is when the problem requires the simulation of an underlying Brownian motion. In this case, one can use the *Brownian Bridge Technique* (Caflisch et al., 1995; Morokoff and Caflisch, 1997), which uses the Brownian bridge property to generate the Brownian motion's steps in an arbitrary order.

Let $0 \leqslant t_1 < \cdots < t_s$. For a standard Brownian motion $B(\cdot)$, the usual way of generating a path $B(t_1), \ldots, B(t_s)$ from a uniform point $\mathbf{u} = (u_1, \ldots, u_s)$ is as follows:

$t_0 = B(t_0) = 0$
for $j = 1$ **to** s
  $B(t_j) = B(t_{j-1}) + \sqrt{t_j - t_{j-1}}\Phi^{-1}(u_j)$ // where $\Phi(x) = \mathrm{P}(\mathrm{N}(0,1) \leqslant x)$
end for

An alternative idea is to try using the first few coordinates of $\mathbf{u}$ to specify as much as possible the behavior of $B(\cdot)$, so that functions of $B(\cdot)$ will (hopefully) have a small effective dimension in the truncation sense. The Brownian bridge technique does that by first generating $B(t_s)$, then $B(t_{\lfloor s/2 \rfloor})$, then $B(t_{\lfloor s/4 \rfloor})$ and $B(t_{\lfloor 3s/4 \rfloor})$, and so on. This can be done easily since the Brownian bridge property tells us that for any $u < v < w$, we have that $B(v) \mid (B(u) = a, B(w) = b)$ has a normal distribution with mean $a(w-v)/(w-u) + b(v-u)/(w-u)$ and variance $(v-u)(w-v)/(w-u)$. Similar techniques can be used to generate Poisson processes, as discussed in Fox (1999).

For Brownian motion, the above technique can be generalized by observing that the standard method to generate $B(\cdot)$ can be written as

$$\begin{pmatrix} B(t_1) \\ \vdots \\ B(t_s) \end{pmatrix} = \mathbf{A} \begin{pmatrix} z_1 \\ \vdots \\ z_s \end{pmatrix},$$

where

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & 1 & 0 & \ldots & 0 \\ & & \vdots & & \\ 1 & 1 & 1 & \ldots & 1 \end{pmatrix},$$

if we assume that $t_j - t_{j-1} = 1$ for each $j = 1, \ldots, s$, and the $z_j = \Phi^{-1}(u_j)$ are i.i.d. standard normal variables. Replacing $\mathbf{A}$ by a matrix $\mathbf{B}$ such that $\mathbf{B}\mathbf{B}^{\mathrm{T}} =$

$\mathbf{A}\mathbf{A}^{\mathrm{T}} =: \boldsymbol{\Sigma}$ is called a *generalized Brownian bridge technique* in (Morokoff and Caflisch, 1997). For example, Acworth et al. (1997) use a principal components analysis to define $\mathbf{B}$, i.e., they take $\mathbf{B} = \mathbf{P}\mathbf{D}^{1/2}$, where $\mathbf{P}$'s columns are formed by the eigenvectors of the covariance matrix $\boldsymbol{\Sigma}$, and $\mathbf{D}$ is a diagonal matrix containing the corresponding eigenvalues of $\boldsymbol{\Sigma}$ in decreasing order. This method was shown to numerically outperform the Brownian bridge technique in Acworth et al. (1997), but its computation time is much longer since to simulate n Brownian motion paths, it runs in $\mathrm{O}(ns^2)$ rather than the $\mathrm{O}(ns)$ required for the standard and Brownian bridge methods. Following this work, Åkesson and Lehoczy (2000) proposed a modification reducing the computation time for the principal components' method.

It is important to be aware that these dimension reduction methods do not provide estimators with reduced variance for all problems. They aim at improving the simulation of the Brownian motion paths, but they do not take into account how these paths contribute to the value of the function to be estimated. For example, Papageorgiou (2002) provides numerical results showing that for a certain type of digital option in finance, the Brownian bridge technique produces estimators with a far worse error than the standard method does.

For simulation problems that do not rely on Brownian motions or Poisson processes, it is still possible to come up with dimension reduction techniques. For example, using *Conditional Monte Carlo* (CMC) typically amounts to reducing the number of input variables that need to be generated, thereby resulting in an automatic reduction of the (nominal) dimension. See for example L'Ecuyer and Lemieux (2000), where CMC and RQMC are used to simulate a stochastic activity network.

4 Constructing quasi-random point sets

As seen in the previous sections, the success of RQMC methods in practice relies on the availability of highly-uniform point sets that interact in a constructive way with the integrand. Of course, the applicability of these methods would be greatly diminished if one had to choose a specific point set for each problem in order for this interaction to work properly. A more reasonable approach is to have a few good constructions whose parameters have been chosen so that for a large number of integrands, they should produce good results, that is, estimators with lower variance than MC.

Currently, most of the highly-uniform point sets used in practice come from two families of constructions. *Lattice Rules* (Niederreiter, 1992; Sloan and Joe, 1994), which generalize the early construction of Korobov (1959), and *digital nets and sequences* (Niederreiter, 1992; Tezuka, 1995), which generalize the Sobol' sequence. Further, we describe these families and some of their most widely used constructions. We do not discuss polynomial integration lattices

and their extensions, and refer the reader to Lemieux and L'Ecuyer (2003), Niederreiter (2003) and L'Ecuyer (2004) for information on those.

4.1 Lattice rules

Lattice rules construct approximations for (1) by using a *lattice point set* of the form $P_n = L \cap [0,1)^s$, with L an *integration lattice* defined as

$$L = \left\{ \mathbf{x} = \sum_{j=1}^{s} z_j \mathbf{v}_j \colon \mathbf{z} = (z_1, \ldots, z_s) \in \mathbb{Z}^s \right\}, \tag{8}$$

where the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_s \in \mathbb{R}^s$ are linearly independent and form a *basis* for L. In other words, a lattice is obtained by taking all integer linear combinations of the vectors in its basis. In addition, L must contain $\mathbb{Z}^s$ – this is what makes it an *integration* lattice – which implies that each $\mathbf{v}_j$ contains only rational numbers. As an example, a Korobov rule is based on a lattice with a basis of the form $\mathbf{v}_1 = (1, a, a^2, \ldots, a^{s-1})/n \bmod 1$, $\mathbf{v}_j = \mathbf{e}_j$ for $j = 2, \ldots, s$, where $\mathbf{e}_j$ is a vector of zeros with a one in the jth position.

A given lattice does not have a unique basis. However, its *rank* r and *invariants* $n_1, \ldots, n_r$ are uniquely determined integers having the property that (i) P_n can be written as

$$P_n = \left\{ \frac{j_1}{n_1}\mathbf{z}_1 + \cdots + \frac{j_r}{n_r}\mathbf{z}_r \colon 0 \leqslant j_l < n_l, l = 1, \ldots, r \right\}, \tag{9}$$

where $\mathbf{z}_1, \ldots, \mathbf{z}_r$ are linearly independent integer vectors; (ii) n_{j+1} divides n_j, for $j = 1, \ldots, r-1$ and $n = n_1, \ldots, n_r$ (Sloan and Joe, 1994). The formulation (9) is easier to work with than (8) is when the time comes to choose specific lattice constructions. More precisely, one can choose the rank and invariants beforehand, and then proceed to a search for "good" vectors $\mathbf{z}_1, \ldots, \mathbf{z}_r$ according to some criterion. When the rank is larger than one, these vectors are not uniquely determined in (9), and so computer searches must be done carefully in order to avoid too much duplication. See Sloan and Walsh (1990) for examples of computer searches where the rank is 2.

In practice, most lattice rules are based on lattices of rank 1. In this case, the lattice point set can be written as

$$P_n = \left\{ \frac{i}{n}(z_1, \ldots, z_s) \bmod 1 \colon 0 \leqslant i < n \right\},$$

where $(z_1, \ldots, z_s) \in \mathbb{Z}^s$ is the *generating vector*. The parameters to be determined are thus $z_1, \ldots, z_s$, and there is no loss of generality in assuming that $z_1 = 1$ and $1 \leqslant z_j < n$ for $j = 2, \ldots, s$. Restricting the search to rank-1 rules certainly simplifies the search procedure. In addition, rank-1 rules are *fully projection regular*, a property meaning that each projection $P_n(I)$ contains n distinct points. Lattices of higher rank do not provide point sets with this desirable property.

Korobov rules are an example of rank-1 rules. As seen in Section 2, only one parameter, the generator a of P_n, needs to be specified. This provides additional speedup for the search procedure. Moreover, Korobov point sets are dimension-stationary. Tables of good generators a for various values of n can be found in, e.g., Haber (1983), L'Ecuyer and Lemieux (2000).

In our discussion of lattice rules, we have assumed that the number of points n was fixed. In applications where a certain level of accuracy is desired, the required number of evaluation points may not be known in advance. In this case, constructions whose number of points can be increased indefinitely – just like with digital sequences – are better suited. Recently, lattice rules whose size n can be extended have been studied (Hickernell and Hong, 1997; Hickernell et al., 2001; Hickernell and Niederreiter, 2003) and parameters for such sequences can be found in Hickernell et al. (2001). Another recent advance in the theory of lattice rules is the development of component-by-component constructions for rank-1 rules that minimize a certain criterion (more precisely, a worst-case error in some class of functions). Tables of parameters for such point sets can be found in, e.g., Sloan et al. (2002), Sloan and Rezstov (2002). The constructions studied in Sloan et al. (2002) have the additional advantage of achieving a *strong tractability error bound* in some space of functions. This roughly means that the number of points required to keep the error bounded as the dimension s increases does not grow exponentially with s.

4.2 Digital nets and sequences

Before giving a general description of digital nets and sequences, let us first present a one-dimensional low-discrepancy sequence due to van der Corput (1935). This sequence makes use of the *radical-inverse function* φ_b, a central tool in the design of digital nets. Let n be a nonnegative integer and consider its unique digital expansion in base b given by

$$n = \sum_{i=0}^{\infty} a_i b^i, \tag{10}$$

where $0 \leqslant a_i < b$, and $a_i = 0$ for all sufficiently large i, i.e., the sum in (10) is actually finite. Then we have

$$\varphi_b(n) = \sum_{i=0}^{\infty} a_i b^{-i-1}.$$

The van der Corput sequence in base 2 $u_0, u_1, \ldots$ is defined by

$$u_n = \varphi_2(n), \quad n \geqslant 0,$$

i.e., the nth term in the sequence is obtained by applying the radical inverse function in base 2 to n.

To generalize this construction to point sets or sequences of higher dimensions, at least two ideas can be exploited: (i) for each dimension $j = 1, \ldots, s$, apply a different linear transformation to the coefficients a_i before applying φ_b; (ii) use a different base b for each dimension. The Sobol' sequence is based on the first idea, while the Halton sequence uses the second one. More generally, applying the first idea in an arbitrary base b is what gives rise to the general notion of digital net and sequence, which we now describe following Niederreiter (1992). (Our description is not as general as the one given there, but general enough to cover most constructions used in practice.)

Let b be a prime and $k \geqslant 0$ be an integer. A *digital net in base b* with $n = b^k$ points is a point set P_n defined by *s generating matrices* $\mathbf{C}_1, \ldots, \mathbf{C}_s$ of size $k \times k$, with entries in $\mathbb{Z}_b$. The jth coordinate $u_{i,j}$ of the ith point is obtained as $u_{i,j} = \sum_{l=1}^{k} y_{j,l} b^{-l}$, where

$$\begin{bmatrix} y_{j,1} \\ \vdots \\ y_{j,k} \end{bmatrix} = \mathbf{C}_j \begin{bmatrix} a_0 \\ \vdots \\ a_{k-1} \end{bmatrix}, \tag{11}$$

and the a_l's come from the digital expansion $i = \sum_{l=0}^{k-1} a_l b^l$ of i in base b. So $u_{i,j}$ is obtained by first applying a linear transformation determined by the matrix $\mathbf{C}_j$ to the coefficients $a_0, \ldots, a_{k-1}$ in the digital expansion of i, and then by applying the radical-inverse function to these transformed coefficients $y_{j,1}, \ldots, y_{j,k}$. It is possible to define digital nets in a base b that is not prime. One simply needs to choose a commutative ring R of cardinality b. The generating matrices then contain elements in that ring R, and bijections going from $\mathbb{Z}_b$ to R and from R to $\mathbb{Z}_b$ must be applied to the a_i's and the $y_{j,l}$, respectively. This setup can actually be used for a prime base b as well.

Related to digital nets are *digital sequences*, which are infinite sequences of points rather than point sets P_n of finite size n. Points from a digital sequence are obtained as above, but with generating matrices of infinite size.

We now describe three special cases of digital sequences: the Halton, Sobol' and Faure sequences. As we outlined previously, an s-dimensional Halton sequence is obtained by juxtaposing s van der Corput sequences in different bases. More precisely, the jth coordinate of the ith point is obtained as

$$u_{ij} = \varphi_{b_j}(i),$$

where the b_j, $j = 1, \ldots, s$, are integers larger than 1 typically chosen as the first s prime numbers. Formally, this is not a digital sequence, but we feel it is important to present this construction because of its wide use in practice (e.g., see Spanier, 1995; Kollig and Keller, 2002). Many improvements have been proposed for this sequence since its introduction in 1960 (Braaten and Weller, 1979; Struckmeier, 1995; Tuffin, 1998).

The Sobol' sequence is a digital sequence in base 2. It was proposed by Sobol' (1967) before the concept of digital net was defined in Niederreiter

(1987). Thus, its definition does not directly use generating matrices, although it can be defined that way. Instead, it relies on *direction numbers* $v_{j,1}, v_{j,2}, \ldots$ that must be chosen for each coordinate $j = 1, \ldots, s$. These direction numbers are rational numbers of the form

$$v_{j,l} = \frac{m_{j,l}}{2^l} = \sum_{p=1}^{l} v_{j,l,p} 2^{-p}, \quad l \geqslant 1,$$

where $m_{j,l}$ is an odd integer smaller than 2^l. Also needed is a primitive polynomial $f_j(z) = z^q + \alpha_{j,1} z^{q-1} + \cdots + \alpha_{j,q}$ over $\mathbb{F}_2$, the finite field with two elements, for each $j = 1, \ldots, s$. The method described in Sobol' (1967) chooses $f_j(z)$ to be the jth polynomial in a list of primitive polynomials over $\mathbb{F}_2$ sorted by increasing degree.

For a given j, if $f_j(z)$ is of degree $q(j)$, then only the first $q(j)$ direction numbers $v_{j,1}, \ldots, v_{j,q(j)}$ must be chosen. The next ones are obtained through the recurrence

$$v_{j,l} = \alpha_{j,1} v_{j,l-1} \oplus \cdots \oplus \alpha_{j,q(j)-1} v_{j,l-q(j)+1} \oplus \frac{v_{j,l-q(j)}}{2^{q(j)}}, \quad l > q(j),$$

where $\oplus$ denotes a bit-by-bit exclusive-or operation and $v_{j,l-q}/2^{q(j)}$ means that the binary expansion of $v_{j,q(j)}$ is shifted by $q(j)$ positions to the right. The jth coordinate of the ith point in the sequence is then defined as

$$u_{i,j} = a_0 v_{j,1} \oplus \cdots \oplus a_{k-1} v_{j,k},$$

where $a_0, \ldots, a_{k-1}$ are the coefficients in the binary expansion of i.

Hence the parameters that must be chosen are the $q(j)$ first direction numbers for each $j = 1, \ldots, s$. Equivalently, one must specify $m_{j,1}, \ldots, m_{j,q(j)}$ for each j. Sobol' and Levitan (1976) give a table of values of $m_{j,l}$ for $j \leqslant 40$ that is used in the implementation given by Bratley and Fox (1988). Other implementations of the Sobol' sequence can be found in the Finder software (*http://www.cs.columbia.edu/~ap/html/finder.html*), the SamplePack software (*http://www.uni-kl.de/AG-Heinrich/SamplePack.html*) and the RandQMC library (*http://www.math.ucalgary.ca/~lemieux/randqmc.html*, Lemieux et al., 2002).

The quality of digital nets and sequences is often measured by a quantity called t. Niederreiter (1992) calls an s-dimensional digital net in base b with b^m points a (t, m, s)-net, where t refers to this quality parameter. The smaller t is, the more uniform the net is. Similarly, an s-dimensional digital (t, s)-sequence is a digital sequence such that for each $m \geqslant 0$, the first b^m points form a (u, m, s)-net, for some $u \leqslant t$. Sobol' (1967) calls his sequence an LP_τ-sequence, where τ refers to this value of t. The Faure sequence is a digital sequence with $t = 0$ (Faure, 1982). For this property to hold, we must have $b \geqslant s$, and thus the base b increases with s. This sequence uses the Pascal matrix $\mathbf{P}$, whose entry on the kth row and lth column is $\binom{k}{l}$ for $k \geqslant l$, and 0 otherwise. More precisely, the generating matrix $\mathbf{C}_j$ for the Faure sequence is

taken to be the transpose of the Pascal matrix $\mathbf{P}$ raised to the power $j-1$, and where each entry is reduced modulo b. Tezuka (1995) has introduced a construction called a *generalized Faure sequence*, in which $\mathbf{C}_j = \mathbf{A}_j(\mathbf{P}^{\mathrm{T}})^{j-1}$, where each $\mathbf{A}_j$ is a non-singular lower-triangular matrix. Specific choices for these $\mathbf{A}_j$ can be found in Tezuka and Tokuyama (1994) and Faure (2001).

5 Recurrence-based point sets

We have seen in Chapter 3 how to define highly-uniform point sets by using a pseudorandom number generator (PRNG). Let us first recall briefly how this construction works.

Let R be a finite set. Assume we have a PRNG based on a transition function ψ from R to R and an output function g from R to $[0, 1)$. The sequence $x_0, x_1, \ldots$, obtained using the recurrence

$$x_i = \psi(x_{i-1}), \quad i \geqslant 1, \tag{12}$$

has a period of at most $|R|$. The associated *recurrence-based point set* is defined by

$$P_n = \big\{\big(g(x_0), g(x_1), \ldots, g(x_{s-1})\big)\colon\ x_0 \in R\big\}.$$

Thus $n = |R|$ for this construction, as long as g is one-to-one. As seen in Chapter 3, different kinds of PRNGs can been used in this manner to define highly-uniform point sets. For example, recall from Section 2 that using an LCG yields a Korobov point set.

Let us now discuss properties of recurrence-based point sets that are useful in practice. First, these point sets are fully projection regular and dimension-stationary. In addition, they can handle problems where the integrand f has an infinite dimension. This is because a given point in P_n is of the form

$$\mathbf{u}_i = \big(g(x_0), \ldots, g(x_{s-1})\big)$$

for some $x_0 \in R$. Therefore the size s of $\mathbf{u}_i$ can be increased indefinitely simply by continuing to run the recurrence (12). Of course, this means that if s exceeds $|R|$, the coordinates in $\mathbf{u}_i$ will eventually repeat. This would be bad if no randomization was applied, since it would mean that a simulation based on $\mathbf{u}_i$ reuses the same input numbers. However, by randomizing P_n appropriately, there will not be any repetition. For example, suppose we use a random shift $\mathbf{v} = (v_1, v_2, \ldots, v_s)$ as in Section 2 to obtain $\tilde{\mathbf{u}}_i = (\mathbf{u}_i + \mathbf{v}) \bmod 1$. Then even if, say, $u_{i,j} = u_{i,j+|R|}$, since $v_j \neq v_{j+|R|}$ with probability 1, then $\tilde{u}_{i,j} \neq \tilde{u}_{i,j+|R|}$ with probability 1 as well.

This brings us to the discussion of some implementation ideas that can be useful in the simulation context. As noted above, recurrence-based point sets can be used for infinite-dimensional problems. One way to implement this is to choose an a priori size S for the dimension of the point set. When a point

$\mathbf{u}_i$ needs to be of a larger dimension than S (i.e., the simulation based on that point needs more than S random numbers), we just need to know which element of R produced the last coordinate of $\mathbf{u}_i$, i.e., which $x_{S-1} \in R$ is such that $u_{i,S} = g(x_{S-1})$. Then we can obtain $u_{i,S+1}, u_{i,S+2}, \ldots$, by computing $x_S, x_{S+1}, \ldots$ from (12) and putting $u_{i,S+l+1} = g(x_{S+l})$, for $l \geqslant 0$.

Another possibility is to use an alternative definition for P_n, which says that P_n contains the overlapping vectors of all the cycles of a PRNG. More precisely,

$$P_n = \bigcup_{x_0 \in R} C(x_0), \tag{13}$$

where

$$C(x_0) = \big\{\big(g(x_i), g(x_{i+1}), \ldots, g(x_{i+s-1})\big)\colon\ 0 \leqslant i < \tau(x_0)\big\},$$

and $\tau(x_0)$ is the period of the sequence $\{x_l, l \geqslant 0\}$. Of course, if $x, y \in R$ are in the same cycle, then $C(x) = C(y)$, so the number of distinct sets in the union (13) is equal to the number of distinct cycles of the PRNG. The idea is then to store these distinct cycles into separate arrays and record their respective lengths. Arbitrary coordinates $u_{i,j}$ can then be easily retrieved by doing a simple arithmetic computation to figure out which cycle contains $\mathbf{u}_i$ and in what position. Here is an example to illustrate how it works.

Example 4. Suppose we have a PRNG over a finite set R of size 128, with 4 cycles of respective length $\tau_1 = 1$, $\tau_2 = 7$, $\tau_3 = 15$ and $\tau_4 = 105$. Let u_l^c be the lth output number in the cth cycle. Then P_n can be defined so that for $i = 0, \ldots, n-1$,

$$\mathbf{u}_i = \big(u_{l(i)}^{c(i)}, u_{l(i)+1 \bmod \tau_{c(i)}}^{c(i)}, \ldots, u_{l(i)+s-1 \bmod \tau_{c(i)}}^{c(i)}\big),$$

where

$$\big(c(i), l(i)\big) = \begin{cases} (1, 1) & \text{if } i = 0, \\ (2, i) & \text{if } 1 \leqslant i < 8, \\ (3, i-7) & \text{if } 8 \leqslant i < 23, \\ (4, i-22) & \text{if } i \geqslant 23. \end{cases}$$

This alternative implementation has been used in Dembeck (2003). It is very useful in cases where the dimension is very large, e.g., $s \gg n$, because instead of having to store an s-dimensional vector representing the point, coordinates can be generated online, and only the n numbers output by the PRNG in all its cycles need to be stored. This is also useful when one needs to partition points into blocks that are not used at the same time. This can happen when information is sequentially gathered through time to update the simulation, such as in *sequential Monte Carlo methods* (Doucet et al., 2001), or in the context of *perfect simulation* (Propp and Wilson, 1996). When s is not too large and points of a fixed dimension s are used one after the other in the simulation, the $\mathbf{u}_i$'s

should instead be generated by shifting the coordinates by one position in the current cycle (unless the index i corresponds to a change of cycle $c(i)$, in which case the point has to be filled in from the first s numbers of the next cycle). This kind of implementation is used in the QMC portion of the SSJ package (L'Ecuyer et al., 2002).

6 Randomization techniques and variance results

We already saw in Section 2 a simple randomization method that consists in adding a random uniform vector $\mathbf{v} \in [0,1)^s$ to each point in P_n. Although this method can be used to randomize any highly-uniform point set (Lemieux and L'Ecuyer, 2000; Morohosi and Fushimi, 2000; Kollig and Keller, 2002), it was originally proposed for lattice rules by Cranley and Patterson (1976). Below, we discuss this randomization and two other ones that are better suited for digital nets. But first, we outline a general framework that includes most randomization methods.

6.1 General principles

Let $P_n = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\}$ be a highly-uniform point set, and let $\mathbf{v}$ be a uniform random vector in some space Ω. To randomize P_n, a randomization function $r : \Omega \times [0,1)^s \to [0,1)^s$ is needed in order to obtain a randomized version $\widetilde{P}_n = \{\tilde{\mathbf{u}}_0, \ldots, \tilde{\mathbf{u}}_{n-1}\}$ of P_n defined by

$$\tilde{\mathbf{u}}_i = r(\mathbf{v}, \mathbf{u}_i).$$

For example, in the Cranley–Patterson method, $\Omega = [0,1)^s$ and $r(\mathbf{v}, \mathbf{u}_i) = (\mathbf{u}_i + \mathbf{v}) \bmod 1$.

As mentioned in the Introduction, the function r should be chosen so that (i) $r(\mathbf{v}, \mathbf{u})$ is uniformly distributed over $[0,1)^s$ for each $\mathbf{u}$, and (ii) $\widetilde{P}_n$ has the same highly-uniform properties as P_n. Another property that holds for most randomization techniques is that they can be written in a product form, i.e., Ω is of the form $\Omega = \widetilde{\Omega} \times \cdots \times \widetilde{\Omega}$, and $r(\mathbf{v}, \mathbf{u}) = (\tilde{r}(v_1, u_1), \ldots, \tilde{r}(v_s, u_s))$, where $v_j \in \widetilde{\Omega}$, and $\tilde{r} : \widetilde{\Omega} \times [0,1) \to [0,1)$. This property simplifies the implementation when s is infinite, since when additional coordinates are required, one simply needs to generate additional random components v in $\widetilde{\Omega}$. In practice, this can be done by using similar ideas as those discussed in Section 5, i.e., one can fix an a priori bound S, generate and store $v_1, \ldots, v_S$, memorize the state x_S of the PRNG used to generate the v_j. Then, when additional coordinates need to be randomized, set the PRNG's state to x_S and generate v_{S+1}, v_{S+2}, etc.

6.2 Shift modulo 1

This is the Cranley–Patterson method that we have already discussed. It is easy to see that this randomization method satisfies property (i) above. As

for (ii), assume that P_n is a lattice point set and that its uniformity is measured by the largest distance between adjacent parallel hyperplanes that together cover the points in P_n (this is the quantity measured by the spectral test discussed in Chapter 3). Obviously, the uniformity of $\widetilde{P}_n$ is then the same as that of P_n. This randomization also satisfies the product-form property.

In the case where $P_n = L \cap [0,1)^s$ is a lattice point set, the variance of the estimator $\hat{\mu}_{\mathrm{RQMC}}$ based on a randomly shifted lattice point set $\widetilde{P}_n$ satisfies the following (L'Ecuyer and Lemieux, 2000):

Proposition 5. *If f is square-integrable, then for a randomly shifted lattice point set we have*

$$\mathrm{Var}(\hat{\mu}_{\mathrm{RQMC}}) = \sum_{\mathbf{0}\neq\mathbf{h}\in L^\perp} \left|\hat{f}(\mathbf{h})\right|^2,$$

where $L^\perp = \{\mathbf{h} \in \mathbb{R}^s\colon\ \mathbf{h}\cdot\mathbf{u}_i \in \mathbb{Z}$ for each $\mathbf{u}_i \in P_n\}$ is the dual lattice of L and $\hat{f}(\mathbf{h})$ is the Fourier coefficient of f in $\mathbf{h}$.

In comparison, for the MC estimator we have

$$\mathrm{Var}(\hat{\mu}_{\mathrm{MC}}) = \frac{1}{n}\sum_{\mathbf{0}\neq\mathbf{h}\in\mathbb{Z}^s} \left|\hat{f}(\mathbf{h})\right|^2.$$

It can be shown that the dual lattice $L^\perp$ has n times less elements than $\mathbb{Z}^s$ (see, e.g., Sloan and Joe, 1994). Using this fact and the above proposition, we can see that for the randomly shifted lattice estimator to have a smaller variance than the MC estimator, the squared Fourier coefficients of f must be smaller on average over the dual lattice than they are over $\mathbb{Z}^s$. In practice, f is often such that the largest Fourier coefficients are those associated with small $\mathbf{h}$'s. Hence one could argue that to get an estimator with a smaller variance than the MC estimator, the lattice L on which P_n is based should be chosen so that its dual lattice does not contain short vectors. Recall from Chapter 3 that the quantity measured by the spectral test is also equal to the inverse of the length of the shortest vector in the dual lattice $L^\perp$. This means that using criteria based on the spectral test such as those discussed in Chapter 3 can be justified from the above variance analysis.

We refer the reader to L'Ecuyer and Lemieux (2002) and the references therein for more details on criteria that can be used to choose lattice point sets, and to Morohosi and Fushimi (2000) for a variance analysis of randomly shifted digital nets. Note that although this randomization method does not provide a guaranteed variance reduction, it is often used in practice because of its simplicity.

6.3 Digital shift

The analog of the Cranley–Patterson method for a digital net in base b is to use a digital shift in base b (see L'Ecuyer and Lemieux (2002) and the refer-

ences therein). In this case, the randomization function r is defined by

$$r(\mathbf{v}, \mathbf{u}) = \mathbf{u} \bigoplus_b \mathbf{v},$$

where $\mathbf{v} \in \Omega = [0,1)^s$ and $\bigoplus_b$ is a coordinate-wise addition of the base b expansion of $\mathbf{u}$ and $\mathbf{v}$, i.e., for $\mathbf{u} = (\sum_{l=1}^{\infty} u_{1,l} b^{-l}, \ldots, \sum_{l=1}^{\infty} u_{s,l} b^{-l})$ and $\mathbf{v} = (\sum_{l=1}^{\infty} v_{1,l} b^{-l}, \ldots, \sum_{l=1}^{\infty} v_{s,l} b^{-l})$, we have that

$$\mathbf{u} \bigoplus_b \mathbf{v} = \left(\sum_{l=1}^{\infty} \big((u_{1,l} + v_{1,l}) \bmod b\big) b^{-l}, \right.$$
$$\left. \ldots, \sum_{l=1}^{\infty} \big((u_{s,l} + v_{s,l}) \bmod b\big) b^{-l} \right).$$

The uniformity of P_n as measured by the parameter t and the *resolution* (see Chapter 3) is preserved by this randomization.

The variance of a digitally shifted net can be analyzed in a way that closely mimics the results for shifted lattice rules: the Fourier coefficients are instead coefficients from the Walsh expansion of f, and a notion of *dual space* of the digital net P_n replaces the dual lattice in the above analysis. We refer the reader to L'Ecuyer and Lemieux (2002) and L'Ecuyer (2004) for more details. In particular, more details can be found in these papers on selection criteria for digital nets that can be related to the variance expression obtained in that context.

6.4 Scrambling

The scrambling method was proposed by Owen (1995) to randomize digital nets. Alternative scrambling strategies have been studied in Matoušek (1998), Faure et al. (2001), Hong and Hickernell (2003), Faure and Tezuka (2003) and Owen (2003). Here we briefly discuss two of them, using the terminology from Owen (2003): the *nested uniform scrambling* of Owen (1995) and the *affine matrix scrambling* of Matoušek (1998) (called *random linear scrambling* in that paper).

Let P_n be a digital net in base b. Nested uniform scrambling applies random uniform permutations to the digits of each coordinate $u_{i,j}$ in its base b expansion. More precisely, if we write

$$u_{i,j} = \sum_{l=1}^{\infty} y_l b^{-l},$$

then the nested uniform scrambling maps $u_{i,j}$ to

$$\tilde{u}_{i,j} = \sum_{l=1}^{\infty} \pi_{\cdot, y_1, \ldots, y_{l-1}}(y_l) b^{-l},$$

where $\pi_{\cdot,y_1,\ldots,y_{l-1}}(\cdot)$ is a random uniform permutation of $[0, 1, \ldots, b-1]$ that depends on $y_1, \ldots, y_{l-1}$. All permutations used for a given j and across different values of j are independent.

The cost in time and space required by this method is quite large, and for that reason several alternatives have been proposed. Here we discuss only one and refer the reader to Owen (2003) for additional information. In affine matrix scrambling, we have that $\tilde{u}_{i,j} = \sum_{l=1}^{\infty} \tilde{y}_l b^{-l}$, with

$$\tilde{y}_l = \sum_{k=1}^{l} L_{l,k} y_k + d_l,$$

where $L_{l,l} \in \mathbb{Z}_b \setminus \{0\}$, $L_{l,k} \in \mathbb{Z}_b$ for $k < l$, and $d_l \in \mathbb{Z}_b$ are randomly and uniformly chosen. An alternative description is to say that with this scrambling, each generating matrix is multiplied by a random lower-triangular matrix of infinite size with elements in $\mathbb{Z}_b$ and nonzero elements on the diagonal. A digital shift is then performed.

These two scrambling methods produce estimators $\hat{\mu}_{\mathrm{RQMC}}$ satisfying the following proposition.

Proposition 6 (Owen, 1998b; Hong and Hickernell, 2003; Owen, 2003). *Under either scrambling method described above, if P_n is a (t, m, s)-net in base b, then for any square-integrable function f,*

$$\mathrm{Var}(\hat{\mu}_{\mathrm{RQMC}}) \leqslant b^t \left(\frac{b+1}{b-1}\right)^s \frac{\sigma^2}{n}.$$

If f is sufficiently smooth (i.e., its mixed partial derivatives satisfy a Lipschitz condition: see Owen (1998b) for the details), then

$$\mathrm{Var}(\hat{\mu}_{\mathrm{RQMC}}) \in \mathrm{O}\big(n^{-3} \log^s n\big).$$

What this result says is that estimators based on scrambled nets cannot do worse than the MC estimator (up to a constant with respect to n), and that for a function that is smooth enough, the variance is in $\mathrm{O}(n^{-3} \log^s n)$, which is significantly better than the $\mathrm{O}(n^{-1})$ that we get for the MC estimator.

7 Combination with other variance reduction techniques

It is a natural idea in simulation to try combining different variance reduction techniques, hoping that their individual beneficial effect will add up (or even be enhanced) in the combination. Since RQMC methods can be seen as a way to produce estimators with lower variance than MC, it makes sense to try combining them with more standard variance reduction techniques such as those discussed in Chapters 10 and 11. As one would expect, care must be taken when doing so if we want to make sure that the combination will not backfire

and produce undesirable effects. To illustrate this point, we briefly discuss in this section the combination of RQMC methods with control variates (CV), as studied in Hickernell et al. (2005).

When RQMC is combined with CV, the optimal control variate coefficients are not necessarily the same as when CV is used with plain MC. The idea is as follows. Assume we have a control variable described by a function $g(\mathbf{u})$ over $[0, 1)^s$, with known expectation μ_g. We are looking for the value of β that minimizes the variance of

$$\hat{\mu}_{\mathrm{RQMC}}(f) + \beta\big(\mu_g - \hat{\mu}_{\mathrm{RQMC}}(g)\big), \tag{14}$$

where $\hat{\mu}_{\mathrm{RQMC}}(f)$ and $\hat{\mu}_{\mathrm{RQMC}}(g)$ are the RQMC estimators of μ and μ_g, respectively. As seen in Chapter 10, with plain MC the optimal β is given by $\beta^*_{\mathrm{MC}} = \mathrm{Cov}(f(\mathbf{u}), g(\mathbf{u}))/\mathrm{Var}(g(\mathbf{u}))$. In the RQMC context, it is clear from (14) that the optimal β should instead be

$$\beta^*_{\mathrm{RQMC}} = \frac{\mathrm{Cov}(\hat{\mu}_{\mathrm{RQMC}}(f), \hat{\mu}_{\mathrm{RQMC}}(g))}{\mathrm{Var}(\hat{\mu}_{\mathrm{RQMC}}(g))},$$

which in general is not equal to β^*_{MC}. The optimal β^*_{RQMC} can be estimated using i.i.d. replications of $\hat{\mu}_{\mathrm{RQMC}}(f)$ and $\hat{\mu}_{\mathrm{RQMC}}(g)$, as is done to estimate the variance of $\hat{\mu}_{\mathrm{RQMC}}(f)$. Although there are some pathological cases for which using β^*_{MC} instead of β^*_{RQMC} can lead to a significant increase in variance, experiments in Hickernell et al. (2005) suggest that for problems frequently encountered in practice, using the wrong optimal coefficient may not significantly affect the variance. We refer the reader to Hickernell et al. (2005) for more on this topic, including asymptotic variance analyses and alternative ways to estimate β^*_{RQMC}.

8 Future directions

There is still much work to do in order to make quasi-random numbers a well-known and widely used tool in simulation. First, it would be useful to have simulation packages that incorporate several RQMC methods as alternatives to the use of pseudorandom numbers. Progress in this direction may come from a package developed by Pierre L'Ecuyer and his collaborators (L'Ecuyer et al., 2002).

As seen in Section 3, the interplay between the function f to be integrated and the highly-uniform point set P_n used plays an important role in the success of RQMC methods. Being able to extract important features of f in an online fashion would thus be useful for choosing an appropriate construction for P_n. For example, being able to know what are the important projections of f in its ANOVA decomposition, or what its effective dimension is in some sense would be helpful. Recent papers (Lemieux and Owen, 2001; Jiang and Owen, 2003; Liu and Owen, 2003; Wang and Fang, 2003; Wang and Sloan, 2003) take some steps in that direction.

The search for better constructions is another area where new contributions would be helpful. For example, the Sobol' sequence is widely used but in practice, its implementation requires an upper bound on the dimension s. It would be desirable to have alternative constructions in base 2 (for a quick implementation) with an infinite number of points and dimension. Niederreiter (2003) gives theoretical results proving the existence of such sequences, but no constructions have been suggested so far.

Acknowledgements

The author wishes to thank the editors and anonymous reviewer for their helpful comments and suggestions. This work was supported by NSERC Grant #RGP238959.

References

Acworth, P., Broadie, M., Glasserman, P. (1997). A comparison of some Monte Carlo and quasi-Monte Carlo techniques for option pricing. In: Hellekalek, P., Niederreiter, H. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*. *Lecture Notes in Statistics*, vol. 127. Springer-Verlag, New York, pp. 1–18.

Åkesson, F., Lehoczy, J.P. (2000). Path generation for quasi-Monte Carlo simulation of mortgage-backed securities. *Management Science* 46, 1171–1187.

Braaten, E., Weller, G. (1979). An improved low-discrepancy sequence for multidimensional quasi-Monte Carlo integration. *Journal of Computational Physics* 33, 249–258.

Bratley, P., Fox, B.L. (1988). Algorithm 659: Implementing Sobol's quasirandom sequence generator. *ACM Transactions on Mathematical Software* 14 (1), 88–100.

Business Week (1994). Suddenly, number theory makes sense to industry. *Business Week* (June 20, 1994), 172–174.

Caflisch, R.E., Moskowitz, B. (1995). Modified Monte Carlo methods using quasi-random sequences. In: Niederreiter, H., Shiue, P.J.-S. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*. *Lecture Notes in Statistics*, vol. 106. Springer-Verlag, New York, pp. 1–16.

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

Cranley, R., Patterson, T.N.L. (1976). Randomization of number theoretic methods for multiple integration. *SIAM Journal on Numerical Analysis* 13 (6), 904–914.

Davis, P., Rabinowitz, P. (1984). *Methods of Numerical Integration*, 2nd edition. Academic Press, New York.

Dembeck, D.C. (2003). Dynamic numerical integration using randomized quasi-Monte Carlo methods. Master's thesis, University of Calgary.

Doucet, A., de Freitas, N., Gordon, N. (Eds.) (2001). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, Berlin.

Faure, H. (1982). Discrépance des suites associées à un système de numération. *Acta Arithmetica* 61, 337–351.

Faure, H. (2001). Variations on $(0, s)$-sequences. *Journal of Complexity* 17, 741–753.

Faure, H., Tezuka, S. (2001). Another random scrambling of digital (t, s) sequences. In: Fang, K.-T., Hickernell, F., Niederreiter, H. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2000*. Springer-Verlag, Berlin, pp. 242–256.

Faure, H., Tezuka, S. (2003). I-binomial scrambling of digital nets and sequences. *Journal of Complexity* 19, 744–757.

Fox, B.L. (1999). *Strategies for Quasi-Monte Carlo*. Kluwer Academic Publishers, Boston, MA.

Haber, S. (1983). Parameters for integrating periodic functions of several variables. *Mathematics of Computation* 41, 115–129.

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

Hickernell, F.J. (1998a). A generalized discrepancy and quadrature error bound. *Mathematics of Computation* 67, 299–322.

Hickernell, F.J. (1998b). Lattice rules: How well do they measure up? In: Hellekalek, P., Larcher, G. (Eds.), *Random and Quasi-Random Point Sets*. *Lecture Notes in Statistics*, vol. 138. Springer-Verlag, New York, pp. 109–166.

Hickernell, F.J. (1998c). What affects accuracy of quasi-Monte Carlo quadrature? In: Niederreiter, H., Spanier, J. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 1998*. Springer-Verlag, New York, pp. 16–55.

Hickernell, F.J., Hong, H.S. (1997). Computing multivariate normal probabilities using rank-1 lattice sequences. In: Golub, G.H., Lui, S.H., Luk, R.J., Plemmons, F.T. (Eds.), *Proceedings of the Workshop on Scientific Computing (Hong Kong)*. Springer-Verlag, Singapore, pp. 209–215.

Hickernell, F.J., Niederreiter, H. (2003). The existence of good extensible rank-1 lattices. *Journal of Complexity* 19, 286–300.

Hickernell, F.J., Wang, X. (2001). The error bounds and tractability of quasi-Monte Carlo methods in infinite dimension. *Mathematics of Computation* 71, 1641–1661.

Hickernell, F.J., Hong, H.S., L'Ecuyer, P., Lemieux, C. (2001). Extensible lattice sequences for quasi-Monte Carlo quadrature. *SIAM Journal on Scientific Computing* 22, 1117–1138.

Hickernell, F.J., Lemieux, C., Owen, A.B. (2005). Control variates for quasi-Monte Carlo. *Statistical Science* 20, 1–31.

Hoeffding, W. (1948). A class of statistics with asymptotically normal distributions. *Annals of Mathematical Statistics* 19, 293–325.

Hong, H., Hickernell, F. (2003). Algorithm 823: Implementing scrambled digital sequences. *ACM Transactions on Mathematical Software* 29, 95–109.

Jiang, T., Owen, A.B. (2003). Quasi-regression with shrinkage. *Mathematics and Computers in Simulation* 62, 231–241.

Kollig, T., Keller, A. (2002). Efficient bidirectional path tracing by randomized quasi-Monte Carlo integration. In: Fang, T., Hickernell, F.J., Niederreiter, H. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2000*. Springer-Verlag, Berlin, pp. 290–305.

Korobov, N.M. (1959). The approximate computation of multiple integrals. *Doklady Akademii Nauk SSSR* 124, 1207–1210 (in Russian).

Law, A.M., Kelton, W.D. (2000). *Simulation Modeling and Analysis*, 3rd edition. McGraw-Hill, New York.

L'Ecuyer, P. (2004). Polynomial integration lattices. In: Niederreiter, H. (Ed.), *Monte Carlo and Quasi-Monte Carlo Methods 2002*. Springer-Verlag, Berlin, pp. 73–98.

L'Ecuyer, P., Lemieux, C. (2000). Variance reduction via lattice rules. *Management Science* 46 (9), 1214–1235.

L'Ecuyer, P., Lemieux, C. (2002). Recent advances in randomized quasi-Monte Carlo methods. In: Dror, M., L'Ecuyer, P., Szidarovszki, F. (Eds.), *Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications*. Kluwer Academic Publishers, Boston, pp. 419–474.

L'Ecuyer, P., Meliani, L., Vaucher, J. (2002). SSJ: A framework for stochastic simulation in Java. In: Dror, M., Yücesan, E., Chen, C.-H., Snowdon, J.L., Charnes, J.M. (Eds.), *Proceedings of the 2002 Winter Simulation Conference*. IEEE Press, New York, pp. 234–242.

Lemieux, C., L'Ecuyer, P. (1999). Lattice rules for the simulation of ruin problems. In: *Proceedings of the 1999 European Simulation Multiconference, vol. 2*. The Society for Computer Simulation, Ghent, Belgium, pp. 533–537.

Lemieux, C., L'Ecuyer, P. (2000). A comparison of Monte Carlo, lattice rules and other low-discrepancy point sets. In: Niederreiter, H., Spanier, J. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 1998*. Springer-Verlag, Berlin, pp. 326–340.

Lemieux, C., L'Ecuyer, P. (2003). Randomized polynomial lattice rules for multivariate integration and simulation. *SIAM Journal on Scientific Computing* 24 (5), 1768–1789.

Lemieux, C., Owen, A. (2001). Quasi-regression and the relative importance of the ANOVA components of a function. In: Fang, T., Hickernell, F.J. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2000*. Springer-Verlag, Berlin, pp. 331–344.

Lemieux, C., Cieslak, M., Luttmer, K. (2002). RandQMC user's guide: A package for randomized quasi-Monte Carlo methods in C. Technical Report 2002-712-15, Department of Computer Science, University of Calgary.

Liu, R., Owen, A.B. (2003). Estimating mean dimensionality. Technical Report 2003-14, Department of Statistics, Stanford University.

Matoušek, J. (1998). On the L_2-discrepancy for anchored boxes. *Journal of Complexity* 14, 527–556.

Morohosi, H., Fushimi, M. (2000). A practical approach to the error estimation of quasi-Monte Carlo integration. In: Niederreiter, H. (Ed.), *Monte Carlo and Quasi-Monte Carlo Methods 1998*. Springer-Verlag, Berlin, pp. 377–390.

Morokoff, W.J., Caflisch, R.E. (1994). Quasi-random sequences and their discrepancies. *SIAM Journal on Scientific Computing* 15, 1251–1279.

Morokoff, W.J., Caflisch, R.E. (1997). Quasi-Monte Carlo simulation of random walks in finance. In: Hellekalek, P. (Ed.), *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*. *Lecture Notes in Statistics*, vol. 127. Springer-Verlag, New York, pp. 340–352.

New York Times (1995). From I.B.M., help in intricate trading. *The New York Times* (September 25, 1995).

Niederreiter, H. (1987). Point sets and sequences with small discrepancy. *Monatshefte für Mathematik* 104, 273–337.

Niederreiter, H. (1992). *Random Number Generation and Quasi-Monte Carlo Methods*. *SIAM CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 63. SIAM, Philadelphia.

Niederreiter, H. (2003). The existence of good extensible polynomial lattice rules. *Monatshefte für Mathematik* 139, 295–307.

Niederreiter, H., Xing, C. (1996). Low-discrepancy sequences and global function fields with many rational places. *Finite Fields and Their Applications* 2, 241–273.

Ninomiya, S., Tezuka, S. (1996). Toward real-time pricing of complex financial derivatives. *Applied Mathematical Finance* 3, 1–20.

Owen, A.B. (1995). Randomly permuted (t, m, s)-nets and (t, s)-sequences. In: Niederreiter, H., Shiue, S. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*. *Lecture Notes in Statistics*, vol. 106. Springer-Verlag, New York, pp. 299–317.

Owen, A.B. (1997a). Monte Carlo variance of scrambled equidistribution quadrature. *SIAM Journal on Numerical Analysis* 34 (5), 1884–1910.

Owen, A.B. (1997b). Scrambled net variance for integrals of smooth functions. *The Annals of Statistics* 25 (4), 1541–1562.

Owen, A.B. (1998a). Latin supercube sampling for very high-dimensional simulations. *ACM Transactions on Modeling and Computer Simulation* 8 (1), 71–102.

Owen, A.B. (1998b). Scrambling Sobol and Niederreiter–Xing points. *Journal of Complexity* 14, 466–489.

Owen, A.B. (2002). Necessity of low effective dimension. Manuscript.

Owen, A.B. (2003). Variance and discrepancy with alternative scramblings. *ACM Transactions on Modeling and Computer Simulation* 13, 363–378.

Papageorgiou, A. (2002). The Brownian bridge does not offer a consistent advantage in quasi-Monte Carlo integration. *Journal of Complexity* 18 (1), 171–186.

Papageorgiou, A. (2003). Sufficient conditions for fast quasi-Monte Carlo convergence. *Journal of Complexity* 19 (3), 332–351.

Paskov, S. (1997). New methodologies for valuing derivatives. In: Pliska, S., Dempster, M. (Eds.), *Mathematics of Derivative Securities*. Cambridge University Press, Cambridge, UK, pp. 545–582.

Paskov, S., Traub, J. (1995). Faster valuation of financial derivatives. *Journal of Portfolio Management* 22, 113–120, see also *http://www.cs.columbia.edu/~ap/html/information.html*.

Propp, J.G., Wilson, D.B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms* 9 (1–2), 223–252.

Richtmyer, R.D. (1951). On the evaluation of definite integrals and a quasi-Monte Carlo method based on properties of algebraic numbers. Technical Report LA-1342, Los Alamos Scientific Laboratory.

Sloan, I.H., Joe, S. (1994). *Lattice Methods for Multiple Integration*. Clarendon Press, Oxford.

Sloan, I.H., Rezstov, A. (2002). Component-by-component construction of good lattice rules. *Mathematics of Computation* 71, 263–273.

Sloan, I.H., Walsh, L. (1990). A computer search of rank 2 lattice rules for multidimensional quadrature. *Mathematics of Computation* 54, 281–302.

Sloan, I.H., Woźniakowski, H. (1998). When are quasi-Monte Carlo algorithms efficient for high dimensional integrals? *Journal of Complexity* 14, 1–33.

Sloan, I.H., Kuo, F.Y., Joe, S. (2002). On the step-by-step construction of quasi-Monte Carlo integration rules that achieve strong tractability error bounds in weighted Sobolev spaces. *Mathematics of Computation* 71, 1609–1640.

Sobol', I.M. (1967). The distribution of points in a cube and the approximate evaluation of integrals. *Computational Mathematics and Mathematical Physics (Moscow)* 7, 86–112.

Sobol', I.M. (2001). Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation* 55, 271–280.

Sobol', I.M., Asostsky, D.I. (2003). One more experiment on estimating high-dimensional integrals by quasi-Monte Carlo methods. *Mathematics and Computers in Simulation* 62, 255–263.

Sobol', I.M., Levitan, Y.L. (1976). The production of points uniformly distributed in a multidimensional cube. Preprint 40, Institute of Applied Mathematics, USSR Academy of Sciences. (In Russian.)

Spanier, J. (1995). Quasi-Monte Carlo methods for particle transport problems. In: Niederreiter, H., Shiue, S. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*. *Lecture Notes in Statistics*, vol. 106. Springer-Verlag, New York, pp. 121–148.

Struckmeier, J. (1995). Fast generation of low-discrepancy sequences. *Journal of Computational and Applied Mathematics* 91, 29–41.

Tan, K.S., Boyle, P. (2000). Applications of randomized low discrepancy sequences to the valuation of complex securities. *Journal of Economic Dynamics and Control* 24, 1747–1782.

Tezuka, S. (1995). *Uniform Random Numbers: Theory and Practice*. Kluwer Academic Publishers, Norwell, MA.

Tezuka, S., Tokuyama, T. (1994). A note on polynomial arithmetic analogue of Halton sequences. *ACM Transactions on Modeling and Computer Simulation* 4, 279–284.

The Economist (1995). Is Monte Carlo Bust? *The Economist* (August 12, 1995), 63.

Tuffin, B. (1998). A new permutation choice in Halton sequences. In: Hellekalek, P., Larcher, G., Niederreiter, H., Zinterhof, P. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods*. *Lecture Notes in Statistics*, vol. 127. Springer-Verlag, New York, pp. 427–435.

van der Corput, J.G. (1935). Verteilungsfunktionen: I, II. *Koninklijke Nederlandse Akademie van Wetenschappen, Proceedings* 38, 813–821, 1058–1066.

Wang, X., Fang, K.-T. (2003). The effective dimension and quasi-Monte Carlo integration. *Journal of Complexity* 19, 101–124.

Wang, X., Sloan, I.H. (2003). Why are high-dimensional finance problems of low effective dimension? Applied Mathematics Report AMR03/20, University of New South Wales.