Finance Stoch (2009) 13: 307–349
DOI 10.1007/s00780-009-0095-y

Quasi-Monte Carlo methods with applications in finance

Pierre L'Ecuyer

Received: 30 May 2008 / Accepted: 20 August 2008 / Published online: 23 July 2009

Abstract We review the basic principles of quasi-Monte Carlo (QMC) methods, the randomizations that turn them into variance-reduction techniques, the integration error and variance bounds obtained in terms of QMC point set discrepancy and variation of the integrand, and the main classes of point set constructions: lattice rules, digital nets, and permutations in different bases. QMC methods are designed to estimate s-dimensional integrals, for moderate or large (perhaps infinite) values of s. In principle, any stochastic simulation whose purpose is to estimate an integral fits this framework, but the methods work better for certain types of integrals than others (e.g., if the integrand can be well approximated by a sum of low-dimensional smooth functions). Such QMC-friendly integrals are encountered frequently in computational finance and risk analysis. We summarize the theory, give examples, and provide computational results that illustrate the efficiency improvement achieved. This article is targeted mainly for those who already know Monte Carlo methods and their application in finance, and want an update of the state of the art on quasi-Monte Carlo methods.

Keywords Monte Carlo · Quasi-Monte Carlo · Variance reduction · Effective dimension · Discrepancy · Hilbert spaces

Mathematics Subject Classification (2000) 65C05 · 68U20 · 91B28

JEL Classification C15 · C63

P. L'Ecuyer (✉)
DIRO, Université de Montréal, C.P. 6128, succ. Centre-Ville, Montréal, QC, H3C 3J7, Canada
e-mail: lecuyer@iro.umontreal.ca

1 Introduction

1.1 Monte Carlo

Monte Carlo (MC) simulation is the primary method for pricing complex financial derivatives, such as contracts whose payoff depends on several correlated assets or on the entire sample path of an asset price. The option price μ is written as an integral that represents the mathematical expectation of the discounted payoff under a so-called risk-neutral probability measure. This expectation is usually with respect to a non-uniform density over the real space, but with a change of variables, it can be rewritten as an integral over the s-dimensional unit hypercube $(0,1)^s = \{\mathbf{u} = (u_1, \ldots, u_s) : 0 < u_j < 1 \text{ for all } j\}$, i.e.,

$$\begin{aligned}\mu = \mu(f) &= \int_0^1 \cdots \int_0^1 f(u_1, \ldots, u_s)\, du_1 \cdots du_s = \int_{(0,1)^s} f(\mathbf{u})\, d\mathbf{u} \\ &= \mathbb{E}\big[f(\mathbf{U})\big], \end{aligned} \tag{1.1}$$

for some function $f : (0,1)^s \to \mathbb{R}$, where $\mathbf{u}$ represents a point in $(0,1)^s$, and $\mathbf{U} \sim U(0,1)^s$ is a random point with the uniform distribution over the unit hypercube [24, 44]. For example, if the discounted payoff is a function g of s independent standard normal random variables $Z_1, \ldots, Z_s$, we can write $f(\mathbf{u}) = g(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_s))$ where Φ is the distribution function of a standard normal random variable.

In this paper, we assume that the integral is already written in the form (1.1) for a fixed positive integer s, and we want to estimate μ. This s represents the number of calls to the underlying random number generator used in our simulation. In situations where this number of calls is random and unbounded, s can be taken as infinite, with the usual assumption that with probability one, only a finite number of coordinates of $\mathbf{U}$ need to be explicitly generated.

In the *Monte Carlo method*, the estimator is

$$\hat{\mu}_n = \frac{1}{n} \sum_{i=0}^{n-1} f(\mathbf{U}_i), \tag{1.2}$$

where $\mathbf{U}_0, \ldots, \mathbf{U}_{n-1}$ are independent random vectors uniformly distributed over $(0,1)^s$ and n is the number of replications (a constant). This estimator is unbiased and has variance σ^2/n, where

$$\sigma^2 := \mathrm{Var}\big[f(\mathbf{U}_i)\big] = \int_{(0,1)^s} f^2(\mathbf{u})\, d\mathbf{u} - \mu^2.$$

If $\sigma^2 < \infty$, then $\hat{\mu}_n$ obeys a central limit theorem, and we can rely on it to compute a confidence interval on μ, whose width converges roughly as $O(\sigma n^{-1/2})$. The use of MC for pricing financial options was first proposed by Boyle [6]. The techniques have evolved tremendously since then. Recent accounts can be found in [24, 44].

In this paper, we assume that the reader is already familiar with MC methods in finance, and our discussion is focused on the use of QMC as an alternative. Much of what we say applies (in principle) to any MC simulation model, and not only to models encountered in finance.

1.2 Quasi-Monte Carlo

Quasi-Monte Carlo replaces the independent random points $\mathbf{U}_i$ in (1.2) by a set of n deterministic points, $P_n = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\}$, which cover the unit hypercube $(0, 1)^s$ more evenly (uniformly) than a typical set of random points [79, 98]. The point set P_n is called a *design* by some statisticians. The estimator $\hat{\mu}_n$ is replaced by the deterministic approximation

$$\bar{\mu}_n = \frac{1}{n} \sum_{i=0}^{n-1} f(\mathbf{u}_i). \tag{1.3}$$

The two main classes of constructions for P_n, to be discussed later, are integration lattices and digital nets. QMC methods can also be designed for integrals over other areas than the unit cube, by constructing point sets directly in those areas, but transforming the integral to the form (1.1) is usually more convenient.

There are many ways of measuring the uniformity of P_n; this is usually done via measures of non-uniformity called *discrepancies* [17, 18, 28–31, 73, 79]. Many of them are defined as the worst-case integration error, with P_n, over all functions f of unit norm in some Hilbert (or Banach) space $\mathcal{H}$ [17, 29, 37, 102, 113]. In this type of setting, there holds a general worst-case error bound of the form

$$|\bar{\mu}_n - \mu| \le D(P_n) V(f) \tag{1.4}$$

for all $f \in \mathcal{H}$, where $V(f) = \|f - \mu\|_{\mathcal{H}}$, the norm of $f - \mu$ in $\mathcal{H}$, can be interpreted as a measure of *variation* of the function f, and $D(P_n)$ is the discrepancy of P_n. When $\mathcal{H}$ is a Hilbert space, the discrepancy can be identified with the norm of some "worst-case" function in $\mathcal{H}$, and (1.4) is a form of Cauchy–Schwarz inequality. The right-hand side is the product of two terms: one depends only on f and the other depends only on P_n. Thus, if the function f that we want to integrate has bounded variation $V(f)$, for a sequence of point sets $\{P_n, n \ge 1\}$, the integration error converges to 0 at worst at the same rate as $D(P_n)$. If this rate beats $O(n^{-1/2})$, then we are doing asymptotically better than MC in two ways: the convergence is faster and we have a worst-case bound instead of just a confidence interval.

Important questions of interest are then: For a given Hilbert space $\mathcal{H}$ and a given $f \in \mathcal{H}$, how can we make sure that $V(f) < \infty$? Can we compute or bound this variation? How easily? Can we easily compute $D(P_n)$ for large point sets? Can we construct point sets P_n with small discrepancy $D(P_n)$? What is the best rate of convergence we can achieve for $D(P_n)$ when $n \to \infty$? Do we know how to construct explicit sequences of point sets $\{P_n, n \ge 1\}$ that achieve this optimal rate? What about the dependence on the dimension s? We shall partly address these questions.

A classic instance of (1.4), often used to justify QMC, is the *Koksma–Hlawka inequality* [79]

$$|\bar{\mu}_n - \mu| \le D^*(P_n) V_{\mathrm{HK}}(f), \tag{1.5}$$

where $V_{\mathrm{HK}}(f)$ denotes the variation of f in the sense of Hardy and Krause (see [90] for the definition and a discussion), and $D^*(P_n)$ is the *star discrepancy*, defined as follows. For each point $\mathbf{u} \in (0,1)^s$, we consider the absolute difference between the volume of the box $[\mathbf{0}, \mathbf{u})$ with corners at the origin and at $\mathbf{u}$, and the fraction of P_n that fall in that box, and let $D^*(P_n)$ be the supremum of this quantity over all $\mathbf{u} \in (0,1)^s$.

There are explicit constructions of sequences $\mathbf{u}_0, \mathbf{u}_1, \ldots$, in any dimension s, for which $D^*(P_n) = O(n^{-1}(\ln n)^s)$ when P_n contains the first n points of the sequence. Thus, for a function f with $V_{\mathrm{HK}}(f) < \infty$, a QMC approximation based on such a sequence gives a worst-case error bound in (1.5) that converges as $O(n^{-1}(\ln n)^s)$. From the practical viewpoint, however, this classical theory is deceptive. First, as soon as the dimension s exceeds 7 or 8, to have $n^{-1}(\ln n)^s < n^{-1/2}$ for all $n \geq n_0$, we need an n_0 that is much too large to be practical. For $s = 8$, for example, we already need $n_0 \approx 1.79 \times 10^{29}$. Second, the bound in (1.5) is typically much too hard to compute to be useful for error assessment. Third, whenever f is unbounded or has a discontinuity not aligned with the faces of the unit hypercube, $V_{\mathrm{HK}}(f)$ is infinite [90]. This is quite common in financial applications. For example, the payoff of a standard call option under the Black–Scholes model is unbounded. In the case of an average over multiple assets or multiple observation times (such as for a Bermudan–Asian option), it also has a kink not aligned with the axes. Barrier options have a discontinuous payoff. Discontinuity also arises when estimating sensitivities (the Greeks) for options whose payoff is otherwise continuous but not everywhere differentiable [24].

In the last decade, researchers have turned their attention to variants of (1.4) based on other types of discrepancies and variations, for which $V(f) < \infty$ under less restrictive assumptions (in some cases) and for which $D(P_n)$ can be computed efficiently. Popular settings include Sobolev classes of functions for which the partial derivatives up to a given order are assumed to be square-integrable, and which are also defined as reproducing kernel Hilbert spaces. The square variation is defined as a weighted sum, over all subsets of coordinates, of the integrated squared partial derivatives over the subspace determined by these coordinates. A nice feature of this setting is that by a clever choice of weights, and for standard types of point set constructions, there are efficient algorithms for computing the corresponding discrepancy, existence results for point sets with discrepancy below a given value, and concrete methods for constructing such point sets. Moreover, for any constant $\alpha \geq 1/2$, by making strong enough assumptions on the smoothness of f (via the square-integrability of high-order partial derivatives), and by assuming that f is periodic with period 1 with respect to each coordinate, we can obtain a convergence rate of $O(n^{-\alpha+\delta})$ for any $\delta > 0$ [18, 79, 98].

Convergence of the worst-case error can be characterized from the complexity theory viewpoint as follows [17, 37, 101, 104]. For a given family of function spaces $\mathcal{H}_s$ indexed by their dimension s, for $s \geq 1$ and any $\epsilon \in (0,1)$, let $n(\epsilon, s)$ be the minimal number of points in a QMC integration rule such that

$$\sup_{f \in \mathcal{H}_s,\, \|f\| \leq 1} |\bar{\mu}_n - \mu| \leq \epsilon \sup_{f \in \mathcal{H}_s,\, \|f\| \leq 1} |\mu|.$$

The family is said to be *QMC-tractable* if $n(\epsilon, s)$ increases only polynomially fast in $1/\epsilon$ and s; that is, if there are non-negative constants C, p and q, such that $n(\epsilon, s) \leq C\epsilon^{-p}s^q$. If this holds with $q = 0$ (so the required number of points does not

depend on the dimension), then the family is called *strongly QMC-tractable*. For families of weighted Sobolev spaces, where the coordinates, or subsets of coordinates, have non-negative *weights* indicating their importance in the function f, necessary and sufficient conditions on the weights for tractability and strong tractability are given for example in [17, 18, 37, 100–102]. These conditions show that tractability occurs (roughly) if the integrand depends mostly on a small number of coordinates, or can be well approximated by a sum of low-dimensional functions. This is related to the notions of low effective dimension and variance ratio discussed in Sect. 1.4.

Instead of the worst-case error, one may also study the average integration error for a random function f in a given class $\mathcal{F}$. Under mild conditions on $\mathcal{F}$, one can construct a reproducing kernel Hilbert space $\mathcal{H}$ for which the average error over $\mathcal{F}$ equals the worst-case error over functions $f \in \mathcal{H}$ with variation $V(f) \le 1$. So average error and worst-case error are equivalent in some sense [34] (see Sect. 2.2).

These error bounds and convergence rates are nice, but they nevertheless have practical limitations. For typical applications in finance, $V(f)$ is too hard to compute, and may be infinite. Yet, despite the fact that the worst-case error bounds are not practical, QMC happens to give much more accurate estimates than MC, for certain types of integrands encountered in finance, sometimes in hundreds of dimensions or more [1, 9, 43, 57, 94]. Some explanation of this success will be given in the following. The integration error can be estimated by using randomized versions of QMC, as we now explain.

1.3 Randomized quasi-Monte Carlo

The difficulty to obtain reliable error estimates with QMC can be addressed by switching to *randomized QMC* (RQMC), which turns QMC into a variance-reduction technique [4, 64, 65, 86–88]. The idea is to randomize P_n so that

(a) it retains its high uniformity when taken as a set, and
(b) each individual point is a random vector with the uniform distribution over $(0, 1)^s$.

An *RQMC point set* is one that satisfies these two conditions. One simple randomization that satisfies these conditions, for an arbitrary point set P_n, is a *random shift modulo 1* [13, 64, 98]: Generate a *single* point $\mathbf{U}$ uniformly over $(0, 1)^s$ and add it to each point of P_n, modulo 1, coordinate-wise. Another one is a *random digital shift in base b* [63, 65, 73]: Generate again $\mathbf{U}$ uniformly over $(0, 1)^s$, expand each of its coordinates in base b, and add the digits, modulo b, to the corresponding digits of each point of P_n.

Let $\hat{\mu}_{n,\mathrm{rqmc}}$ denote the estimator (1.2) in which $\mathbf{U}_0, \ldots, \mathbf{U}_{n-1}$ are the n randomized points of an RQMC point set. Under these two conditions, it is easily seen that $\mathbb{E}[\hat{\mu}_{n,\mathrm{rqmc}}] = \mu$, and we hope to have the inequality

$$\sigma^2_{n,\mathrm{rqmc}} := n \operatorname{Var}[\hat{\mu}_{n,\mathrm{rqmc}}] = n\mathbb{E}\big[(\hat{\mu}_{n,\mathrm{rqmc}} - \mu)^2\big] < n \operatorname{Var}[\hat{\mu}_n] = \sigma^2. \tag{1.6}$$

Because the RQMC estimator is unbiased, this variance is the mean square error. The worst-case variance over a Banach space $\mathcal{H}$,

$$\sup_{V(f)\le 1} \operatorname{Var}[\hat{\mu}_{n,\mathrm{rqmc}}],$$

is the square of the so-called *random-case error* [33, 38]. The square *worst-case error*, on the other hand, is defined as [33, 38]

$$\mathbb{E}\Big[\sup_{V(f)\le 1}(\hat{\mu}_{n,\mathrm{rqmc}}-\mu)^2\Big].$$

In the situation where f is a random function with some probability measure over $\mathcal{H}$ such that $\mathbb{E}[V^2(f)]=1$, the square *average-case error* is $\mathbb{E}[\mathrm{Var}\,[\hat{\mu}_{n,\mathrm{rqmc}}]]$ [38], where $\mathbb{E}$ is with respect to both the randomization of P_n and the random function. For the worst-case error, we randomize the rule first, then a devil selects a worst-case function f for the realized randomization. For the random-case error, the devil first selects f, then we randomize the rule. The random-case error can be smaller and may converge faster than the worst-case error. In fact, for the Sobolev space of functions with square-integrable mixed partial derivatives up to order α, it is known that optimal RQMC rules give $O(n^{-\alpha+\delta})$ worst-case error and $O(n^{-\alpha-1/2+\delta})$ random-case error [33]. However, explicit RQMC rules that achieve this random-case rate are still unknown, except for $\alpha=1$, where scrambled nets achieve the optimal rate [86, 87].

To estimate $\sigma^2_{n,\mathrm{rqmc}}$ and compute a confidence interval on μ, we can apply m independent randomizations to P_n and compute $\bar{X}_m$ and $S^2_{x,m}$, the sample mean and sample variance of the m independent realizations $X_1,\ldots,X_m$ of $\hat{\mu}_{n,\mathrm{rqmc}}$. Then, $\mathbb{E}[\bar{X}_m]=\mu$ and $\mathbb{E}[S^2_{x,m}]=m\,\mathrm{Var}\,[\bar{X}_m]=\sigma^2_{n,\mathrm{rqmc}}/n$ [64, 65]. If we assume that X_j is approximately normally distributed, then $\sqrt{m}(\bar{X}_m-\mu)/S_{x,m}$ has approximately the Student distribution with $m-1$ degrees of freedom, and this can be used to compute a confidence interval on μ in a standard way.

How should we select m? For a fixed total number mn of evaluations of f, a larger m gives a more accurate variance estimator while a larger n usually provides a more accurate estimator of μ. If our main target is really to estimate μ, we should normally select m somewhere between 5 and 25. But if it is important to obtain a good variance estimator, for example to compare the efficiencies of RQMC and MC, it makes sense to take a larger m.

For certain Hilbert spaces of (smooth) functions f and specific classes of (uniform) point sets P_n, variance bounds and asymptotic variance expressions (as a function of n, for either the worst-case f or an average f in the Hilbert space) can be found in [17, 18, 27, 38, 87, 89, 113], for example. These variance bounds are typically of the form $\sigma^2_{n,\mathrm{rqmc}}=O(n^{-2\alpha+\delta})$ for some $\alpha>0$ and any $\delta>0$, often for $\alpha\ge 1/2$. Note that this gives $\mathrm{Var}\,[\bar{X}_m]=O(m^{-1}n^{-2\alpha+\delta})$. There are many interesting situations where $\sigma^2_{n,\mathrm{rqmc}}$ is bounded by an expression that converges to zero faster than the squared worst-case error of the corresponding deterministic QMC method. This gives another important justification for the randomization.

In (1.2) and (1.3), the n points are given equal weights, and one may ask if we could do better with unequal weights. For a fixed f and a fixed deterministic point set P_n, this is indeed generally the case. However, it is shown in [4] that for RQMC, under mild conditions satisfied by typical RQMC point sets, equal weights are optimal.

Two important classes of RQMC point sets are lattice rules with a random shift modulo 1, and digital nets with a random digital shift. For these two cases, the lattice or digital net structure of the point set is preserved by the shift, and explicit expressions for the variance are available in terms of the squared Fourier or Walsh

coefficients of f [64, 65]. Then, by considering classes of functions for which these coefficients satisfy appropriate convergence conditions, one can obtain an arbitrary convergence rate for the variance as a function of n [64, 65]. However, such conditions are difficult to verify in practice (except for special cases). The ANOVA decomposition discussed in the next subsection is a more aggregated decomposition that is often more convenient to work with.

Numerical illustrations where RQMC is really effective (empirically) are given in [7, 24, 43, 56, 64], and in Sect. 8.

1.4 Variance decomposition and effective dimension

In finance applications, integrands f typically have several dozen or even several hundred dimensions, and (1.4) does not seem practically useful. In fact, a little thinking immediately reveals that filling evenly (say) the 100-dimensional unit hypercube would require an excessive number of points: just to have one point near each corner, we already need $n = 2^{100}$ points! Then, how can we justify our hope for a substantial variance reduction in (1.6) when s is large?

The explanation is that in many cases, f can be well approximated by a sum of low-dimensional functions that depend only on a small number of coordinates of $\mathbf{u}$. In those cases, for QMC or RQMC to be effective, it suffices that these low-dimensional functions are integrated with small error. For example, if $s = 100$ and f can be well approximated by a sum of two-dimensional functions $f_{\mathfrak{u}}$, where each $\mathfrak{u}$ satisfies $\mathfrak{u} = \{i, j\} \subset \{1, \ldots, s\} =: \mathcal{S}$ and the corresponding $f_{\mathfrak{u}}$ depends only on the two coordinates $\{u_i, u_j\}$, then it suffices to construct P_n so that for each function $f_{\mathfrak{u}}$ in the approximation, the projection of P_n over the unit square in which $f_{\mathfrak{u}}$ is defined covers that square very evenly.

For RQMC, this argument can be made rigorous using a *functional ANOVA decomposition* of f, as follows [71, 88, 110]. If $\sigma^2 < \infty$, there is a unique decomposition of f of the form

$$f(\mathbf{u}) = \mu + \sum_{\mathfrak{u} \subseteq \mathcal{S},\, \mathfrak{u} \neq \phi} f_{\mathfrak{u}}(\mathbf{u}) \tag{1.7}$$

where each $f_{\mathfrak{u}} : (0,1)^s \to \mathbb{R}$ depends only on $\{u_i, i \in \mathfrak{u}\}$, the $f_{\mathfrak{u}}$ integrate to zero and are orthogonal, and the variance decomposes as $\sigma^2 = \sum_{\mathfrak{u} \subseteq \mathcal{S}} \sigma_{\mathfrak{u}}^2$ where $\sigma_{\mathfrak{u}}^2 = \mathrm{Var}[f_{\mathfrak{u}}(\mathbf{U})]$ for $\mathbf{U}$ uniformly distributed over $(0,1)^s$. The $f_{\mathfrak{u}}$ are defined recursively by $f_\phi = \mu$ (a constant function) and

$$f_{\mathfrak{u}}(\mathbf{u}) = \int_{(0,1)^{s-|\mathfrak{u}|}} f(\mathbf{u})\, d\mathbf{u}_{\bar{\mathfrak{u}}} - \sum_{\mathfrak{v} \subset \mathfrak{u}} f_{\mathfrak{v}}(\mathbf{u})$$

for $\phi \neq \mathfrak{u} \subseteq \mathcal{S}$, where the first integral is with respect to the coordinates of $\mathbf{u}$ whose indexes are not in $\mathfrak{u}$, denoted by $\mathbf{u}_{\bar{\mathfrak{u}}}$. For each set of coordinates $\mathfrak{u}$, let $P_n(\mathfrak{u})$ denote the projection of P_n over the subspace determined by $\mathfrak{u}$. For a given function f, if

$$\sum_{\mathfrak{u} \in \mathcal{J}} \sigma_{\mathfrak{u}}^2 \geq \rho \sigma^2 \tag{1.8}$$

for a class $\mathcal{J}$ of small subsets of $\mathcal{S}$ and some ρ close to 1, and if we can construct P_n so that the projections $P_n(\mathfrak{u})$ are highly uniform for all $\mathfrak{u} \in \mathcal{J}$, then the important variance terms $\sigma_{\mathfrak{u}}^2$ can be reduced significantly, thus reducing the overall variance.

If (1.8) holds for $\mathcal{J} = \{\mathfrak{u} : |\mathfrak{u}| \le d\}$, the set of all projections of dimension d or less, we say that f has *effective dimension d in proportion ρ in the superposition sense* [88]. If it holds for $\mathcal{J} = \{\mathfrak{u} \subseteq \{1, \ldots, d\}\}$, we say that f has effective dimension d in proportion ρ in the *truncation sense* [9]. The latter can sometimes be achieved by redefining the function f without changing the expectation μ, via a change of variables, in a way that the first few uniforms account for most of the variance in f [2, 9, 43, 56, 74, 112]. In other words, we change the way the uniforms are used to generate the estimator in the simulation. We shall give examples of this in Sect. 8. In the context of financial applications, this can be achieved for example by bridge sampling techniques, principal component analysis, generating the random numbers in a different order, and replacing certain random variables by their conditional expectations [1, 2, 24, 43, 57, 76].

The *truncation variance ratio of order d* is $\sum_{\mathfrak{u} \subseteq \{1,\ldots,d\}} \sigma_{\mathfrak{u}}^2 / \sigma^2$, and the *superposition variance ratio of order d* is $\sum_{\mathfrak{u}:|\mathfrak{u}| \le d} \sigma_{\mathfrak{u}}^2 / \sigma^2$. This is the fraction of variance explained by the first d coordinates, and by the d-dimensional (or less) projections, respectively.

Another decomposition used in this paper is the Fourier expansion of f, written as

$$f(\mathbf{u}) = \sum_{\mathbf{h} \in \mathbb{Z}^s} \hat{f}(\mathbf{h}) \exp(2\pi\iota\, \mathbf{h} \cdot \mathbf{u}), \tag{1.9}$$

with Fourier coefficients

$$\hat{f}(\mathbf{h}) = \int_{(0,1)^s} f(\mathbf{u}) \exp(-2\pi\iota\, \mathbf{h} \cdot \mathbf{u})\, d\mathbf{u},$$

where $\iota = \sqrt{-1}$. The Fourier expansion refines the ANOVA decomposition in the sense that if f has ANOVA expansion (1.7) and Fourier expansion (1.9), then the Fourier coefficient $\hat{f}_{\mathfrak{v}}(\mathbf{h})$ of $f_{\mathfrak{v}}$ is $\hat{f}(\mathbf{h})$ if $\mathfrak{u}(\mathbf{h}) = \mathfrak{v}$, and 0 otherwise, where $\mathfrak{u}(\mathbf{h}) = \{j : h_j \ne 0\}$ for $\mathbf{h} = (h_1, \ldots, h_s) \in \mathbb{Z}^s$.

QMC point sets are usually constructed by making sure that certain types of projections have good uniformity. One basic desirable property is that all projections contain n distinct points, the same number as in P_n. The point set is then called *fully projection-regular* [64, 98]. Rectangular grids do not satisfy this property. Constructions for which several projections are identical are also interesting, because this makes their analysis easier. We say that P_n is *dimension-stationary* [68] if whenever $1 \le i_1 < \cdots < i_\eta < s$ and $1 \le j \le s - i_\eta$, we have $P_n(\{i_1, \ldots, i_\eta\}) = P_n(\{i_1 + j, \ldots, i_\eta + j\})$. This means that $P_n(\mathfrak{u})$ depends only on the *spacings* between the indices in $\mathfrak{u}$.

1.5 Outline

The remainder of this paper is organized as follows. The next section explains the definition of discrepancies, variations, error bounds, and variance bounds, defined via

reproducing kernel Hilbert spaces. We see that QMC can provide much faster convergence (asymptotically) than MC, especially for smooth periodic functions. In Sects. 3 and 4, we summarize the main properties of the two principal types of point sets constructions: lattice rules and digital nets. Other types of constructions are briefly mentioned in Sect. 5. In Sect. 6, we outline some ways of transforming the function f to reduce its variability or to increase its truncation or superposition variance ratio of a given order. This includes changes of variables, bridge sampling, and principal component sampling (for Brownian processes). In Sect. 8, we provide numerical illustrations with option pricing models based on geometric Brownian motion and variance-gamma processes. The concluding section mentions other extensions not covered in the paper, and some topics worthy of further investigation.

2 Error and variance bounds via reproducing kernel Hilbert spaces

2.1 RKHS theory and discrepancies

We start with a symmetric and positive semi-definite function $K : [0,1]^{2s} \to \mathbb{R}$, called the *kernel*. Symmetric and positive semi-definite means that for any $\mathbf{v}_1, \ldots, \mathbf{v}_m \in [0,1]^s$ and $m \geq 0$, the $m \times m$ matrix whose (i,j) entry is $K(\mathbf{v}_i, \mathbf{v}_j)$ is symmetric and positive semi-definite. For each $\mathbf{u} \in [0,1]^s$, let $K_{\mathbf{u}} = K(\mathbf{u}, \cdot) : [0,1]^s \to \mathbb{R}$. Consider the space of functions

$$\mathcal{H}_{K,0} = \left\{ f = \sum_{i=1}^{m} a_i K_{\mathbf{v}_i} \ : \ \mathbf{v}_i \in [0,1]^s, \ a_i \in \mathbb{R}, \ m \geq 0 \right\},$$

together with the inner product

$$\langle f, g \rangle_K = \left\langle \sum_{i=1}^{m} a_i K_{\mathbf{v}_i}, \ \sum_{j=1}^{\ell} b_j K_{\mathbf{w}_j} \right\rangle_K = \sum_{i=1}^{m} \sum_{j=1}^{\ell} a_i b_j K(\mathbf{v}_i, \mathbf{w}_j),$$

and let $\mathcal{H}_K$ be $\mathcal{H}_{K,0}$ to which we have added all limits of Cauchy sequences in $\mathcal{H}_{K,0}$. This $\mathcal{H}_K$ is a *reproducing kernel Hilbert space* (RKHS) with kernel K [106]. The norm of a function f in this space is $\|f\|_K = \langle f, f \rangle_K^{1/2}$. The kernel has a reproducing property in the sense that $f(\mathbf{u}) = \langle f, K_{\mathbf{u}} \rangle_K$ for all $f \in \mathcal{H}_K$.

RKHS theory tells us that for any point set P_n, there is an explicit function $\xi \in \mathcal{H}_K$ representing the error functional

$$\xi(\mathbf{u}) = \langle \xi, K_{\mathbf{u}} \rangle + K = \mathrm{Err}(K_{\mathbf{u}}, P_n) = \frac{1}{2} \sum_{i=0}^{n-1} K(\mathbf{u}, \mathbf{u}_i) - \int_{[0,1)^s} K(\mathbf{u}, \mathbf{v}) \, d\mathbf{v},$$

where

$$\mathrm{Err}(f, P_n) = \frac{1}{n} \sum_{i=0}^{n-1} f(\mathbf{u}_i) - \mu(f)$$

is the QMC error for f with P_n. Moreover,

$$\big|\mathrm{Err}(f, P_n)\big| = \big|\langle \xi, f-\mu\rangle_K\big| \le \|\xi\|_K \cdot \|f-\mu\|_K = D(P_n)V(f), \tag{2.1}$$

where the inequality is just Cauchy–Schwarz's, $V(f) = \|f-\mu\|_K$ is the variation of f, $D(P_n)$ is the *discrepancy* of P_n, and both depend on K. See [29, 31] for further details. Thus, the discrepancy is the worst-case error over the class of functions $f \in \mathcal{H}_K$ for which $V(f) \le 1$. The (square) discrepancy can be written more explicitly as

$$\begin{aligned} D^2(P_n) &= \|\xi\|_K^2 = \mathrm{Err}^2(\xi, P_n) \\ &= \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} K(\mathbf{u}_i, \mathbf{u}_j) - \frac{2}{n}\sum_{i=0}^{n-1}\int_{[0,1]^s} K(\mathbf{u}_i, \mathbf{v})\, d\mathbf{v} \\ &\quad + \int_{[0,1]^{2s}} K(\mathbf{u}, \mathbf{v})\, d\mathbf{u}\, d\mathbf{v}, \end{aligned} \tag{2.2}$$

which can be computed in $O(n^2 s)$ time if we assume that K and its integral are available in constant time. The inequality (2.1) can be generalized further, for example by using Hölder's inequality instead of Cauchy–Schwarz's [31].

Specific RKHS constructions are examined, for example, in [16–18, 29, 31, 32, 34, 113].

2.2 Random function f

Suppose now that f is a random function in a space $\mathcal{G}$ of functions $f : [0,1]^s \to \mathbb{R}$ endowed with some probability measure, and that the kernel is defined as

$$K(\mathbf{u}, \mathbf{v}) = \mathrm{Cov}\big[f(\mathbf{u}), f(\mathbf{v})\big] = \mathbb{E}\big[f(\mathbf{u}) f(\mathbf{v})\big] - \mu^2(f), \tag{2.3}$$

where the expectation is with respect to the probability law of f and $\mu = \mu(f)$ depends on f. In this case, it can be shown that the expected square error equals the square discrepancy that corresponds to K; i.e., $\mathbb{E}[\mathrm{Err}^2(f, P_n)] = \mathrm{Err}^2(\xi, P_n) = D^2(P_n)$. That is, the root mean square error over $\mathcal{G}$ is the same as the worst-case error over $\{f \in \mathcal{H}_K : V(f) \le 1\}$, and both are equal to $D(P_n)$. As a special case, if f is generated from the Brownian sheet measure, then the corresponding discrepancy turns out to be the classical $\mathcal{L}_2$ star discrepancy in (2.14) below, without the weights [114].

2.3 Random points

In RQMC, we have a fixed f but P_n is randomized and we are interested in the variance $\mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}] = \mathbb{E}[\mathrm{Err}^2(f, P_n)]$, where the expectation is with respect to the randomization of P_n. We have

$$\mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}] \le \mathbb{E}\big[D^2(P_n)V^2(f)\big] = V^2(f) \cdot \mathbb{E}\big[D^2(P_n)\big],$$

where

$$\mathbb{E}\big[D^2(P_n)\big] = \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\mathbb{E}\big[K(\mathbf{U}_i,\mathbf{U}_j)\big] - \int_{[0,1]^{2s}} K(\mathbf{u},\mathbf{v})\,d\mathbf{u}\,d\mathbf{v}. \tag{2.4}$$

If both f and P_n are random, one simply takes the kernel (2.3) in (2.4) to define the discrepancy.

2.4 Korobov spaces

In one important class of RKHS, the kernel has the form

$$K(\mathbf{u},\mathbf{v}) = \sum_{\mathbf{h}\in\mathbb{Z}^s} w(\mathbf{h}) e^{2\pi\iota\mathbf{h}^{\mathsf{t}}(\mathbf{u}-\mathbf{v})} \tag{2.5}$$

where the $w(\mathbf{h})$ are non-negative weights such that $\sum_{\mathbf{h}\in\mathbb{Z}^s} w(\mathbf{h}) < \infty$. The corresponding inner product is

$$\langle f, g\rangle_K = \sum_{\mathbf{h}\in\mathbb{Z}^s} \big[w(\mathbf{h})\big]^{-1} \hat{f}(\mathbf{h})\hat{g}^*(\mathbf{h}),$$

where the $\hat{f}(\mathbf{h})$ are the Fourier coefficients of f and the $\hat{g}^*(\mathbf{h})$ are the complex conjugate Fourier coefficients of g. (When $w(\mathbf{h}) = 0$ we put $1/w(\mathbf{h}) = \infty$ and we use the convention that $0\times\infty = 0$.) This gives a Hilbert space of functions whose Fourier expansion converges absolutely, named a *Korobov space*. The corresponding square discrepancy and square variation are

$$D^2(P_n) = \frac{1}{n^2}\sum_{\mathbf{0}\neq\mathbf{h}\in\mathbb{Z}^s} w(\mathbf{h}) \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} e^{2\pi\iota\mathbf{h}^{\mathsf{t}}(\mathbf{u}_i-\mathbf{v}_j)} = \sum_{\mathfrak{u}\subseteq\mathcal{S}} D_{\mathfrak{u}}^2(P_n) \tag{2.6}$$

and

$$V^2(f) = \sum_{\mathbf{0}\neq\mathbf{h}\in\mathbb{Z}^s} \frac{|\hat{f}(\mathbf{h})|^2}{w(\mathbf{h})} = \sum_{\mathfrak{u}\subseteq\mathcal{S}} V^2(f_{\mathfrak{u}}), \tag{2.7}$$

where $D_{\mathfrak{u}}^2(P_n)$ is the sum of the terms in (2.6) for the vectors $\mathbf{h}$ whose set of non-zero coordinates is exactly $\mathfrak{u}$. That is, the (square) discrepancy and variation are decomposable in accordance with the ANOVA decomposition. Note that

$$D_{\mathfrak{u}}^2(P_n) \neq D^2(P_n(\mathfrak{u})) = \sum_{\mathfrak{v}\subseteq\mathfrak{u}} D_{\mathfrak{v}}^2(P_n).$$

If $\xi(\mathfrak{u})$ represents the error functional with $P_n(\mathfrak{u})$, then $D^2(P_n(\mathfrak{u})) = \|\xi(\mathfrak{u})\|_K^2$ and $D_{\mathfrak{u}}^2(P_n) = \|\xi_{\mathfrak{u}}\|_K^2$ for each $\mathfrak{u}$, where $\xi_{\mathfrak{u}}$ is defined by the ANOVA decomposition of ξ.

A key issue now is the choice of weights. Different choices give rise to various discrepancies [18, 31]. In general, the weights may depend on s, even if this is not explicit in our notation. The weights must be selected so that the discrepancy is not

too hard to compute, low-discrepancy point sets can be constructed, and the corresponding variation $V(f)$ is under control for the functions f of interest in our applications. In view of (2.7), this means that appropriate weights should depend on the behavior of the squared Fourier coefficients of f.

Unfortunately, we rarely know how the Fourier coefficients behave in applications, so there is no simple and definitive way of selecting the weights. It is more convenient if $V(f)$ is expressed in terms of quantities that are easier to compute or to bound. In certain applications, one can bound the partial derivatives of f, for example, and this has motivated choices of weights for which $V(f)$ can be written in terms of those partial derivatives [17, 29, 97, 98, 101].

In [18], the weights $w(\mathbf{h})$ are assumed to have the form

$$w(\mathbf{h}) = w(s,\mathbf{h}) = \gamma_{s,\mathfrak{u}(\mathbf{h})} \prod_{j\in\mathfrak{u}(\mathbf{h})} |h_j|^{-2\alpha} = \gamma_{s,\mathfrak{u}(\mathbf{h})} \prod_{j=1}^{s} \min\bigl(1, |h_j|^{-2\alpha}\bigr)$$

for $\alpha > 1/2$, where $w(\mathbf{0}) = 1$. These authors provide necessary and sufficient conditions on the weights $\gamma_{s,\mathfrak{u}(\mathbf{h})}$ for tractability and strong tractability, and prove the existence of lattice rules for which the worst-case error converges as $O(n^{-\alpha+\delta})$ for any $\delta > 0$, where the hidden constant generally depends on α (and perhaps polynomially on s, unless we have strong tractability). Convergence rate results of this type have been known for a long time for fixed s [79, 98] and also for special choices of the weights [17, 101]. However, there are no explicit constructions available for those lattice rules; they must be found by computer searches for each n and s. When α is an integer, the corresponding kernel can be written as

$$K_\alpha(\mathbf{u},\mathbf{v}) = 1 + \sum_{\phi\neq\mathfrak{u}\subseteq\mathcal{S}} \gamma_{s,\mathfrak{u}} \left[\frac{-(-4\pi^2)^\alpha}{(2\alpha)!}\right]^{|\mathfrak{u}|} \prod_{j\in\mathfrak{u}} B_{2\alpha}\bigl((u_j - v_j) \bmod 1\bigr) \tag{2.8}$$

where $B_{2\alpha}$ is the Bernoulli polynomial of degree 2α (see [98] for the definition; in particular, $B_1(u) = u - 1/2$ and $B_2(u) = u^2 - u + 1/6$), and the square variation is

$$V^2(f) = \sum_{\phi\neq\mathfrak{u}\subseteq\mathcal{S}} \gamma_{s,\mathfrak{u}}^{-1} \bigl(4\pi^2\bigr)^{-\alpha|\mathfrak{u}|} \int_{[0,1]^{|\mathfrak{u}|}} \left| \int_{[0,1]^{s-|\mathfrak{u}|}} \frac{\partial^{\alpha|\mathfrak{u}|} f}{\partial \mathbf{u}_{\mathfrak{u}}^{\alpha}}(\mathbf{u})\, d\mathbf{u}_{\bar{\mathfrak{u}}} \right|^2 d\mathbf{u}_{\mathfrak{u}}, \tag{2.9}$$

where $\mathbf{u}_{\mathfrak{u}}$ represents the coordinates of $\mathbf{u}$ whose indices are in $\mathfrak{u}$ and $\mathbf{u}_{\bar{\mathfrak{u}}}$ represents those whose indices are not in $\mathfrak{u}$. The corresponding RKHS is comprised of periodic functions f of period 1 with respect to each coordinate, because the Bernoulli polynomials of even degree have this property. Note that the kernel (2.8) and the square variation (2.9) are already decomposed in the same way as the ANOVA decomposition of f.

With these general weights, the kernel and the discrepancy are too difficult to compute in general. The sum in (2.5) is infinite, and the number of terms in (2.8) grows exponentially with s. For this reason, more restricted classes of weights, for which simplified and more easily computable expressions for the kernel are available, have been examined in the literature. In particular, the weights are said to have the *product*

form if $\gamma_{s,\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_{s,j}$ for some nonnegative constants $\gamma_{s,1}, \ldots, \gamma_{s,s}$, which may depend on s. This gives

$$w(\mathbf{h}) = \prod_{j\in\mathfrak{u}(\mathbf{h})} \gamma_{s,j} |h_j|^{-2\alpha}. \tag{2.10}$$

Then, the RKHS $\mathcal{H}_K$ is a tensor product of s one-dimensional Hilbert spaces, and the kernel becomes

$$K_\alpha(\mathbf{u}, \mathbf{v}) = \prod_{j=1}^{s} \left[1 - \gamma_j \frac{(-4\pi^2)^\alpha}{(2\alpha)!} B_{2\alpha}\big((u_j - v_j) \bmod 1\big)\right], \tag{2.11}$$

which can be computed in $O(s)$ time, so the discrepancy can be computed in $O(n^2 s)$ time. Thus, things simplify nicely from the computational viewpoint. For product weights, we have strong tractability if and only if $\sup_{s\geq 1}(\gamma_{s,1} + \cdots + \gamma_{s,s}) < \infty$, and tractability if and only if $\sup_{s\geq 1} \sum_{j=1}^{s} \ln(1 + \gamma_{s,1})/\ln(s+1) < \infty$ [18]. That is, the weights $\gamma_{s,j}$ must decrease fast enough with j or s. Note that these conditions do not guarantee fast convergence in n. Under the stronger condition that $\sup_{s\geq 1} \sum_{j=1}^{s} \sqrt{\gamma_{s,j}} < \infty$, there exist point sets P_n such that $D(P_n) = O(n^{-1+\delta})$ uniformly in s.

The weights are said to be *order-dependent* if $\gamma_{s,\mathfrak{u}}$ depends only on the cardinality of $\mathfrak{u}$ (and perhaps on s). They are *finite-order* weights if there is an integer q independent of s such that $\gamma_{s,\mathfrak{u}} = 0$ whenever $|\mathfrak{u}| > q$. Finite-order weights can be appropriate for function spaces of low effective dimension in the superposition sense: If the effective dimension does not exceed q, then we can disregard the quality of the projections of P_n over the subspaces of dimension larger than q, by putting their weights $\gamma_{s,\mathfrak{u}}$ to zero. There are important applications in financial derivative pricing where the effective dimension does not exceed 2 or 3 in proportion 0.99 or more [110–113], and for which finite-order weights make sense. For order-dependent and finite-order weights, if the weights are bounded, tractability holds and lattice rules can be constructed that yield an $O(n^{-\alpha+\delta})$ discrepancy [18, 96, 100] for smooth functions.

2.5 Periodic smooth functions

For periodic smooth functions f with period 1 with respect to each coordinate, and whose partial derivatives up to some (integer) order $\alpha \geq 1$ are all square integrable, we can achieve an $O(n^{-\alpha+\delta})$ convergence of the worst-case error in a RKHS strongly related to that obtained from (2.8), and defined as follows [29, 31]. We select a periodic function $g : [0, 1] \to \mathbb{R}$, with $g(0) = g(1)$, whose derivative $g^{(\alpha)}$ of order α is essentially bounded over $[0, 1]$, and with $\int_0^1 g^{(\nu)}(u)\, du = 0$ for $\nu = 0, 1, \ldots, \alpha$. Let $M = \int_0^1 [g^{(\alpha)}(u)]^2\, du$ and define the kernel

$$K(\mathbf{u}, \mathbf{v}) = 1 + \sum_{\phi\neq\mathfrak{u}\subseteq\mathcal{S}} \gamma_{s,\mathfrak{u}} \prod_{j\in\mathfrak{u}} \left[M + g(u_j) + g(v_j) - \frac{(-1)^\alpha}{(2\alpha)!} B_{2\alpha}\big((u_j - v_j) \bmod 1\big)\right]. \tag{2.12}$$

The corresponding discrepancy can be written easily via (2.2).

If $g(u) = 0$, the kernel (2.12) is equivalent to the Korobov kernel (2.8) (the factor $4\pi^2$ can be incorporated in the weights $\gamma_{s,\mathfrak{u}}$).

As another special case, if we take $\alpha = 1$ and $g(u) = -B_2(u)/2$, we obtain a weighted $\mathcal{L}_2$-*unanchored discrepancy* which can be interpreted as follows. For each subset $\mathfrak{u}$ of coordinates and $\mathbf{u}, \mathbf{v} \in [0,1]^{|\mathfrak{u}|}$, let $D(P_n(\mathfrak{u}), \mathbf{u}, \mathbf{v})$ be the *local discrepancy* for the $|\mathfrak{u}|$-dimensional box $[\mathbf{u}, \mathbf{v})$ (with opposite corners at $\mathbf{u}$ and $\mathbf{v}$), defined as the absolute difference between the volume of this box and the fraction of the points that fall in it. The square weighted $\mathcal{L}_2$-unanchored discrepancy can be written as

$$\big[D_2(P_n)\big]^2 = \sum_{\phi \neq \mathfrak{u} \subseteq \mathcal{S}} \gamma_{s,\mathfrak{u}} \int_{[0,1]^{|\mathfrak{u}|}} \int_{[\mathbf{0},\mathbf{v}]} D^2\big(P_n(\mathfrak{u}), \mathbf{u}, \mathbf{v}\big)\, d\mathbf{u}\, d\mathbf{v}.$$

2.6 Spaces of non-periodic smooth functions

A class of discrepancies for smooth functions (not necessarily periodic) can be defined by selecting an arbitrary function $g : [0,1] \to \mathbb{R}$, whose first derivative g' is essentially bounded over $[0,1]$, and for which $\int_0^1 g(u)\, du = 0$. Let $M = \int_0^1 [g'(u)]^2\, du$ and define the kernel

$$K(\mathbf{u}, \mathbf{v}) = 1 + \sum_{\phi \neq \mathfrak{u} \subseteq \mathcal{S}} \gamma_{s,\mathfrak{u}} \prod_{j \in \mathfrak{u}} \big[M + g(u_j) + g(v_j) + (1/2) B_2\big((u_j - v_j) \bmod 1\big) + B_1(u_j) B_1(v_j)\big]. \tag{2.13}$$

As a special case, by taking $g(u) = 1/6 - u^2/2$, the product in (2.13) becomes $\prod_{j \in \mathfrak{u}} \min[1 - u_j, 1 - v_j]$, and this kernel gives a weighted $\mathcal{L}_2$-*star discrepancy* whose square can be written as

$$\big[D_2^*(P_n)\big]^2 = \sum_{\phi \neq \mathfrak{u} \subseteq \mathcal{S}} \gamma_{s,\mathfrak{u}} \int_{[0,1]^{|\mathfrak{u}|}} D^2\big(P_n(\mathfrak{u}), \mathbf{0}, \mathbf{u}\big)\, d\mathbf{u}. \tag{2.14}$$

By taking $g(u) = 0$ for all u, we obtain a discrepancy for which the corresponding square variation is

$$V^2(f) = \sum_{\phi \neq \mathfrak{u} \subseteq \mathcal{S}} \gamma_{s,\mathfrak{u}}^{-1} \int_{[0,1]^{|\mathfrak{u}|}} \left| \frac{\partial^{|\mathfrak{u}|}}{\partial \mathbf{u}_{\mathfrak{u}}} f_{\mathfrak{u}}(\mathbf{u}_{\mathfrak{u}}) \right|^2 d\mathbf{u}_{\mathfrak{u}}.$$

For these two choices of g, regardless of the weights, it is known how to construct infinite sequences for which $D_2^*(P_n)$ converges as $O(n^{-1+\delta})$ where the hidden constant generally depends on s, and may increase exponentially with s. For product weights, strong tractability holds if and only if $\sup_{s \geq 1} \sum_{j=1}^{s} \gamma_{s,j} < \infty$ [17]. Under the stronger condition $\sup_{s \geq 1} \sum_{j=1}^{s} \gamma_{s,j}^{1/2} < \infty$, it is known how to make concrete constructions that achieve the rate $O(n^{-1+\delta})$ with a hidden constant that does not depend on s [49].

2.7 Hilbert spaces based on the Walsh expansion

In Sect. 2.4, we constructed the RKHS based on a Fourier expansion of f. Similar constructions can be made for expansions with respect to other types of orthogonal bases. There are many possibilities, yielding a large collection of discrepancies. One of them, the Walsh expansion, is interesting because it goes along very nicely with the digital net constructions.

Select a prime integer base $b \geq 2$. Let $\mathbb{N}_0 = \{0, 1, \dots\}$. For $\mathbf{h} = (h_1, \dots, h_s) \in \mathbb{N}_0^s$ and $\mathbf{u} = (u_1, \dots, u_s) \in [0, 1)^s$, where

$$h_j = \sum_{i=0}^{\ell_j - 1} h_{j,i} b^i, \qquad u_j = \sum_{\ell \geq 1} u_{j,\ell} b^{-\ell} \in [0, 1),$$

the digits $h_{j,i}$ and $u_{j,\ell}$ are in $\mathbb{Z}_b$, and $u_{j,\ell} \neq b - 1$ for infinitely many ℓ, define

$$\langle \mathbf{h}, \mathbf{u} \rangle = \sum_{j=1}^{s} \sum_{\ell=0}^{\ell_j - 1} h_{j,\ell} u_{j,\ell+1} \bmod b. \tag{2.15}$$

The *Walsh expansion* in base b of $f : [0, 1)^s \to \mathbb{R}$ is

$$f(\mathbf{u}) = \sum_{\mathbf{h} \in \mathbb{N}_0^s} \tilde{f}(\mathbf{h}) e^{2\pi \iota \langle \mathbf{h}, \mathbf{u} \rangle / b},$$

with Walsh coefficients

$$\tilde{f}(\mathbf{h}) = \int_{[0,1)^s} f(\mathbf{u}) e^{-2\pi \iota \langle \mathbf{h}, \mathbf{u} \rangle / b} \, d\mathbf{u}.$$

In analogy with the Korobov spaces, we can adopt the kernel

$$K(\mathbf{u}, \mathbf{v}) = \sum_{\mathbf{h} \in \mathbb{N}_0^s} w(\mathbf{h}) e^{2\pi \iota \langle \mathbf{h}, (\mathbf{u} \ominus_b \mathbf{v}) \rangle / b}$$

for some weights $w(\mathbf{h})$, where $\ominus_b$ denotes the digit-wise subtraction modulo b. If $\sum_{\mathbf{h} \in \mathbb{N}_0^s} w(\mathbf{h}) < \infty$, this gives a Hilbert space of functions f whose Walsh expansion converges absolutely. The square discrepancy is

$$D^2(P_n) = \sum_{\mathrm{u} \subseteq \mathcal{S}} D_{\mathrm{u}}^2(P_n) = \frac{1}{n^2} \sum_{\mathbf{0} \neq \mathbf{h} \in \mathbb{N}_0^s} w(\mathbf{h}) \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} e^{2\pi \iota \langle \mathbf{h}, (\mathbf{u}_i \ominus_b \mathbf{u}_j) \rangle / b} \tag{2.16}$$

and the square variation of f is

$$V^2(f) = \sum_{\mathrm{u} \subseteq \mathcal{S}} V^2(f_{\mathrm{u}}) = \sum_{\mathbf{0} \neq \mathbf{h} \in \mathbb{N}_0^s} \frac{|\tilde{f}(\mathbf{h})|^2}{w(\mathbf{h})}.$$

Again, the choice of weights is a key issue. We can consider similar types of weights as for the Korobov spaces, and obtain similar discrepancies and error bounds. For

product weights, the discrepancy simplifies as in (2.11), for example. These spaces are convenient for the analysis of digital nets. We shall return to this in Sect. 4.6.

2.8 Random shifts for RQMC

For RQMC, in the case of the random shift modulo 1, $\mathbb{E}[D^2(P_n)]$ is obtained with $\mathbb{E}[K(\mathbf{U}_i, \mathbf{U}_j)] = \mathbb{E}[K(\mathbf{u}_i + \mathbf{U}, \mathbf{u}_j + \mathbf{U})] = \mathbb{E}[K(\mathbf{u}_i - \mathbf{u}_j + \mathbf{U}, \mathbf{U})] =: K_{\mathrm{sh}}(\mathbf{u}_i, \mathbf{u}_j)$, which depends only on the difference $(\mathbf{u}_i - \mathbf{u}_j) \bmod 1$ (it is shift-invariant) [30, 31]. But any such shift-invariant kernel has a Fourier expansion of the form

$$K_{\mathrm{sh}}(\mathbf{u}, \mathbf{v}) = \sum_{\mathbf{h} \in \mathbb{Z}^s} w(\mathbf{h}) e^{2\pi \iota \mathbf{h}^{\mathrm{t}}(\mathbf{u}-\mathbf{v})}$$

with non-negative Fourier coefficients $\hat{K}_{\mathrm{sh}}(\mathbf{h}) = w(\mathbf{h})$ (because the kernel is non-negative definite). We recover the Korobov kernel (2.5). The corresponding mean square discrepancy can then be written as

$$\mathbb{E}\big[D^2(P_n)\big] = \frac{1}{n} \sum_{\mathbf{0} \neq \mathbf{h} \in \mathbb{Z}^s} w(\mathbf{h}) \sum_{i=0}^{n-1} e^{2\pi \iota \mathbf{h}^{\mathrm{t}} \mathbf{u}_i}. \tag{2.17}$$

With product weights as in (2.10), we end up with the kernel (2.11).

For a random digital shift in base b, a similar expression can be obtained, but with the Fourier expansion replaced by a Walsh expansion in base b. In that case, we have $\mathbb{E}[K(\mathbf{U}_i, \mathbf{U}_j)] = \mathbb{E}[K(\mathbf{u}_i \ominus_b \mathbf{u}_j \oplus_b \mathbf{U}, \mathbf{U})] =: K_{\mathrm{dsh}}(\mathbf{u}_i, \mathbf{u}_j)$, which depends only on $\mathbf{u}_i \ominus_b \mathbf{u}_j$, where $\oplus_b$ denotes the digit-wise addition modulo b. Thus, this kernel has a Walsh expansion of the form

$$K_{\mathrm{dsh}}(\mathbf{u}, \mathbf{v}) = \sum_{\mathbf{h} \in \mathbb{N}_0^s} w(\mathbf{h}) e^{2\pi \iota \langle \mathbf{h}, \mathbf{u} \ominus_b \mathbf{v} \rangle / b}$$

with non-negative Walsh coefficients $\tilde{K}_{\mathrm{dsh}}(\mathbf{h}) = w(\mathbf{h})$. The corresponding mean square discrepancy can be written as

$$\mathbb{E}\big[D^2(P_n)\big] = \frac{1}{n} \sum_{\mathbf{0} \neq \mathbf{h} \in \mathbb{N}_0^s} w(\mathbf{h}) \sum_{i=0}^{n-1} e^{2\pi \iota \langle \mathbf{h}, \mathbf{u}_i \rangle / b}. \tag{2.18}$$

2.9 Periodizing the function

We saw that better convergence rates can be achieved in spaces of periodic smooth functions than for non-periodic functions. It seems like a good idea, then, to look for changes of variables that can periodize a smooth non-periodic function [32, 98]. For convenience, this is usually achieved by applying a one-dimensional change of variable one coordinate at a time (more general changes of variables are discussed in Sect. 6). A general class of such transformations change the integrand $f(u_1, \ldots, u_s)$ into $f(\varphi(u_1), \ldots, \varphi(u_s))|\varphi'(u_1) \cdots \varphi'(u_s)|$, for some appropriate smooth one-to-one transformation $\varphi : [0, 1] \to [0, 1]$, such that $\varphi^{(\ell)}(0) = \varphi^{(\ell)}(1) = 0$ for $\ell = 1, \ldots, \alpha$. The new function has the same integral over $[0, 1]^s$ as the original one. Specific

choices of φ proposed in the literature include polynomial and trigonometric functions whose degrees or frequencies increase with α [7, 40, 98]. A major problem with this type of transformation is that while making the function periodic, it can also increase its variation $V(f)$. In particular, we have to be careful that $|\varphi'(u)|$ does not become too large.

A variant of this is to select a continuous transformation $\varphi : [0, 1] \to [0, 1]$ (not necessarily one-to-one) with the property that $\int_a^b \varphi(u)\, du = b - a$ for every interval $[a, b] \subseteq [0, 1]$, and $\varphi(0) = \varphi(1)$. Then,

$$\int_{[0,1]^s} f(\mathbf{u})\, d\mathbf{u} = \int_{[0,1]^s} f\big(\varphi(\mathbf{u})\big)\, d\mathbf{u},$$

where $\varphi(\mathbf{u}) = (\varphi(u_1), \ldots, \varphi(u_s))$. Note that this can be conveniently implemented by transforming the points P_n by applying φ to each coordinate, and keeping f unchanged (so the simulation program that computes f needs no change). This means in particular that if we are in a RKHS with kernel K, then the discrepancy of the transformed points is obtained simply by replacing $K(\mathbf{u}, \mathbf{v})$ by $K(\varphi(\mathbf{u}), \varphi(\mathbf{v}))$ in the discrepancy formula [32]. This applies to either deterministic or randomized points. When the points are randomized (RQMC), this transformation must be applied *after* the randomization.

Perhaps the simplest such transformation takes $\varphi(u) = 2u$ for $u \le 1/2$ and $\varphi(u) = 2(1 - u)$ for $u > 1/2$. It stretches each coordinate of each point $\mathbf{u}_i$ by a factor of two, then folds back the coordinates that become larger than 1. It is known as the *baker's transformation*. Equivalently, this transformation can be visualized as contracting the graph of f (for any given coordinate) horizontally by a factor of two, so the function is now defined over the interval $[0, 1/2]$ only, and then making a mirror copy over the interval $[1/2, 1]$, so the transformed function is now symmetric with respect to $1/2$, and its periodic continuation of period 1 is continuous.

By replacing the kernel $K(\mathbf{u}, \mathbf{v})$ with $K(\varphi(\mathbf{u}), \varphi(\mathbf{v}))$ in the appropriate discrepancy expressions, Hickernell [32] obtains explicit discrepancy expressions for an arbitrary point set randomized by a random shift followed by a baker's transformation, for a Hilbert space of (non-periodic) functions with square integrable partial derivatives of order 2. He also provides a simplified expression for the case of a randomly shifted lattice, and uses it to show the existence of lattice rules for which this discrepancy is $O(n^{-2+\delta})$ for fixed s. In other words, adding the baker's transformation to the random shift reduces the variance from $O(n^{-2+\delta})$ to $O(n^{-4+\delta})$ for non-periodic smooth functions. A similar result applies to a digital net with a random digital shift [14]. Empirical results showing significant variance reductions provided by the baker's transformation can be found in [57, 62], for example.

A natural question to ask at this point is: To what extent do the integrands encountered in finance fit the function spaces examined here? The truth is that these integrands rarely belong to the smooth Sobolev spaces of functions with square-integrable partial derivatives of high order. In fact, these integrands are often non-differentiable at some points. Nevertheless, RQMC with point sets constructed along the lines discussed in this paper often provides significant variance reduction (and efficiency improvement), as illustrated in Sect. 8. The links between this success and the convergence results discussed here have yet to be clarified. It seems that the main explanation of RQMC effectiveness in finance is the fact that integrands are often

of very low effective dimension in the superposition sense, and can often be transformed to have low effective dimension in the truncation sense. The important low-dimensional functions of the ANOVA decomposition are often non-differentiable on a lower-dimensional manifold (e.g., at one point in the one-dimensional case), but are smooth elsewhere, and RQMC appears to work well for these types of functions.

3 Lattice rules

3.1 Definition

An *integration lattice* is a vector space of the form

$$L_s = \left\{ \mathbf{v} = \sum_{j=1}^{s} h_j \mathbf{v}_j \text{ such that each } h_j \in \mathbb{Z} \right\},$$

where $\mathbf{v}_1, \ldots, \mathbf{v}_s \in \mathbb{R}^s$ are linearly independent over $\mathbb{R}$ and where L_s contains $\mathbb{Z}^s$, the set of integer vectors. The QMC approximation of μ with $P_n = L_s \cap [0,1)^s$ is a *lattice rule* [98]. The matrix $\mathbf{V}$ whose *rows* are the basis vectors $\mathbf{v}_1^t, \ldots, \mathbf{v}_s^t$ is a *generator matrix* of L_s. The columns of its inverse $\mathbf{W} = \mathbf{V}^{-1}$ form a basis for the *dual lattice*, defined as

$$L_s^* = \left\{ \mathbf{h} \in \mathbb{R}^s : \mathbf{h}^t \mathbf{v} \in \mathbb{Z} \text{ for all } \mathbf{v} \in L_s \right\}.$$

It turns out that L_s contains $\mathbb{Z}^s$ if and only if all entries of $\mathbf{W}$ are integers. In this case, one has $n = \det(\mathbf{W})$ (the determinant of $\mathbf{W}$) and all coordinates of all points of P_n are multiples of $1/n$. Each projection of L_s over a subset $\mathfrak{u}$ of coordinates is also an integration lattice $L(\mathfrak{u})$, with dual $L^*(\mathfrak{u})$, and which determines a lower-dimensional lattice rule based on the point set $P_n(\mathfrak{u})$.

The *rank* of L_s is the smallest r such that one can find a basis of the form $\mathbf{v}_1, \ldots, \mathbf{v}_r, \mathbf{e}_{r+1}, \ldots, \mathbf{e}_s$, where $\mathbf{e}_j$ is the jth unit vector in s dimensions. Most lattice rules used in practice are of rank 1 [64]; their corresponding point set can be written as

$$\begin{aligned} P_n &= \left\{ \mathbf{v} = i\mathbf{v}_1 \bmod 1,\ i = 0, \ldots, n-1 \right\} \\ &= \left\{ (i\mathbf{a}_1 \bmod n)/n,\ i = 0, \ldots, n-1 \right\}, \end{aligned}$$

where $\mathbf{a}_1 = (a_1, \ldots, a_s)$ and $\mathbf{v}_1 = \mathbf{a}_1/n$. The set P_n is fully projection-regular if and only if $r = 1$ and $\gcd(a_j, n) = 1$ for each j, and in that case there is no loss of generality in assuming that $a_1 = 1$.

A *Korobov* rule is a lattice rule of rank 1 for which

$$\mathbf{a}_1 = \left(1, a, a^2 \bmod n, \ldots, a^{s-1} \bmod n\right)$$

for some $a \in \mathbb{Z}_n$. Its point set can be written as

$$\begin{aligned} P_n = \big\{ &(x_0/n, \ldots, x_{s-1}/n) \text{ such that } x_0 \in \mathbb{Z}_n \text{ and} \\ &x_j = a x_{j-1} \bmod n \text{ for all } j > 0 \big\}. \end{aligned}$$

This is the set of all vectors of s successive values produced by a linear congruential generator (LCG) with modulus n and multiplier a, from all possible initial states (including 0) [64]. The equivalent recurrence $x_j/n = u_j = au_{j-1} \bmod 1$ offers a convenient way to enumerate the points, especially when this recurrence has full period $n-1$, i.e., when n is prime and a is a primitive element modulo n. For the ith point, we can start with $u_1 = i/n$ and apply the recurrence to obtain the successive coordinates $u_2, \ldots, u_s$. Korobov point sets are actually infinite-dimensional, because an unlimited number of coordinates are defined by this recurrence.

Sequence of embedded lattices $L_s^{(1)} \subset L_s^{(2)} \subset L_s^{(3)} \subset \cdots$, constructed so that each lattice contains the previous one, have been studied in [11, 13, 35, 47]. They permit one to increase the cardinality of P_n sequentially, until a given accuracy has been achieved, for example. A simple (and practical) case where $L_s^{(j)}$ is a Korobov lattice with $n = n^{(j)} = 2^j$ and multiplier $a = a^{(j)} = a^{(j+1)} \bmod 2^j$ is proposed in [35]. An infinite sequence can be defined simply by specifying an infinite sequence of multipliers that satisfy this condition.

3.2 Randomization, discrepancies, and parameters selection

Lattice point sets are usually randomized simply by a *random shift modulo 1*, as proposed in [13]. The lattice structure of the points is preserved by the shift, in the sense that we have a shifted lattice. L'Ecuyer and Lemieux [64] have shown that with a randomly shifted lattice rule, whenever $\sigma^2 < \infty$, the variance of the RQMC estimator is exactly

$$\mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}] = \sum_{\mathbf{0} \neq \mathbf{h} \in L_s^*} \left|\hat{f}(\mathbf{h})\right|^2, \tag{3.1}$$

where the $\hat{f}(\mathbf{h})$ are the Fourier coefficients of f. The same expression was obtained earlier by Tuffin [105] under stronger conditions. Given that the goal is to minimize the variance, this expression tells us that for a given f, the most relevant discrepancy is exactly the expression (3.1). This suggests a general class of discrepancies (or figures of merit) for lattice rules, of the form [28, 64, 65]

$$\mathcal{M}_w(P_n) = \sum_{\mathbf{0} \neq \mathbf{h} \in L_s^*} w(\mathbf{h}), \tag{3.2}$$

where the weights $w(\mathbf{h})$ try to mimic the anticipated behavior of $|\hat{f}(\mathbf{h})|^2$. If $\mathcal{F}(w, c)$ is the class of functions f whose squared Fourier coefficients satisfy $|\hat{f}(\mathbf{h})|^2 \leq c\, w(\mathbf{h})$ for all $\mathbf{h} \in \mathbb{Z}^s$, for some constant c, then we have the variance bound $\mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}] \leq c\, \mathcal{M}_w(P_n)$ for any integrand $f \in \mathcal{F}(w, c)$. With no randomization, the integration error $\bar{\mu}_n - \mu$ is also given by (3.1), but with the square removed, and under the restrictive condition that the Fourier expansion converges absolutely. We also had the latter condition for the Korobov spaces earlier. It is important to underline that the variance expression (3.1) holds under a much weaker condition than that. Only the sum of *squared* Fourier coefficients needs to converge. The function f does not have to be bounded, for example. Owen [91] studies QMC for unbounded integrands from another viewpoint.

In the case of an integration lattice, it turns out that $\sum_{i=0}^{n-1} e^{2\pi\iota \mathbf{h}^t \mathbf{u}_i} = n$ if $\mathbf{h} \in L_s^*$, and 0 otherwise [98, Lemma 2.7], and therefore the criterion (3.2) is exactly equivalent to both (2.6) and (2.17). We are back to the same weight selection problem. The square discrepancy for the kernel (2.11) simplifies to

$$D^2(P_n) = -1 + \frac{1}{n} \sum_{i=0}^{n-1} \prod_{j=1}^{s} \left(1 + \gamma_j \frac{(-4\pi^2)^\alpha}{(2\alpha)!} B_{2\alpha}(u_j)/2\right), \tag{3.3}$$

which can be computed in time $O(ns)$, and is also a weighted version of a criterion known as $P_{2\alpha}$ [29]. (Several authors name it P_α [98]; their α corresponds to our 2α. The weights used in [29] are $\tilde{\beta}_j^2 = 4\pi^2 \gamma_j^{1/\alpha}$.) The unweighted case of this criterion, where $\gamma_j = 1$ for all j, with $\alpha = 1$, was examined long ago [98] and specific parameters were proposed. Good lattices for more general weights can be found in [83, 84, 99], for example. For the same Hilbert space as in (3.3), a discrepancy that takes into account the random shift modulo 1 followed by a baker's transformation, computable in $O(ns)$ time, is given in [32], (16).

There has been strong interest recently in a technique called component-by-component (CBC) construction, for rank-1 lattices [11, 17, 49, 50, 83, 96, 99]. The idea, for a given n, is to select the components a_j of the vector $\mathbf{a}_1 = (a_1, \ldots, a_s)$ iteratively: Start with $a_1 = 1$, and at step j, with the previously selected components $a_1, \ldots, a_{j-1}$ fixed, select a_j to optimize a given figure of merit (discrepancy) for the j-dimensional lattice with generating vector $(a_1, \ldots, a_j)$. This greedy technique drastically reduces the number of possibilities that need to be examined for $\mathbf{a}_1$. A remarkable result is that by restricting the search in this way, one still obtains lattice rules that achieve the same theoretical rate of convergence for the discrepancy as for the best provable bounds. This has been established for several Hilbert spaces of functions by various authors. Fast computational algorithms have also been designed to speed up this search [11, 83, 84]. These CBC algorithms can compute the vector $\mathbf{a}_1$ in $O(n \log(n) s)$ time using $O(n)$ memory for product weights, and in $O(n[\log(n)s + s^2])$ time using $O(ns)$ memory for order-dependent weights. They are fast enough to allow on-request (just-in-time) construction of lattice rules for parameters s, n, and weights that are relevant for a given application and can be provided by the user, e.g., from a simulation program.

Most of the (square) discrepancies mentioned so far can be written as sums over the projections $\mathfrak{u}$ and/or over the points $\mathbf{u}_i$ or the pairs $(\mathbf{u}_i, \mathbf{u}_j)$. Several other types of discrepancies proposed in the literature are written as a supremum over similar terms (or a minimum, with the terms inverted). The idea is that in view of (3.1), we may want to use a criterion of the form

$$\mathcal{M}'_w(P_n) = \sup_{\mathbf{0} \neq \mathbf{h} \in L_s^*} w(\mathbf{h}) \tag{3.4}$$

or equivalently

$$1/\mathcal{M}'_w(P_n) = \min_{\mathbf{0} \neq \mathbf{h} \in L_s^*} 1/w(\mathbf{h}) \tag{3.5}$$

instead of (3.2). Two figures of merit of the form (3.5), proposed long ago, are the Zaremba index, where $1/w(\mathbf{h}) = \prod_{j\in\mathfrak{u}(\mathbf{h})} |h_j|$, and the trigonometric degree (plus 1), where $1/w(\mathbf{h}) = \sum_{j\in\mathfrak{u}(\mathbf{h})} |h_j|$ [12, 98]. The second one represents the minimal number of hyperplanes to cover all the points of P_n, minus 1 in some cases [48]. If we take $1/w^2(\mathbf{h}) = \sum_{j\in\mathfrak{u}(\mathbf{h})} h_j^2$ instead, then the criterion (3.4) corresponds to the spectral test, used to assess the quality of random number generators [28, 48, 55], and which represents the largest distance between equidistant parallel hyperplanes that cover all the points. Note that all these criteria measure the length of the shortest non-zero vector $\mathbf{h}$ in the dual lattice, using a different notion of length. Other definitions of lengths could be used as well. In each case, all the functions f whose Fourier expansion can be written only in terms of vectors $\mathbf{h}$ whose length is smaller than the criterion are integrated exactly (with zero error, or zero variance in the case of a randomly shifted rule) by the lattice rule.

These criteria based on worst-case length can be generalized by weighting the lengths in some way, for example as a function of $\mathfrak{u}(\mathbf{h})$. This is often done by using a theoretical upper bound $w_d^*(n)$ on the length of the shortest non-zero vector in the dual lattice for an integration lattice of density n in d dimensions. The length $1/w(\mathbf{h})$ is divided by $w^*_{|\mathfrak{u}(\mathbf{h})|}(n)$ to provide a standardized number between 0 and 1 [54, 55, 64, 68]. More generally, we might divide $1/w(\mathbf{h})$ by $\tilde{w}_{\mathfrak{u}(\mathbf{h})}(n)$, for some numbers $\tilde{w}_{\mathfrak{u}}(n) \geq w^*_{|\mathfrak{u}|}(n)$, to reduce the weights of the projections (or coordinate subsets) deemed less important. When s is large, to speed up the computations, we may restrict ourselves to a subclass $\mathcal{J}$ of the sets of indices $\mathfrak{u} \subseteq \mathcal{S}$, and put $w(\mathbf{h}) = 0$ whenever $\mathfrak{u}(\mathbf{h}) \notin \mathcal{J}$. This gives a criterion of the general form

$$\min_{\mathfrak{u}\in\mathcal{J}} \min_{\mathbf{0}\neq\mathbf{h}\in L_s^*;\, \mathfrak{u}(\mathbf{h})=\mathfrak{u}} \left[w(\mathbf{h})\tilde{w}_{\mathfrak{u}}(n)\right]^{-1}. \tag{3.6}$$

This type of criterion, with $1/w(\mathbf{h})$ taken as the Euclidean norm of $\mathbf{h}$, has been proposed in [64, 68] and used to compute tables of parameters for Korobov rules [64]. Simplified versions of it have been used for a long time to measure the quality of random number generators [48, 54, 58]. One advantage of this type of criterion is that for certain choices of length (including the Euclidean one), its computation time is pretty much independent of n, so it can be convenient for large values of n. On the other hand, its computing time is generally exponential in $\tilde{d} = \sup\{|\mathfrak{u}| : \mathfrak{u} \in \mathcal{J}\}$, and also linear in $|\mathcal{J}|$. Values of $\tilde{d}$ of up to a few dozen can nevertheless be handled [54, 55, 64].

With all these potential selection criteria for lattice rules, one would certainly wonder which one should be used in practice for typical finance applications, and what is the difference of RQMC variance between the rules selected via different criteria. There is still no clear and complete answer to these questions. Partial results, for simplified models, can be found in [108, 110–113].

4 Digital nets and sequences

4.1 Definition and constructions

The following digital method, introduced by Niederreiter [77, 79], provides the second main class of construction methods of low-discrepancy point sets and sequences. Let $b \geq 2$ be an arbitrary integer, usually a prime, called the *base*. A net of $n = b^k$ points in s dimensions is defined by selecting s *generator matrices* $\mathbf{C}_1, \ldots, \mathbf{C}_s$, where each $\mathbf{C}_j$ is (in theory) an $\infty \times k$ matrix with elements in $\mathbb{Z}_b = \{0, \ldots, b-1\}$. The matrix $\mathbf{C}_j$ determines the coordinate j of all the points. To define the ith point $\mathbf{u}_i$, for $i = 0, \ldots, b^k - 1$, we write the digital expansion of i in base b and multiply the vector of its digits by $\mathbf{C}_j$, modulo b, to obtain the digits of the expansion of $u_{i,j}$, the jth coordinate of $\mathbf{u}_i$. That is,

$$i = \sum_{\ell=0}^{k-1} a_{i,\ell} b^{\ell},$$

$$\begin{pmatrix} u_{i,j,1} \\ u_{i,j,2} \\ \vdots \end{pmatrix} = \mathbf{C}_j \begin{pmatrix} a_{i,0} \\ a_{i,1} \\ \vdots \\ a_{i,k-1} \end{pmatrix} \mod b, \tag{4.1}$$

$$u_{i,j} = \sum_{\ell=1}^{\infty} u_{i,j,\ell} b^{-\ell}, \quad \text{and} \quad \mathbf{u}_i = (u_{i,1}, \ldots, u_{i,s}). \tag{4.2}$$

The resulting point set is a *digital net in base b*. In practice, the expansion (4.2) is truncated to r digits for some r, so each $\mathbf{C}_j$ is an $r \times k$ matrix. Typically, r is equal to k, or is slightly larger, or is selected so that b^r is near 2^{31}. We assume that the first k rows of each $\mathbf{C}_j$ form a non-singular $k \times k$ matrix; in this case, each one-dimensional projection $P_n(\{j\})$ of P_n over the jth coordinate, truncated to the first k digits in base b, is equal to the set $\mathbb{Z}_n/n = \{0, 1/n, \ldots, (n-1)/n\}$. However, these numbers are enumerated in a different order for the different coordinates. In other words, the first k rows of $\mathbf{C}_j$ implement a permutation of $\mathbb{Z}_n/n$. The choice of these permutations is the key factor for the uniformity of P_n and of its projections $P_n(\mathfrak{u})$. If all the permutations were the same, all the points would fall along the main diagonal of the unit hypercube. More generally, each projection $P_n(\mathfrak{u})$ of a digital net (or sequence) is also a digital net (or sequence), with generator matrices $\mathbf{C}_{j_1}, \ldots, \mathbf{C}_{j_d}$ if $\mathfrak{u} = \{i_1, \ldots, i_d\}$.

What we just gave is a somewhat simplified definition of a digital net. It covers the most popular constructions. The setting of [79] is more general; one can apply bijections (or permutations) to the digits of $\mathbb{Z}_b$ before and after the multiplication by $\mathbf{C}_j$. This is done by taking an arbitrary ring R of cardinality b, and defining bijections $\psi_\ell : \mathbb{Z}_b \to R$ for $\ell = 0, \ldots, k-1$, and $\eta_{j,\ell} : R \to \mathbb{Z}_b$ for $\ell = 1, \ldots, r$ and $j = 1, \ldots, s$. In (4.1), each digit $a_{i,\ell}$ is replaced by $\psi_\ell(a_{i,\ell})$, and the multiplications

by $\mathbf{C}_j$ are done in the ring R. Then, in (4.2), each $u_{i,j,\ell}$ is replaced by $\eta_{j,\ell}(u_{i,j,\ell})$. These bijections give additional opportunity for improving the uniformity. If $R = \mathbb{Z}_b$, they are equivalent to permuting the digits of $\mathbb{Z}_b$. If b is a power of a prime, then R can be taken as a finite field, so the multiplications by $\mathbf{C}_j$ are performed in the finite field (allowing this was the original reason for introducing the bijections).

If each $\mathbf{C}_j$ has an infinite number of columns, then we have an infinite sequence of points, called a *digital sequence in base b*. The first k columns determine the first b^k points, for any k. Well-known examples are the sequences of Sobol' [103] in base 2, of Faure [20] in prime base b, of Niederreiter [77], and of Niederreiter and Xing [82]. With an infinite sequence of matrices $\mathbf{C}_j$, we have an *infinite-dimensional digital net*. These infinite sequences of columns and matrices are typically defined via recurrences (each column and matrix being a function of the previous ones).

When n is fixed, we can enumerate the points in any order, so one (simple) possibility is to use the identity permutation for the first k digits of the first coordinate. That is, the points are enumerated by their first coordinate. To do that, the first k rows of $\mathbf{C}_1$ must form the reflected identity matrix, with a 1 in row $k - c + 1$ of each column c and zeroes everywhere else. If the first k rows of $\mathbf{C}_j$ form the identity instead, then the corresponding output (looking at the first k digits only) is the first n elements of the *van der Corput sequence* in base b, defined as $\psi_b(0), \psi_b(1), \psi_b(2), \ldots$, where $\psi_b : \mathbb{N} \to [0, 1)$ is the *radical inverse* function in base b, i.e.,

$$\psi_b(i) = a_0 b^{-1} + a_1 b^{-2} + \cdots + a_{k-1} b^{-k}$$

if i is a k-digit integer in base b which has the digital b-ary expansion $i = a_0 + a_1 b + \cdots + a_{k-1} b^{k-1}$. The first n elements of a van der Corput sequence fill up the unit interval quite uniformly for any large enough n. The uniformity is better when n is a power of b. When n is not fixed in advance, for example if we add points until we think the estimate is sufficiently accurate, then we need a digital sequence and the best constructions typically take $\mathbf{C}_1$ (truncated to its first k rows) equal to the identity.

4.2 Measures of uniformity via equidistribution

For a vector $\mathbf{q} = (q_1, \ldots, q_s)$ with non-negative integer coordinates, and a base $b \geq 2$, if we partition the jth axis into b^{q_j} equal parts for each j, we obtain a partition of $[0, 1)^s$ into $b^{q_1 + \cdots + q_s}$ rectangular boxes of the same size and shape. We call it a $\mathbf{q}$-*equidissection in base b*. A point set P_n of cardinality $n = b^k$ (usually a digital net in base b) is $\mathbf{q}$-*equidistributed in base b* if each box of this equidissection contains the same number of points of P_n, i.e., exactly b^t points where $t = k - q_1 - \cdots - q_s$. For a digital net in base b, this property is easy to verify: It holds if and only if the set of $k - q = q_1 + \cdots + q_t$ rows that comprise the first q_j rows of $\mathbf{C}_j$, for $j = 1, \ldots, s$, is linearly independent in the finite ring R (regardless of the bijections ψ_ℓ and $\eta_{j,\ell}$).

The point set P_n is called a (t, k, s)-*net in base b* if it is $(q_1, \ldots, q_s)$-equidistributed whenever $q_1 + \cdots + q_s \leq k - t$ [79]. We call the smallest such t the t-*value* of the net. A digital sequence $\{\mathbf{u}_0, \mathbf{u}_1, \ldots\}$ in s dimensions is a (t, s)-*sequence in base b* if for all integers $k > 0$ and $\nu \geq 0$, the point set $Q(k, \nu) = \{\mathbf{u}_i : i = \nu b^k, \ldots, (\nu + 1)b^k - 1\}$ is a (t, k, s)-net in base b. The t-value is the most widely used figure of merit for

digital nets. Its justification is that for a fixed s, for (t,k,s)-nets or (t,s)-sequences with bounded t-value, the star discrepancy $D^*(P_n)$ converges as $O(n^{-1}(\log n)^{s-1})$.

Ideally, we should like the t-value to be zero, but there are theoretical bounds on the best that can be achieved. In particular, a $(0,k,s)$-net in base b can exist only if $b \geq s-1$, and a $(0,t)$-sequence in base b can exist only if $b \geq t$. Lower bounds for general pairs (b,s), together with the best values achieved by known constructions, are tabulated in [95]. As an illustration, for $b=2$, $s=20$, and $k=16$, so $n=2^{16}$, we know that the t-value cannot get below 9. Reaching this optimal value only guarantees equidistribution when there are at most $2^7=128$ boxes that contain at least $2^9=512$ points each.

The difficulty is that a small t-value would require equidistribution for a very rich family of partitions into rectangular boxes, and this becomes impossible when t is too small. This explains the large lower bounds on the t-value. One alternative is to consider a smaller family of partitions; for example, only cubic boxes [53, 63, 65]. The largest ℓ for which P_n is $(\ell,\ldots,\ell)$-equidistributed is called the *s-dimensional resolution* of P_n. This value cannot exceed $\lfloor k/s \rfloor$ and we call the difference $\delta = \lfloor k/s \rfloor - \ell$ the *resolution gap* of P_n.

These definitions also apply to the projections $P_n(\mathrm{u})$ of P_n, for $\mathrm{u} = \{i_1,\ldots,i_d\} \subset \mathcal{S}$. Let t_u and δ_u denote the t-value and the resolution gap associated with $P_n(\mathrm{u})$, and $t^*_{|\mathrm{u}|}$ the lower bound on t_u. Simple measures of non-uniformity for digital nets can be defined by [56, 65, 69, 92]

$$\max_{\mathrm{u}\in\mathcal{J}} \gamma_\mathrm{u}\delta_\mathrm{u}, \quad \text{or} \quad \sum_{\mathrm{u}\in\mathcal{J}} \gamma_\mathrm{u}\delta_\mathrm{u}, \quad \text{or}$$

$$\max_{\mathrm{u}\in\mathcal{J}} \gamma_\mathrm{u}\left[t_\mathrm{u} - t^*_{|\mathrm{u}|}\right], \quad \text{or} \quad \sum_{\mathrm{u}\in\mathcal{J}} \gamma_\mathrm{u}\left[t_\mathrm{u} - t^*_{|\mathrm{u}|}\right],$$

for some non-negative weights γ_u, where $\mathcal{J}$ is a preselected class of index sets u. The choice of $\mathcal{J}$ and of the weights γ_u is a matter of compromise. If $\mathcal{J}$ contains too many sets, the selection criterion is more costly to compute, and its best possible value is larger, which means that the criterion is generally less demanding for the important projections. The weights are sometimes taken all equal to 1.

4.3 Classical constructions

The oldest and most popular type of digital sequence, introduced by Sobol' [103], is in base 2 and uses upper triangular binary matrices $\mathbf{C}_j$ with 1 s on the diagonal. These matrices have an infinite number of rows and columns. In each column, the bits above the diagonal are taken from the binary expansion of some real number which Sobol' calls *direction number*. These direction numbers obey a bitwise recurrence across columns. Their choice determines the quality of the net. The original values proposed by Sobol' were selected to provide $(1,\ldots,1)$- and $(2,\ldots,2)$-equidistribution only (i.e., by considering only the first two bits of each coordinate). In particular, no attention was paid to the quality of the projections $P_n(\mathrm{u})$ in two or more dimensions, and the uniformity of these projections often turns out to be quite bad [75]. Different direction numbers, based on stronger equidistribution properties, are proposed

in [46, 67]. Generator matrices can be defined easily for arbitrary additional coordinates without modifying the previous ones, so the dimension is virtually infinite.

Faure [20] proposed digital sequences with generator matrices $\mathbf{C}_j = \mathbf{P}^{j-1} \bmod b$, where the base b is prime and $\mathbf{P}$ is a $k \times k$ upper triangular matrix whose entry (ℓ, c) is the number of ways of choosing $\ell - 1$ elements among $c - 1$ for $\ell \le c$ and is 0 for $\ell > c$. This gives $\mathbf{C}_1 = \mathbf{I}$ (the identity) and $\mathbf{C}_j = \mathbf{P}\mathbf{C}_{j-1} \bmod b$. Faure [20] proved that if b is prime and $b \ge s$, this sequence is a $(0, s)$-sequence in base b. Unfortunately, the condition $b \ge s$ is a practical limitation when the dimension s is large. Moreover, since the choice of b depends on s, the dimension must be fixed a priori.

Niederreiter [77] and Niederreiter and Xing [82] proposed sequences and nets for arbitrary prime power bases and nets in base 2 with better t-value than those of Sobol'. The sequence of [77] can be viewed as infinite-dimensional, but not that of [82].

For all these sequences, if we fix the number of points $n = b^k$, we can take $\mathbf{C}_1$ equal to the reflected identity, so the first coordinate of point i is i/n, and we move up all other coordinates by one position. In a sense, we save one coordinate. From a (t, s)-sequence, we can then obtain a $(t, k, s + 1)$-net for any k.

Other types of digital net constructions can be found in [5, 19, 63, 69, 80, 92, 95] and the references given there.

4.4 Polynomial lattice rules

Polynomial integration lattices are a special case of digital nets. They are similar to the ordinary integration lattices, except that they are defined in different spaces. Let b be an arbitrary integer larger than 1, $\mathbb{Z}_b$ the residue ring of integers modulo b (the base), $\mathbb{Z}_b[z]$ the ring of polynomials with coefficients in $\mathbb{Z}_b$, and $\mathbb{L}_b$ the ring of formal Laurent series with coefficients in $\mathbb{Z}_b$, of the form $\sum_{\ell=\omega}^{\infty} x_\ell z^{-\ell}$, where $x_\ell \in \mathbb{Z}_b$. A *polynomial integration lattice* is defined as

$$\mathcal{L}_s = \left\{ \mathbf{v}(z) = \sum_{j=1}^{s} q_j(z)\mathbf{v}_j(z) \text{ such that each } q_j(z) \in \mathbb{Z}_b[z] \right\},$$

where $\mathbf{v}_1(z), \ldots, \mathbf{v}_s(z) \in \mathbb{L}_b^s$ are defined by $\mathbf{v}_j(\mathbf{z}) = \mathbf{a}_j(z)/P(z)$, where $P(z) = z^k + \alpha_1 z^{k-1} + \cdots + \alpha_k \in \mathbb{Z}_b[z]$ and each $\mathbf{a}_j(z)$ is a vector of polynomials of degree less than k. Note that $(\mathbb{Z}_b[z])^s \subseteq \mathcal{L}_s$.

An output mapping $\varphi : \mathbb{L}_b \to \mathbb{R}$ is defined by

$$\varphi\left(\sum_{\ell=\omega}^{\infty} x_\ell z^{-\ell}\right) = \sum_{\ell=\omega}^{\infty} x_\ell b^{-\ell}.$$

The *polynomial lattice rule* uses the node set $P_n = \varphi(\mathcal{L}_s) \cap [0, 1)^s = \varphi(\mathcal{L}_s \cap \mathbb{L}_{b,0})$, where $\mathbb{L}_{b,0} = \mathbb{L}_b \bmod \mathbb{Z}_b[z]$. Polynomial lattice rules of rank 1 were introduced in [78]. They were generalized to rules of arbitrary rank over a finite field in [66, 69] and over the ring $\mathbb{Z}_b$ in [56]. Most of the properties of ordinary lattice rules have counterparts for the polynomial rules [56]. In particular, figures of merit similar to (3.4), (3.5), and (3.6) can be defined in terms of shortest vectors in the dual lattices, and CBC constructions can provide good parameters for discrepancies based on the Walsh expansion, of the general form (2.16), with product weights [14–16].

4.5 Randomizations and scramblings

It was pointed out in [63, 64, 66] that for a polynomial integration lattice, a random shift of $\mathcal{L}_s$ in the space of formal series is equivalent to a *random digital shift in base b*, already introduced in Sect. 1.3 and appropriate for digital nets in general: Generate $\mathbf{U} = (U_1, \ldots, U_s)$ uniformly over $(0, 1)^s$, write the digital expansion in base b of each of its coordinates, say $U_j = \sum_{\ell=1}^{\infty} d_{j,\ell} b^{-\ell}$, then add $d_{j,\ell}$ modulo b to the ℓth digit of the digital expansion in base b of the jth coordinate of each $\mathbf{u}_i \in P_n$. For $b = 2$, the digit-wise addition modulo b is a bitwise exclusive-or; it can be computed very quickly on a computer. For a digital net, this digital shift, and all the scrambles that we discuss below, preserve the $\mathbf{q}$-equidistribution properties for all vectors $\mathbf{q}$, as well as the (t, k, s)-net properties. That is, if P_n is $\mathbf{q}$-equidistributed before the shift or the scramble, it remains so after it, regardless of its realization.

The random digital shift provides an unbiased estimator of μ with a small amount of work. However, a deeper scrambling (more randomization) sometimes gives more variance reduction because the average point set over the class in which we randomize can have better uniformity in the larger class (with more randomization) than in the restricted class (with the random shift only).

Owen [85] proposed a *nested uniform scrambling*, for digital nets, which randomly permutes the values $\{0, \ldots, b-1\}$ used for the digits $u_{i,j,\ell}$, independently across the coordinates and across the digits. He showed in [87] that for functions whose mixed partial derivatives satisfy a Lipschitz condition, with a (t, k, s)-net scrambled in this way, the RQMC estimator has a variance of $O(n^{-3}(\log n)^s)$. Unfortunately, this method requires $(1 + b + \cdots + b^{\ell-1})s$ independent permutations to scramble the first ℓ digits, so it is very time-consuming. For $b = 2$ and $\ell = 20$, for example, we would need more than one million permutations for each coordinate. A simplified (and faster) implementation is proposed in [23]. In practice, one can take $\ell = k - 1$ and then apply an independent random digital shift to the remaining bits $u_{i,j,k}, u_{i,j,k+1}, \ldots$ for each (i, j). This is equivalent to the full scrambling.

A class of less expensive approaches that perform well, for digital nets, are the *linear scrambles* [21, 39, 73, 89], which multiply each matrix $\mathbf{C}_j$ by a random invertible matrix $\mathbf{M}_j$, modulo b. Usually, $\mathbf{M}_j$ is a $w \times w$ lower triangular matrix, with invertible elements modulo b on its diagonal, and it multiplies $\mathbf{C}_j$ on the left. Each $\mathbf{M}_j$ can be generated at random in some class, or can be constructed to minimize some measure of discrepancy. In a version proposed in [39, 73], the diagonal entries are generated uniformly over $\{1, \ldots, b-1\}$, and the entries below the diagonal are generated uniformly over $\{0, \ldots, b-1\}$, all independently. A linear scramble alone does not provide an unbiased estimator, but its combination with a random digital shift modulo b does. The *striped matrix scramble* proposed by [89] adds the constraint that in any given column, all entries below the diagonal are equal to the diagonal entry, which is generated randomly over $\{1, \ldots, b-1\}$. In base 2, all entries on or below the diagonal are equal to 1. With this scramble, the points enjoy global and local antithetic properties in each dimension, and yields $O(n^{-4})$ variance for smooth one-dimensional integrands (and therefore for the smooth one-dimensional components of the ANOVA decomposition).

4.6 Variance and error analysis via Walsh expansions

Digital nets have a *dual space*

$$\mathcal{C}_s^* = \left\{\mathbf{h} \in \mathbb{N}_0^s : \langle \mathbf{h}, \mathbf{u}\rangle = 0 \text{ for all points } \mathbf{u} \in P_n\right\}$$

that plays a similar role as the dual lattice L_s^* for integration lattices [56, 65, 66, 69, 81]. This space is closed with respect to digital addition in base b and multiplication by an integer modulo b, so it is actually a lattice, even though the digital net itself is not necessarily the intersection of a lattice with the unit hypercube [56]. The dual space $\mathcal{C}_s^*$ indicates which digits of the points are equidistributed, in the following sense. The non-zero digits $h_{j,\ell}$ of any given $\mathbf{h} \in \mathbb{N}_0^s$ form a bit mask that selects a set of digits $u_{j,\ell+1}$ of the points via (2.15). The vector of these selected digits is equidistributed, i.e., each possibility appears the same number of times for this vector, if and only if $\mathbf{h}$ is *not* in $\mathcal{C}_s^*$. In other words, each vector $\mathbf{h}$ corresponds to a given partition of $[0,1)^s$, and the point set is equidistributed for a given class of such partitions if and only if no $\mathbf{h}$ from that class belongs to $\mathcal{C}_s^*$. Uniformity criteria can be defined in terms of the length of a shortest non-zero vector in $\mathcal{C}_s^*$, for some definition of length on the vectors $\mathbf{h}$ [56, 69, 81]. The resolution gap and the t-value, for example, can be expressed in this way.

Recently, Dick [16] introduced a generalization of the t-value, named the $t(\alpha)$-value, based on a length of $\mathbf{h}$ defined as the sum of the α most significant non-zero digits of each coordinate of $\mathbf{h}$, for a given integer $\alpha \geq 1$. This definition of length is a way to select which vectors $\mathbf{h}$ should be kept out of $\mathcal{C}_s^*$. He constructs explicit digital sequences with bounded $t(\alpha)$-value, and proves that for such sequences, for the Sobolev space of functions with square-integrable mixed partial derivatives up to order α, the worst-case error converges as $O(n^{-\alpha+\delta})$. With a random digital shift, the root mean square error converges at the same rate. Dick's remarkable result provides the first explicit construction that achieves this optimal rate for this Sobolev class.

It is proved in [65, 66] that whenever $\sigma^2 < \infty$, the variance of the RQMC estimator of $\mu(f)$ for a digital net with a random digital shift modulo b is exactly

$$\mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}] = \sum_{\mathbf{0}\neq\mathbf{h}\in\mathcal{C}_s^*} \left|\tilde{f}(\mathbf{h})\right|^2, \tag{4.3}$$

where the $\tilde{f}(\mathbf{h})$ are the coefficients of the Walsh expansion of f. This expression is the ultimate discrepancy measure for a given f, for this RQMC scheme.

As for lattice rules, this suggests a general class of discrepancies of the form [65]

$$\mathcal{M}_w(P_n) = \sum_{\mathbf{0}\neq\mathbf{h}\in\mathcal{C}_s^*} w(\mathbf{h}), \tag{4.4}$$

with weights $w(\mathbf{h})$ that should match the behavior of $|\tilde{f}(\mathbf{h})|^2$. If $\mathcal{F}(w,c)$ is the class of functions f whose squared Walsh coefficients satisfy $|\tilde{f}(\mathbf{h})|^2 \leq c\, w(\mathbf{h})$ for all $\mathbf{h} \in \mathbb{N}_0^s$, then $\mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}] \leq c\, \mathcal{M}_w(P_n)$ for any $f \in \mathcal{F}(w,c)$. Without the randomization, the integration error $\bar{\mu}_n - \mu$ is also given by (4.3), but with the square removed, and under the restrictive condition that the Walsh expansion converges absolutely.

In the case of a digital net, $\sum_{i=0}^{n-1} e^{2\pi\iota\langle \mathbf{h},\mathbf{u}_i\rangle/b} = n$ if $\mathbf{h} \in C_s^*$, and 0 otherwise, and then (4.4) is equivalent to both (2.16) and (2.18).

Much of what applies to lattice rules has a counterpart for digital nets (or at least to their polynomial lattice subclass) [15, 16, 56, 65, 69].

5 Other types of constructions

5.1 Hammersley point sets and Halton sequence

Digital nets and sequences can be further generalized by allowing different bases for the different coordinates, say b_j for coordinate j. For example, the point sets introduced long ago by Hammersley [26] have

$$\mathbf{u}_i = \big(i/n, \psi_{b_1}(i), \psi_{b_2}(i), \ldots, \psi_{b_{s-1}}(i)\big),$$

for $i = 0, \ldots, n-1$, where the basis b_j used for coordinate j is the jth smallest prime number. Here, $\mathbf{C}_0$ is the reflected identity and $\mathbf{C}_j$ is the identity for all $j > 0$. The corresponding infinite sequence, proposed by Halton [25], takes

$$\mathbf{u}_i = \big(\psi_{b_1}(i), \psi_{b_2}(i), \ldots, \psi_{b_s}(i)\big)$$

for all $i \geq 0$, where b_j is again the jth smallest prime. One drawback is that b_j becomes quite large for large j. In any case, the identity matrices $\mathbf{C}_j$ could also be replaced by more general generating matrices, which may give room to improve the uniformity.

In [109], the Halton sequence is randomized simply by selecting the starting point $\mathbf{u}_0$ randomly over $(0, 1)^s$, truncating its coordinates to a finite digital expansion, and exploiting the fact that there is a simple way of getting $\psi_b(i+1)$ directly from $\psi_b(i)$, so the successive points can be generated without knowing their indices i in the original sequence. They show that this method satisfies conditions (a) and (b) of Sect. 1.3. In their numerical experiments, it performs much better than randomly shifting (modulo 1) the Halton sequence.

5.2 Recurrence-based point sets

Infinite-dimensional point sets can be defined in a similar way as random number generators by selecting a state space $\mathcal{S} = \{s_0, \ldots, s_{n-1}\}$ of cardinality n, a transition function $\varphi : \mathcal{S} \to \mathcal{S}$, and an output function $g : \mathcal{S} \to [0, 1)$. The n points $P_n = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\}$ are defined by $\mathbf{u}_i = (g(s_i), g(\varphi(s_i)), g(\varphi^2(s_i)), \ldots)$. Often, φ defines a recurrence of period $n-1$ (it visits all the states except one, which is usually $s_0 = 0$, for which $\varphi(s_0) = s_0$). Then the points are easy to enumerate with very little storage; one can just run the recurrence over its full cycle and take all overlapping vectors of successive values produced by that recurrence, plus the additional vector whose coordinates are all $g(0)$. One example of this is to use a small linear congruential generator with prime modulus n and primitive multiplier a; this is equivalent to a Korobov lattice rule [55, 64]. Another example is to use a small linear feedback shift register generator in base 2, with primitive characteristic polynomial [63, 66]. Point sets are implemented in this way in SSJ [60], for example.

6 Transforming the integrand

6.1 Change of variables

Improving the uniformity of the point set P_n is not the only way to reduce the error or variance. Another way is to change the function f to reduce its variability or its effective dimension without changing its mean. The primary technique for reducing the variability is a change of variable. Define a differentiable one-to-one function $\varphi : [0,1]^s \to [0,1]^s$, where $\varphi(\mathbf{v}) = (\varphi_1(\mathbf{v}), \ldots, \varphi_s(\mathbf{v}))$ for $\mathbf{v} = (v_1, \ldots, v_s) \in [0,1]^s$, and write

$$\mu = \int_{(0,1)^s} f(\mathbf{u})\, d\mathbf{u} = \int_{(0,1)^s} f\big(\varphi(\mathbf{v})\big) J(\mathbf{v})\, d\mathbf{v} = \int_{(0,1)^s} g(\mathbf{v})\, d\mathbf{v},$$

where $J(\mathbf{v})$ is the Jacobian of the transformation φ at $\mathbf{v}$, defined as the determinant of the $s \times s$ matrix whose element (i, j) is $\partial\varphi_i(\mathbf{u})/\partial u_j$. To estimate μ by MC or QMC or RQMC, we compute the function g (instead of f) at each point, and average. Note that this is exactly equivalent to applying a change of measure as in importance sampling; the uniform density of $\mathbf{u}$ is replaced by the density $1/J(\mathbf{v})$ obtained for $\mathbf{u} = \varphi(\mathbf{v})$ when $\mathbf{v}$ is uniform. The aim is to select φ so that g has smaller variation than f. Another reason for using a change of variables is to periodize the function (see Sect. 2.9 and [98]). Other standard variance-reduction techniques, such as control variate and conditional Monte Carlo, for example, can be applied to smooth out the integrand f before applying RQMC.

In theory, there is always a way of reducing the effective dimension to 1, as follows: Replace f by g where $g(\mathbf{u}) = g(u_1) = G^{-1}(u_1)$, where G is the distribution function of the random variable $f(\mathbf{U})$, i.e., $G(x) = \mathbb{P}[f(\mathbf{U}) \le x]$ where $\mathbf{U}$ is uniform over $(0,1)^s$. However, finding this g is usually much too difficult, except in a few special cases (for example, if $f(\mathbf{U})$ can be easily written as a function of a linear combination of normal random variables).

6.2 Bridge sampling and principal component sampling

In the context of financial applications, techniques for reducing the effective dimension have been proposed based on bridge sampling and principal component analysis [1, 22, 24, 43, 57, 76]. To illustrate these ideas, suppose that the integrand of interest can be written as a function of a multivariate normal vector $\mathbf{Y} = (Y_1, \ldots, Y_s)$, with mean zero and covariance matrix $\boldsymbol{\Sigma}$. That is, $\mu = \mathbb{E}[g(\mathbf{Y})]$ for some computable function g, and $g(\mathbf{Y})$ is the estimator. For example, $g(\mathbf{Y})$ can be the payoff of a financial option that depends on the sample path of a multivariate geometric Brownian motion observed at a finite set of epochs. There are many ways of generating the normal vector $\mathbf{Y}$ in this setting. The usual approach is to decompose $\boldsymbol{\Sigma} = \mathbf{A}\mathbf{A}^t$ for some matrix $\mathbf{A}$, generate a vector $\mathbf{Z} = (Z_1, \ldots, Z_s)^t$ where the Z_j are independent standard normal random variables, and return $\mathbf{Y} = \mathbf{A}\mathbf{Z}$. The Z_j are easily generated via $Z_j = \Phi^{-1}(U_j)$ where Φ is the standard normal distribution function and the U_j are independent uniform random variables over $(0,1)$. A fast and accurate approximation method for Φ^{-1}, based on rational Chebyshev approximation, is available in [60], for example.

Now, there are many possibilities for the choice of $\mathbf{A}$. The most common method, the *Cholesky factorization*, takes $\mathbf{A}$ to be lower triangular. A second possibility is an eigen-decomposition, for which $\mathbf{A} = \mathbf{P}\mathbf{D}^{1/2}$ where $\mathbf{D}$ is a diagonal matrix that contains the eigenvalues of $\boldsymbol{\Sigma}$ in decreasing order and $\mathbf{P}$ is an orthogonal matrix whose columns are the corresponding unit-length eigenvectors. This is the decomposition used in standard *principal component analysis* (PCA), and was proposed in [1] to reduce the effective dimension in the truncation sense, in the context of simulating a geometric Brownian motion for option pricing via QMC. It selects the matrix $\mathbf{A}$ so that the maximum amount of variance of $\mathbf{Y}$ comes from Z_1, then the maximum amount of variance conditional on Z_1 comes from Z_2, and so on. Thus, the method concentrates the variance in the first coordinates of $\mathbf{Z}$ as much as possible. If the Z_j are generated by inversion as $Z_i = \Phi^{-1}(U_j)$, then this method minimizes the effective dimension in the truncation sense if we consider the variance of $\mathbf{Y}$.

On the other hand, the PCA technique does not take into account the function g. It may turn out that with the PCA sampling scheme, $g(\mathbf{Y})$ depends very little on Z_1 and very much on Z_{25}, for example, even if Z_1 has more influence on the variance of $\mathbf{Y}$. In such a situation, PCA will miss its target. Ideally, one would like to find a decomposition $\mathbf{A}\mathbf{A}^{\mathrm{t}}$ that minimizes the effective dimension of the integrand $f(\mathbf{U}) = g(\mathbf{Y})$ (in some sense), which depends on g. For example, the goal could be to maximize the fraction of $\mathrm{Var}[g(\mathbf{Y})]$ that comes from Z_1, then maximize the fraction that comes from Z_2 given Z_1, and so on. For non-linear functions g, this is a difficult problem. Imai and Tan [41–43] propose to use a linear approximation $\tilde{g}$ of g, obtained via a first-order Taylor expansion around some "representative" point in the unit cube, to compute each new column of $\mathbf{A}$ so that the corresponding Z_j accounts for the maximal amount of residual variance of the linear approximation. A distinct representative point must be selected for each new column of $\mathbf{A}$. The main problem, however, is to find a good linear approximation. This can be difficult (and impractical) in general (e.g., if g is highly non-linear).

In financial applications, $\mathbf{Y}$ frequently corresponds to the observations of a c-dimensional Brownian process $\{\mathbf{W}(t) = (W_1(t), \ldots, W_c(t)),\ t \geq 0\}$ at times $0 = t_0 < t_1 < \cdots < t_d = T$. Then we have $s = cd$ and

$$\mathbf{Y} = \big(W_1(t_1), \ldots, W_c(t_1), \ldots, W_1(t_d), \ldots, W_c(t_d)\big)^{\mathrm{t}}.$$

For example, we may have a basket of c financial assets whose values evolve as (potentially correlated) geometric Brownian motions (GBMs), and the net payoff at time T is a function g of the c asset values at the fixed observation times.

The standard approach for simulating the Brownian process $\{\mathbf{W}(t),\ t \geq 0\}$ at times $t_1, \ldots, t_d$ is to generate (independent) increments $\mathbf{W}(t_1) - \mathbf{W}(t_0), \ldots, \mathbf{W}(t_d) - \mathbf{W}(t_{d-1})$ sequentially, in that order. We call it *sequential* or *random walk* sampling.

Another way is to write the covariance matrix $\boldsymbol{\Sigma}$ of the s-dimensional vector $\mathbf{Y}$ and decompose it as $\boldsymbol{\Sigma} = \mathbf{A}\mathbf{A}^{\mathrm{t}}$ as we saw earlier. One can take advantage of the fact that $\boldsymbol{\Sigma}$ can be written as a Kronecker product in this case, and use this to speed up the computations, especially for PCA [24]. If $c = 1$ and the decomposition is done with Cholesky, this is equivalent to sequential sampling.

Brownian Bridge sampling (BBS) was proposed in [76] as a tool to reduce the effective dimension for QMC, for this situation (with $c = 1$), by concentrating the variance (or importance) to the first few random numbers. See also [9, 24]. For notational simplicity, we assume that d is a power of two, but the method applies more generally. The idea is to first generate the vector $\mathbf{W}(t_d) = (W_1(t_d), \ldots, W_c(t_d))^{\mathrm{t}}$ from the appropriate c-dimensional normal distribution. Then, we generate $\mathbf{W}(t_{d/2})$ conditional on $(\mathbf{W}(0), \mathbf{W}(t_d))$, then, $\mathbf{W}(t_{d/4})$ conditional on $(\mathbf{W}(0), \mathbf{W}(t_{d/2}))$, and $\mathbf{W}(t_{3d/4})$ conditional on $(\mathbf{W}(t_{d/2}), \mathbf{W}(t_d))$, and so on, until the whole vector $\mathbf{Y}$ has been generated. In general, given $t_a < t < t_b$, the distribution of $\mathbf{W}(t)$ conditional on $(\mathbf{W}(t_a), \mathbf{W}(t_b))$ is multivariate normal with mean and covariance matrix

$$\mathbb{E}\big[\mathbf{W}(t)\big|\mathbf{W}(t_a) = \mathbf{x}, \mathbf{W}(t_b) = \mathbf{y}\big] = \mathbf{W}(t_a) + \big(\mathbf{W}(t_b) - \mathbf{W}(t_a)\big)\frac{t - t_a}{t_b - t_a},$$

$$\mathrm{Var}\big[\mathbf{W}(t)\big|\mathbf{W}(t_a) = \mathbf{x}, \mathbf{W}(t_b) = \mathbf{y}\big] = \frac{(t - t_a)(t_b - t)}{t_b - t_a}\boldsymbol{\Sigma},$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the drift vector and the covariance matrix of $\mathbf{W}$, i.e., $\mathbb{E}[\mathbf{W}(t) - \mathbf{W}(0)] = t\boldsymbol{\mu}$ and $\mathrm{Var}[\mathbf{W}(t) - \mathbf{W}(0)] = t\boldsymbol{\Sigma}$. When $c > 1$, we must again decompose the $(c \times c)$ conditional covariance matrix at each step to generate $\mathbf{W}(t)$ from the conditional distribution. These decompositions can be computed beforehand, either via Cholesky, or PCA, or another method. When the observation points are equally spaced, many of these covariance matrices are the same. Intuitively, BBS reduces the effective dimension in the truncation sense, because the first few random numbers already sketch the general shape of the trajectory, whereas the last ones are only making minor adjustment to it. It is just another (implicit) way of decomposing the matrix $\boldsymbol{\Sigma}$. Like for PCA, the overall impact of BBS on an estimator depends on the function g, and one can construct instances of g for which it increases the effective dimension and the RQMC variance [93, 113].

In one generalization of BBS, one can sample $\mathbf{W}(t_1), \ldots, \mathbf{W}(t_d)$ in an arbitrary order, i.e., according to any given permutation of $t_1, \ldots, t_d$. Lin and Wang [70] show that sampling first at the t_j closest to $3t_d/4$ maximizes the variance explained by Z_1. This rule applies recursively: If $t_\ell < t_d$ is the farthest point already sampled, then the next best point to sample in $\{t_{\ell+1}, \ldots, t_d\}$ is the one nearest $t_\ell + 3(t_d - t_\ell)/4$. If $t_j < t_\ell$ have been sampled but no other point in between, the next best point to sample in this interval (if any) is the one nearest $(t_\ell - t_j)/2$. So in the case of equidistant observation times, the optimal permutation differs from BBS essentially only by not starting immediately at t_d when sampling to the right of the rightmost point already sampled. In empirical experiments, the new permutation improves by a few percent the variance explained by Z_1, and its impact on the other projections is very small.

Wang [107] examines the effect of BBS and PCA sampling on the effective dimension in the superposition and truncation senses, in the context of Asian option pricing models based on the geometric average, zero strike price, and a geometric Brownian motion. He finds explicit formulas for the proportion of variance explained by various subsets of projections, and shows that if T is fixed while $\Delta = t_{j+1} - t_j = T/d \to 0$, the effective dimension remains bounded. This does not hold, however, if Δ is fixed and $T = d\Delta \to \infty$. This model is certainly much simpler than the ones for which we want to use MC or QMC, but it gives some idea of the potential effective dimension

for more elaborate models. Numerical experiments suggest that adding a non-zero strike price and replacing the geometric average by an arithmetic one does not change the behavior of the effective dimension significantly (this may not be true if the strike price is very large, so a positive payoff becomes a rare event, or if the time horizon is long, in which case the geometric and arithmetic averages may differ much more). In representative numerical examples with $d = 64$ and $d = 256$, Wang [107] observed that the variance ratio explained by the first two coordinates Z_1 and Z_2 (the truncation variance ratio of order 2) was very small for standard sequential sampling, over 90% for BBS, and more than 99% with PCA. On the other hand, the variance ratio explained by the one- and two-dimensional projections (the superposition variance ratio of order 2) was more than 99.99% already with the sequential sampling, and even more with BBS and PCA. This means that for this type of application, P_n could be constructed based on a discrepancy with large weights γ_{u} for $|\mathrm{u}| \le 2$, and small (or zero) weight to the higher-dimensional projections. Wang and Sloan [110–113] define and discuss such discrepancies, show how to construct shifted lattice rules with small discrepancy, and obtain bounds of the convergence rate of the error when these rules are used with BBS or PCA.

6.3 Extensions

The BBS methodology applies not only to Brownian processes, but to any Lévy process (a process with stationary and independent increments). However, we need an efficient algorithm to sample from the conditional distribution. We know how to do that for the Poisson and gamma processes, for example [2, 22]. In [2], the technique is applied to the simulation of a variance gamma process, where the underlying gamma processes are simulated by bridge sampling.

PCA sampling can also be applied to generate a Lévy process at times $0 = t_0 < t_1 < \cdots < t_d = T$, in the following way. Suppose the increment over the time interval $(t_{j-1}, t_j]$ has distribution function G_j. We can first generate a standard Brownian motion W at times $0 = \tau_0 < \tau_1 < \cdots < \tau_d$ via PCA, then return $G_j^{-1}(\Phi([W(\tau_j) - W(\tau_{j-1})]/\sqrt{\tau_j - \tau_{j-1}}))$ as the increment of the Lévy process over $(t_{j-1}, t_j]$. The times $\tau_1 < \cdots < \tau_d$ are free parameters that can be selected in some optimal way (to try minimizing the variance of the RQMC estimator).

7 Software

Software implementations of QMC point sets and sequences can be found in [8, 39, 45, 60, 61, 67] and the references given there.

Modern simulation software often provides multiple streams and substreams of random numbers, and facilities to create new streams and to rewind a stream to its starting point, or to the beginning of the current substream, or the next substream [59–61]. These tools facilitate the implementation of variance-reduction methods.

Ideally, one should be able to replace the streams of random numbers by QMC or RQMC point sets or sequences, with no internal modification to the simulation program. This is the philosophy behind the implementation found in SSJ [60, 61],

which offers several types of point sets and sequences. Some are infinite-dimensional. Available constructions include lattice rules, Hammersley point sets and Halton sequences, Sobol', Faure, and Niederreiter sequences, recurrence-based digital nets, and digital nets constructed by other techniques. Randomizations provided include a random shift, a digital random shift, and several matrix scrambles. Randomizations and transformations (such as the baker's transformation, for example) are implemented via container classes that act as filters. For example, to obtain a randomly shifted lattice with a baker's transformation, one can create a point set P for the underlying integration lattice, then a randomly shifted point set P' that contains P, then a baker transformed point set P'' that contains P', and P'' is the desired point set. To enumerate the points of a point set, and the successive coordinates of a point, one uses an *iterator* similar to iterators used to enumerate the elements of lists and other types of collections in Java. For randomized or transformed point sets, the randomizations and transformations are applied automatically when the points and coordinates are enumerated. An important feature of those iterators is that they are interchangeable with the random number streams (they have the same interface). This means that wherever a simulation program needs a stream of random numbers, we can simply provide an iterator to an RQMC point set instead, and this will replace MC by RQMC. For examples of this, see [60, 61].

8 Examples

The following examples are similar to those in [57]. We consider a vector of c GBMs, $\{S_i(t),\ t \geq 0\}$, $1 \leq i \leq c$, where S_i has *drift parameter* r and *volatility parameter* σ_i. That is,

$$S_i(t) = S_i(0)\exp\bigl[\bigl(r - \sigma_i^2/2\bigr)t + \sigma_i B_i(t)\bigr] = S_i(0)\exp\bigl[X_i(t)\bigr]$$

where $X_i(t) = (r - \sigma_i^2/2)t + \sigma_i B_i(t)$ and B_i is a standard Brownian motion. We assume that the B_i are correlated as follows:

$$\mathrm{Cov}\bigl[B_i(t+\delta) - B_i(t),\, B_j(t+\delta) - B_j(t)\bigr] = \rho_{i,j}\delta$$

for all $\delta > 0$. We have an option whose discounted payoff is $e^{-rT}\max[\bar{S} - K,\, 0]$, where

$$\bar{S} = \sum_{i=1}^{c}\sum_{j=1}^{d} w_{i,j} S_i(t_j) \tag{8.1}$$

(a weighted arithmetic average), for fixed observation times $0 < t_1 < \cdots < t_d = T$. In our numerical examples, we take $t_j = jT/d$, and $w_{i,j} = 1/(cd)$ unless indicated otherwise. Denoting

$$\mathbf{Y} = \bigl(X_1(t_1), \ldots, X_c(t_1), X_1(t_2), \ldots, X_c(t_2), \ldots, X_1(t_d), \ldots, X_c(t_d)\bigr)^{\mathrm{t}},$$

the element $((i-1)c + j), (i'-1)c + j')$ of $\boldsymbol{\Sigma}$ is $\rho_{i,i'}\sigma_i\sigma_{i'}\min(t_j, t_{j'})$. In this context, we can use the payoff based on the geometric average $\prod_{i=1}^{c}\prod_{j=1}^{d} S_i(t_j)^{w_{i,j}}$ in place of $\bar{S}$ as a control variate (CV) to reduce the variance. The expectation of this payoff is known exactly [24].

We use the following point sets:

(a) Sobol' nets with a random digital shift only (Sob-S),
(b) Sobol' nets with a left matrix scramble followed by a digital shift (Sob-LMS-S),
(c) Korobov lattice rules with a random shift modulo 1 (Kor-S), and
(d) Korobov lattice rules with a random shift modulo 1 followed by a baker transformation (Kor-S-B).

The primitive polynomials and the direction numbers for the Sobol' sequence were taken from [67]. The lattice rule parameters are from [64]. These point sets are rather standard; better ones could certainly be constructed based on discrepancies adapted to the specific examples, or to a larger class of target applications that contains these examples. Constructing such better point sets and making them available is on our agenda. All non-uniform random variables (mostly normal) were generated by inversion.

The *variance reduction factor* (VRF) is defined as the Monte Carlo variance (per observation) divided by n times the variance of $\hat{\mu}_{n,\text{rqmc}}$ for the randomized QMC method. The RQMC variance was estimated by making $m = 100$ independent replications of the randomization. These VRFs are noisy, with a standard error of about 20 percent or more. The simulations were written in Java using SSJ [60].

Example 1 For our first numerical illustration, we take $c = 10$ independent assets with a single observation time ($d = 1$), and the following parameters: $\rho_{i,j} = 0.4$ for $i \neq j$, $T = 1$, $\sigma_i = 0.5$, $r = 0.05$, $S_i(0) = 100$, and $K = 100$. We consider the payoff based on the arithmetic average (8.1). The exact value and the MC variance per observation are $\mu \approx 15.77$ and $\sigma^2 \approx 674$.

Table 1 gives the empirical variance reduction factors for the selected point sets. We compare two ways of sampling the vector $\mathbf{Y}$ by transforming a 10-dimensional vector of independent standard normals: the usual Cholesky factorization (left number in each table entry) and PCA (right number in each table entry). PCA definitely outperforms the Cholesky factorization, and the combination of PCA with randomized Sobol' nets gives the largest VRFs. As expected, the VRFs (i.e., efficiency gains) increase with n. For the Korobov rules, the baker transformation helps significantly, but the Sobol' nets are doing even better. The left matrix scramble also brings some

Table 1 Variance-reduction factors for Example 1, for Cholesky (left number) and PCA (right number)

	Sobol' nets					
	$n = 2^{14}$		$n = 2^{16}$		$n = 2^{18}$	
Sob+S	289	882	508	3567	1033	10299
Sob+LMS+S	381	4931	491	11452	593	39831
	Korobov lattice rules					
	$n = 16381, a = 5693$		$n = 65521, a = 944$		$n = 262139, a = 21876$	
Kor+S	106	737	30	1614	193	4218
Kor+S+B	185	6820	217	6864	684	20984

Table 2 Variance-reduction factors for Example 2 (250 dimensions)with Cholesky (left) and PCA (right)

	Sobol' nets					
	$n = 2^{14}$		$n = 2^{16}$		$n = 2^{18}$	
Sob+S	10	1299	17	3184	32	6046
Sob+LMS+S	6	4232	4	9219	35	16557
	Korobov lattice rules					
	$n = 16381, a = 5693$		$n = 65521, a = 944$		$n = 262139, a = 21876$	
Kor+S	18	878	18	1504	9	2643
Kor+S+B	50	4553	46	3657	43	7553

Table 3 Estimates of μ, σ^2, and the VRF $\sigma^2/\sigma_{\text{cv}}^2$, for Example 3

d	D_1	K	μ	σ^2	VRF
10	111	90	13.008	105	1.53×10^6
10	111	100	5.863	61	1.07×10^6
10	12	90	11.367	46	5400
10	12	100	3.617	23	3950
120	1	90	11.207	41	5050
120	1	100	3.367	20	4100

variance improvements. All methods require approximately the same CPU time for a given value of n.

Example 2 We modify an example from [41]: we take $c = 10$, $d = 25$, $\rho_{i,j} = 0.4$ for all $i \neq j$, $T = 1$, $r = 0.04$, $\sigma_i = 0.1 + 0.4(i-1)/9$ for all i, $S_i(0) = 100$, and $K = 100$. This gives a 250-dimensional integration problem. The exact value and the MC variance are $\mu \approx 5.818$ and $\sigma^2 \approx 72.3$ (these values are accurate up to the given digits).

The results are in Table 2, in the same format as for Table 1. They are similar. The main difference is that here we have a 250-dimensional problem instead of a 10-dimensional one, so PCA has more room to reduce the effective dimension compared with Cholesky. The VRFs are smaller than in Table 1 with Cholesky, but the improvement provided by PCA over Cholesky is larger.

Example 3 Here we consider an Asian option on a single asset ($c = 1$) whose price follows a GBM process. The payoff is based on the arithmetic average (8.1). We also experiment with the geometric average as a CV to reduce the variance. We examine the improvement of RQMC over MC with and without the CV, with sequential sampling (SEQ), BBS, and PCA, for an example with $S(0) = 100$, $r = \ln(1.09)$, $\sigma_i = 0.2$, $T = 120/365$, $t_j = D_1/365 + (T - D_1/365)(j-1)/(d-1)$ for $j = 1, \ldots, d$, for six combinations of values of (D_1, d, K) given in Table 3. This table provides estimates of the exact value μ, the MC variance σ^2 without the CV, and the VRF $\sigma^2/\sigma_{\text{cv}}^2$, where σ_{cv}^2 is the MC variance with the CV. These values are accurate at least to the

Table 4 VRFs for Example 3 with and without CV, for sequential sampling (SEQ), Brownian bridge sampling (BBS), and PCA sampling. The Sobol' point sets with a random digital shift (Sob-DS) have $2^{16} = 65536$ points, and the Korobov rules with a random shift (Kor-S) and with a random shift followed by a baker's transformation (Kor-SB) have $n = 65521$ and $a = 944$

d	D_1	K	P_n	without CV			with CV		
				SEQ	BBS	PCA	SEQ	BBS	PCA
10	111	90	Sob+DS	9572	12549	14279	63	183	4436
10	111	90	Kor+S	5943	6014	13751	18	29	291
10	111	90	Kor+S+B	88927	256355	563665	90	177	668
10	111	100	Sob+DS	5764	6638	10309	42	82	1913
10	111	100	Kor+S	2224	3682	8782	12	31	397
10	111	100	Kor+S+B	27214	29042	313724	29	61	635
10	12	90	Sob+DS	2205	9053	12175	27	67	434
10	12	90	Kor+S	442	1720	13790	13	50	71
10	12	90	Kor+S+B	1394	26883	446423	31	66	200
10	12	100	Sob+DS	368	2025	9506	21	42	274
10	12	100	Kor+S	63	909	5039	8	26	47
10	12	100	Kor+S+B	133	1317	123650	18	54	119
120	1	90	Sob+DS	325	7079	15101	3	48	483
120	1	90	Kor+S	192	2025	984	5	47	75
120	1	90	Kor+S+B	394	15575	474314	13	55	280
120	1	100	Sob+DS	39	1776	10244	3	48	217
120	1	100	Kor+S	24	672	5538	3	23	29
120	1	100	Kor+S+B	29	1101	162531	9	29	144

digit given. We immediately see that the CV alone (without RQMC) can reduce the variance by a huge factor, especially when d is small and the observation times are close to each other. This is because the geometric and arithmetic averages are almost the same in this case.

Table 4 gives the VRFs of RQMC over MC, with and without the CV, with approximately $n = 2^{16}$ points. It is important to recall that the optimal CV coefficient depends on the RQMC point set and on the sampling method, because it depends on the estimator's variance and its covariance with the CV, which may vary significantly across the methods [36]. In our experiments, these variances and covariances were estimated from the same simulation runs used to compute the estimators of μ.

Without the CV, RQMC reduces the variance by a huge factor, especially when combined with BBS or PCA. The Korobov rule with the random shift and the baker's transformation provides the largest variance reduction. With the CV, significant *additional* VRFs are obtained by the RQMC methods on top of those obtained by the CV alone. In this case, the Sobol' net with a random digital shift is the best performer. As an illustration, in the first row of Table 4, for PCA, the additional VRF over MC+CV is around 4436, whereas the CV alone was already providing a VRF of around 1.53×10^6. The combined VRF with both methods is approximately 6.8×10^9. The CPU times per run are about 20% larger with PCA in this case (in our

Table 5 Variance reduction factors for Example 4 with BGSS (left), BGBS (middle), and DGBS (right)

	Sobol' nets								
	$n = 2^{14}$			$n = 2^{16}$			$n = 2^{18}$		
Sob+S	37	359	585	41	421	1077	75	510	1154
Sob+LMS+S	29	530	557	49	565	995	77	735	1642
	Korobov lattice rules								
	$n = 16381, a = 5693$			$n = 65521, a = 944$			$n = 262139, a = 21876$		
Kor+S	17	54	119	24	138	263	22	285	557
Kor+S+B	52	53	57	44	44	433	92	93	1688

implementation), so plain (naive) MC would take about 5.6×10^9 times more CPU time to yield an estimator with equivalent precision. For $d = 120$, the CPU time for PCA sampling is about three times that of SEQ. With SEQ, our implementation needs about 2.7 seconds to make one million simulation runs and compute the estimators with and without CV for $d = 10$, and about 29 seconds for $d = 120$. These timings are for an AMD Athlon 64-bit processor running at 2.4 GHz.

Example 4 An Asian option under a variance gamma process. We consider now an asset price that evolves according to a *variance gamma* (VG) process S defined by [2, 3, 72]

$$S(t) = S(0)\exp\bigl[rt + X\bigl(G(t; 1, \nu), \theta, \sigma\bigr) + \omega t\bigr],$$

where X is a *Brownian process* with drift and variance parameters θ and σ, G is a *gamma process* (a process with independent gamma increments) with mean and variance parameters 1 and ν, X and G are independent, and $\omega = \ln(1 - \theta\nu - \sigma^2\nu/2)/\nu$. We want to estimate by simulation the value of an *Asian call option*, given by $\mathbb{E}[e^{-rT}\max(\bar{S} - K, 0)]$.

Here, the vector $(S(t_1), \ldots, S(t_d))$ is not multinormal, so the general setting of the previous subsection does not apply. However, the processes G and X (and therefore S) can be generated by sequential sampling (BGSS) or Brownian and gamma bridge sampling (BGBS), as explained in [2, 3]. For BGBS, we use the fact that for any given values $t_a < t < t_b$ and $\tau_a < \tau < \tau_b$, the distribution of $G(t)$ conditional on $(G(t_a), G(t_b))$ is beta with known parameters, and the distribution of $X(\tau)$ conditional on $(X(\tau_a), X(\tau_b))$ is normal with known parameters. This method requires the generation of one gamma variate, $d - 1$ beta variates, and d normal variates. Yet another method, explained in [2, 3], is *difference of gamma bridges sampling* (DGBS). It writes the process $\{S(t), t \geq 0\}$ as a difference of two gamma processes, and requires two gamma variates and $2d - 2$ beta variates.

For a numerical illustration, we take the following parameters from [3]: $\theta = -0.1436$, $\sigma = 0.12136$, $\nu = 0.3$, $r = 0.1$, $T = 1$, $d = 16$, $t_j = j/16$, $K = 101$, and $S(0) = 100$. The exact value and the MC variance are $\mu \approx 5.725$ and $\sigma^2 \approx 29.89$. Table 5 gives the variance reduction factors of QMC compared with MC. DGBS provides the best improvement.

Example 5 Boyle et al. [7] consider a *spread option*, where $d = 1$, $c = 2$, and the payoff is $e^{-rT} \max[S_2(T) - S_1(T) - K, 0]$, which is again a function of a bivariate normal with known covariance matrix $\boldsymbol{\Sigma}$. To generate the payoff, they use importance sampling as follows: Generate $S_1(T)$ from its original distribution, then generate $S_2(T)$ from its conditional distribution given that the payoff is non-zero, and multiply the estimator by the appropriate likelihood ratio. This reduces the variability of the integrand and makes it smoother. For RQMC, they use a two-dimensional lattice rule, and they periodize the function with polynomial and sine transformations. Their best results are with the transformations $\varphi_{\text{poly},4}(u) = u^4(35 - 84u + 70u^2 - 20u^3)$ and $\varphi_{\sin,3}(u) = (12\pi u - 8\sin(2\pi u) + \sin(4\pi u))/12\pi$, which are special cases of well-known classes of transformations. We ran some experiments to compare their proposed transformation with the baker's transformation, which also periodizes the function, and found that $\varphi_{\text{poly},4}$ and $\varphi_{\sin,3}$ gave much larger variance reductions than the baker's transformation, for $n \approx 2^{16}$ and with the same lattices, for this two-dimensional example. We also tried an Asian option with $d = 2$, $t_1 = 1/2$, $t_2 = 1$, $S(0) = 100$, $K = 90$, $r = \ln(1.09)$, and $\sigma_1 = 0.2$, with sequential sampling combined with importance sampling, and the proposed transformations did slightly better than the baker's transformation (approximately by a factor of 2).

However, for higher-dimensional problems, we observed the opposite: these transformations give much larger variance than the baker's transformation. For Example 3 with $d = 10$, $K = 90$, and $n \approx 2^{16}$, for instance, with sequential sampling, the higher-order transformations gave a larger variance than plain Monte Carlo. In other words, they annihilate all the RQMC gain. The explanation is that the higher-order transformations also increase the variation of the function, and the impact of this higher variation increases with s.

9 Conclusion

Our discussion of QMC in this paper was in the context of estimating a mathematical expectation. But QMC can also be used advantageously to estimate something else than an expectation: e.g., for estimating a quantile, or a function of several expectations, or the gradient of an expectation with respect to a vector of parameters [7, 24]. It can also be used to obtain an approximation of a function f over a given domain (see [51] and other references cited there), or to estimate the solution of an optimization problem in which the objective function or the constraints (or both) involve mathematical expectations. This can be used effectively in the context of computing maximum likelihood estimators, for example. QMC can also replace MC in algorithms that combine MC with approximate dynamic programming (e.g., for pricing American-style options) [10, 24]. All these settings have applications in finance.

Other QMC developments that could be of high interest in finance are special methods designed for the simulation of Markov chains over many steps, a setting for which it is difficult to reduce the effective dimension to a small number. A recently developed RQMC method named *array-RQMC* [52, 62] simulates n copies of the chain, advancing all copies by one step using an RQMC point set at each iteration, and induces negative dependence between these copies, so that the empirical distribution of the n states at any given step provides a better estimate of the true distribution than if the n copies were simulated independently. This method provides variance

reduction factors of over 1000 in some examples where the chain evolves over a few hundred steps.

Important topics for ongoing and future research include the development of additional effective methods for reducing the effective dimension, to better understand the ANOVA decomposition in typical finance problems, to better understand the different discrepancies and the impact of their choice in the final variance of the estimators for typical classes of integrands, and to develop software tools that can easily provide appropriate point sets tailored to specific classes of applications. In the short term, classical QMC point set constructions available in popular software should be replaced by constructions whose parameters are selected on the basis of criteria (discrepancies) that take better account of the low-dimensional projections. This was already pointed out in [64], for example.

Acknowledgements This work has been supported by an NSERC, Canada Discovery Grant and a Canada Research Chair to the author. Fred J. Hickernell, Art B. Owen, Richard Simard, and Vasile Sinescu provided suggestions and corrected errors in earlier drafts. Boni Abdel Chabi-Yo, Jean-Sébastien Parent-Chartier, Clément Teule, and Pierre-Alexandre Tremblay helped writing the programs for the numerical experiments.

References

1. Acworth, P., Broadie, M., Glasserman, P.: A comparison of some Monte Carlo and quasi-Monte Carlo techniques for option pricing. In: Hellekalek, P., Larcher, G., Niederreiter, H., Zinterhof, P. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 1996. Lecture Notes in Statistics, vol. 127, pp. 1–18. Springer, New York (1998)
2. Avramidis, T., L'Ecuyer, P.: Efficient Monte Carlo and quasi-Monte Carlo option pricing under the variance-gamma model. Manag. Sci. **52**, 1930–1944 (2006)
3. Avramidis, T., L'Ecuyer, P., Tremblay, P.A.: Efficient simulation of gamma and variance-gamma processes. In: Proceedings of the 2003 Winter Simulation Conference, pp. 319–326. IEEE Press, Piscataway (2003)
4. Ben-Ameur, H., L'Ecuyer, P., Lemieux, C.: Combination of general antithetic transformations and control variables. Math. Oper. Res. **29**, 946–960 (2004)
5. Bierbrauer, J., Edel, Y.: Construction of digital nets from BCH-codes. In: Hellekalek, P., Larcher, G., Niederreiter, H., Zinterhof, P. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 1996. Lecture Notes in Statistics, vol. 127, pp. 221–231. Springer, New York (1998)
6. Boyle, P.: Options: a Monte Carlo approach. J. Financ. Econ. **4**, 323–338 (1977)
7. Boyle, P., Lai, Y., Tan, K.S.: Pricing options using lattice rules. North Am. Actuar. J. **9**(3), 50–76 (2005)
8. Bratley, P., Fox, B.L., Niederreiter, H.: Implementation and tests of low-discrepancy sequences. ACM Trans. Model. Comput. Simul. **2**, 195–213 (1992)
9. Caflisch, R.E., Morokoff, W., Owen, A.: Valuation of mortgage-backed securities using Brownian bridges to reduce effective dimension. J. Comput. Finance **1**, 27–46 (1997)
10. Chaudhary, S.K.: American options and the LSM algorithm: quasi-random sequences and Brownian bridges. J. Comput. Finance **8**, 101–115 (2005)
11. Cools, R., Kuo, F.Y., Nuyens, D.: Constructing embedded lattice rules for multivariate integration. SIAM J. Sci. Comput. **28**, 2162–2188 (2006)
12. Cools, R., Nuyens, D.: A Belgian view on lattice rules. In: Keller, A., Heinrich, S., Niederreiter, H. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 3–21. Springer, Berlin (2008)

13. Cranley, R., Patterson, T.N.L.: Randomization of number theoretic methods for multiple integration. SIAM J. Numer. Anal. **13**, 904–914 (1976)
14. Cristea, L.L., Dick, J., Leobacher, G., Pillichshammer, F.: The tent transformation can improve the convergence rate of quasi-Monte Carlo algorithms using digital nets. Numer. Math. **105**, 413–455 (2007)
15. Dick, J.: Explicit constructions of quasi-Monte Carlo rules for the numerical integration of high-dimensional periodic functions (2008). http://www2.maths.unsw.edu.au/Contacts/profile.php?logname=josi
16. Dick, J.: Walsh spaces containing smooth functions and quasi-Monte Carlo rules of arbitrary high order (2008). http://www2.maths.unsw.edu.au/Contacts/profile.php?logname=josi
17. Dick, J., Sloan, I.H., Wang, X., Wozniakowski, H.: Liberating the weights. J. Complex. **20**, 593–623 (2004)
18. Dick, J., Sloan, I.H., Wang, X., Wozniakowski, H.: Good lattice rules in weighted Korobov spaces with general weights. Numer. Math. **103**, 63–97 (2006)
19. Edel, Y., L'Ecuyer, P.: A coding theoretic approach to building nets with well-equidistributed projections. In: Keller, A., Heinrich, S., Niederreiter, H. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 313–325. Springer, Berlin (2008)
20. Faure, H.: Discrépance des suites associées à un système de numération en dimension s. Acta Arith. **61**, 337–351 (1982)
21. Faure, H., Tezuka, S.: Another random scrambling of digital (t, s)-sequences. In: Fang, K.T., Hickernell, F.J., Niederreiter, H. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2000, pp. 242–256. Springer, Berlin (2002)
22. Fox, B.L.: Strategies for Quasi-Monte Carlo. Kluwer Academic, Boston (1999)
23. Friedel, I., Keller, A.: Fast generation of randomized low-discrepancy point sets. In: Fang, K.T., Hickernell, F.J., Niederreiter, H. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2000, pp. 257–273. Springer, Berlin (2001)
24. Glasserman, P.: Monte Carlo Methods in Financial Engineering. Springer, New York (2004)
25. Halton, J.H.: On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numer. Math. **2**, 84–90 (1960)
26. Hammersley, J.M.: Monte Carlo methods for solving multivariate problems. Ann. N. Y. Acad. Sci. **86**, 844–874 (1960)
27. Heinrich, S., Hickernell, F.J., Yue, R.X.: Optimal quadrature for Haar wavelet spaces. Math. Comput. **73**, 259–277 (2004)
28. Hellekalek, P.: On the assessment of random and quasi-random point sets. In: Hellekalek, P., Larcher, G. (eds.) Random and Quasi-Random Point Sets. Lecture Notes in Statistics, vol. 138, pp. 49–108. Springer, New York (1998)
29. Hickernell, F.J.: A generalized discrepancy and quadrature error bound. Math. Comput. **67**, 299–322 (1998)
30. Hickernell, F.J.: 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, pp. 109–166. Springer, New York (1998)
31. Hickernell, F.J.: What affects the accuracy of quasi-Monte Carlo quadrature?. In: Niederreiter, H., Spanier, J. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 1998, pp. 16–55. Springer, Berlin (2000)
32. Hickernell, F.J.: Obtaining $O(N^{-2+\epsilon})$ convergence for lattice quadrature rules. In: Fang, K.T., Hickernell, F.J., Niederreiter, H. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2000, pp. 274–289. Springer, Berlin (2002)
33. Hickernell, F.J.: My dream quadrature rule. J. Complex. **19**, 420–427 (2003)
34. Hickernell, F.J., Hong, H.S.: The asymptotic efficiency of randomized nets for quadrature. Math. Comput. **68**(226), 767–791 (1999)
35. Hickernell, F.J., Hong, H.S., L'Ecuyer, P., Lemieux, C.: Extensible lattice sequences for quasi-Monte Carlo quadrature. SIAM J. Sci. Comput. **22**, 1117–1138 (2001)
36. Hickernell, F.J., Lemieux, C., Owen, A.B.: Control variates for quasi-Monte Carlo. Stat. Sci. **20**, 1–31 (2005)
37. Hickernell, F.J., Sloan, I.H., Wasilkowski, G.W.: On strong tractability of weighted multivariate integration. Math. Comput. **73**(248), 1903–1911 (2004)
38. Hickernell, F.J., Wózniakowski, H.: The price of pessimism for multidimensional quadrature. J. Complex. **17**, 625–659 (2001)

39. Hong, H.S., Hickernell, F.H.: Algorithm 823: implementing scrambled digital sequences. ACM Trans. Math. Softw. **29**, 95–109 (2003)
40. Hua, L., Wang, Y.: Applications of Number Theory to Numerical Analysis. Springer, Berlin (1981)
41. Imai, J., Tan, K.S.: Enhanced quasi-Monte Carlo methods with dimension reduction. In: Yücesan, E., Chen, C.H., Snowdon, J.L., Charnes, J.M. (eds.) Proceedings of the 2002 Winter Simulation Conference, pp. 1502–1510. IEEE Press, Piscataway (2002)
42. Imai, J., Tan, K.S.: Minimizing effective dimension using linear transformation. In: Niederreiter, H. (ed.) Monte Carlo and Quasi-Monte Carlo Methods 2002, pp. 275–292. Springer, Berlin (2004)
43. Imai, J., Tan, K.S.: A general dimension reduction technique for derivative pricing. J. Comput. Finance **10**, 129–155 (2006)
44. Jäckel, P.: Monte Carlo Methods in Finance. Wiley, Chichester (2002)
45. Joe, S., Kuo, F.Y.: Remark on algorithm 659: Implementing Sobol's quasirandom sequence generator. ACM Trans. Math. Softw. **29**, 49–57 (2003)
46. Joe, S., Kuo, F.Y.: Constructing Sobol' sequences with better two-dimensional projections. SIAM J. Sci. Comput. **30**, 2635–2654 (2008)
47. Joe, S., Sloan, I.H.: Embedded lattice rules for multidimensional integration. SIAM J. Numer. Anal. **29**, 1119–1135 (1992)
48. Knuth, D.E.: The Art of Computer Programming, vol. 2: Seminumerical Algorithms, 3rd edn. Addison-Wesley, Reading (1998)
49. Kuo, F.Y.: Component-by-component constructions achieve the optimal rate of convergence for multivariate integration in weighted Korobov and Sobolev spaces. J. Complex. **19**, 301–320 (2003)
50. Kuo, F.Y., Sloan, I.H.: Lifting the curse of dimensionality. Not. Am. Math. Soc. **52**, 1320–1328 (2005)
51. Kuo, F.Y., Sloan, I.H., Woźniakowski, H.: Lattice rule algorithms for multivariate approximation in the average case setting. J. Complex. **24**, 283–323 (2008)
52. Lécot, C., Tuffin, B.: Quasi-Monte Carlo methods for estimating transient measures of discrete time Markov chains. In: Niederreiter, H. (ed.) Monte Carlo and Quasi-Monte Carlo Methods 2002, pp. 329–343. Springer, Berlin (2004)
53. L'Ecuyer, P.: Maximally equidistributed combined Tausworthe generators. Math. Comput. **65**(213), 203–213 (1996)
54. L'Ecuyer, P.: Good parameters and implementations for combined multiple recursive random number generators. Oper. Res. **47**, 159–164 (1999)
55. L'Ecuyer, P.: Tables of linear congruential generators of different sizes and good lattice structure. Math. Comput. **68**(225), 249–260 (1999)
56. L'Ecuyer, P.: Polynomial integration lattices. In: Niederreiter, H. (ed.) Monte Carlo and Quasi-Monte Carlo Methods 2002, pp. 73–98. Springer, Berlin (2004)
57. L'Ecuyer, P.: Quasi-Monte Carlo methods in finance. In: Ingalls, R.G., Rossetti, M.D., Smith, J.S., Peters, B.A. (eds.) Proceedings of the 2004 Winter Simulation Conference, pp. 1645–1655. IEEE Press, Piscataway (2004)
58. L'Ecuyer, P.: Uniform random number generation. In: Henderson, S.G., Nelson, B.L. (eds.) Simulation. Handbooks in Oper. Res. and Manag. Sci., pp. 55–81. Elsevier, Amsterdam (2006)
59. L'Ecuyer, P.: Pseudorandom number generators. In: Platen, E., Jaeckel, P. (eds.) Simulation Methods in Financial Engineering, Encyclopedia of Quantitative Finance. Wiley (2009, forthcoming). http://www.wiley.com//legacy/wileychi/eqf/index.html
60. L'Ecuyer, P.: SSJ: A Java Library for Stochastic Simulation (2008). Software user's guide. http://www.iro.umontreal.ca/~lecuyer
61. L'Ecuyer, P., Buist, E.: Simulation in Java with SSJ. In: Kuhl, M.E., Steiger, N.M., Armstrong, F.B., Joines, J.A. (eds.) Proceedings of the 2005 Winter Simulation Conference, pp. 611–620. IEEE Press, Piscataway (2005)
62. L'Ecuyer, P., Lécot, C., Tuffin, B.: A randomized quasi-Monte Carlo simulation method for Markov chains. Oper. Res. **56**, 958–975 (2008)
63. L'Ecuyer, P., Lemieux, C.: Quasi-Monte Carlo via linear shift-register sequences. In: Proceedings of the 1999 Winter Simulation Conference, pp. 632–639. IEEE Press, Piscataway (1999)
64. L'Ecuyer, P., Lemieux, C.: Variance reduction via lattice rules. Manag. Sci. **46**, 1214–1235 (2000)
65. L'Ecuyer, P., Lemieux, C.: Recent advances in randomized quasi-Monte Carlo methods. In: Dror, M., L'Ecuyer, P., Szidarovszky, F. (eds.) Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications, pp. 419–474. Kluwer Academic, Boston (2002)
66. Lemieux, C.: L'utilisation de règles de réseau en simulation comme technique de réduction de la variance. PhD thesis, Université de Montréal (2000)

67. Lemieux, C., Cieslak, M., Luttmer, K.: RandQMC User's Guide: A Package for Randomized Quasi-Monte Carlo Methods in C (2004). Software user's guide. http://www.math.uwaterloo.ca/~lemieux/randqmc.html
68. Lemieux, C., L'Ecuyer, P.: On selection criteria for lattice rules and other low-discrepancy point sets. Math. Comput. Simul. **55**(1–3), 139–148 (2001)
69. Lemieux, C., L'Ecuyer, P.: Randomized polynomial lattice rules for multivariate integration and simulation. SIAM J. Sci. Comput. **24**, 1768–1789 (2003)
70. Lin, J., Wang, X.: New Brownian bridge construction in quasi-Monte Carlo methods for computational finance. J. Complex. **24**, 109–133 (2008)
71. Liu, R., Owen, A.B.: Estimating mean dimensionality of analysis of variance decompositions. J. Am. Stat. Assoc. **101**(474), 712–721 (2006)
72. Madan, D.B., Carr, P.P., Chang, E.C.: The variance gamma process and option pricing. Eur. Finance Rev. **2**, 79–105 (1998)
73. Matoušek, J.: Geometric Discrepancy: An Illustrated Guide. Springer, Berlin (1999)
74. Morokoff, W.J.: Generating quasi-random paths for stochastic processes. SIAM Rev. **40**, 765–788 (1998)
75. Morokoff, W.J., Caflisch, R.E.: Quasi-random sequences and their discrepancies. SIAM J. Sci. Comput. **15**, 1251–1279 (1994)
76. Moskowitz, B., Caflisch, R.E.: Smoothness and dimension reduction in quasi-Monte Carlo methods. J. Math. Comput. Model. **23**, 37–54 (1996)
77. Niederreiter, H.: Point sets and sequences with small discrepancy. Monatshefte Math. **104**, 273–337 (1987)
78. Niederreiter, H.: Low-discrepancy point sets obtained by digital constructions over finite fields. Czechoslovak Math. J. **42**, 143–166 (1992)
79. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods. SIAM CBMS-NSF Regional Conference Series in Applied Mathematics. vol. 63, SIAM, Philadelphia (1992)
80. Niederreiter, H.: Digital nets and coding theory. In: Feng, K.Q., Niederreiter, H., Xing, C.P. (eds.) Coding, Cryptography and Combinatorics. Progress in Computer Science and Applied Logic, vol. 23, pp. 247–257. Birkhäuser, Basel (2004)
81. Niederreiter, H., Pirsic, G.: Duality for digital nets and its applications. Acta Arith. **97**, 173–182 (2001)
82. Niederreiter, H., Xing, C.: The algebraic-geometry approach to low-discrepancy sequences. In: Hellekalek, P., Larcher, G., Niederreiter, H., Zinterhof, P. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 1996. Lecture Notes in Statistics, vol. 127, pp. 139–160. Springer, New York (1998)
83. Nuyens, D., Cools, R.: Fast algorithms for component-by-component construction of rank-1 lattice rules in shift-invariant reproducing kernel Hilbert spaces. Math. Comput. Simul. **75**, 903–920 (2006)
84. Nuyens, D., Cools, R.: Fast component-by-component construction of rank-1 lattice rules with a non-prime number of points. J. Complex. **22**, 4–28 (2006)
85. Owen, A.B.: Randomly permuted (t, m, s)-nets and (t, s)-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, pp. 299–317. Springer, Berlin (1995)
86. Owen, A.B.: Monte Carlo variance of scrambled equidistribution quadrature. SIAM J. Numer. Anal. **34**, 1884–1910 (1997)
87. Owen, A.B.: Scrambled net variance for integrals of smooth functions. Ann. Stat. **25**, 1541–1562 (1997)
88. Owen, A.B.: Latin supercube sampling for very high-dimensional simulations. ACM Trans. Model. Comput. Simul. **8**, 71–102 (1998)
89. Owen, A.B.: Variance with alternative scramblings of digital nets. ACM Trans. Model. Comput. Simul. **13**, 363–378 (2003)
90. Owen, A.B.: Multidimensional variation for quasi-Monte Carlo. In: Fan, J., Li, G. (eds.) International Conference on Statistics in Honour of Professor Kai-Tai Fang's 65th Birthday, pp. 49–74 (2005). http://www-stat.stanford.edu/~owen/reports/
91. Owen, A.B.: Quasi-Monte Carlo for integrands with point singularities at unknown locations. In: Niederreiter, H., Talay, D. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2004, pp. 403–417. Springer, Berlin (2006)
92. Panneton, F., L'Ecuyer, P.: Infinite-dimensional highly-uniform point sets defined via linear recurrences in $\mathbb{F}_{2^w}$. In: Niederreiter, H., Talay, D. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2004, pp. 419–429. Springer, Berlin (2006)

93. Papageorgiou, A.: The Brownian bridge does not offer a consistent advantage in quasi-Monte Carlo integration. J. Complex. **18**, 171–186 (2002)
94. Paskov, S., Traub, J.: Faster valuation of financial derivatives. J. Portf. Manag. **22**, 113–120 (1995)
95. Schmid, W.C., Schürer, R.: MinT, the database for optimal (t, m, s)-net parameters (2005). http://mint.sbg.ac.at
96. Sinescu, V., Joe, S.: Good lattice rules based on the general weighted star discrepancy. Math. Comput. **76**(258), 989–1004 (2007)
97. Sloan, I.H.: QMC integration—beating intractability by weighting the coordinate directions. In: Fang, K.T., Hickernell, F.J., Niederreiter, H. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2000, pp. 103–123. Springer, Berlin (2002)
98. Sloan, I.H., Joe, S.: Lattice Methods for Multiple Integration. Clarendon, Oxford (1994)
99. Sloan, I.H., Kuo, F.Y., Joe, S.: On the step-by-step construction of quasi-Monte Carlo rules that achieve strong tractability error bounds in weighted Sobolev spaces. Math. Comput. **71**, 1609–1640 (2002)
100. Sloan, I.H., Wang, X., Wozniakowski, H.: Finite-order weights imply tractability of multivariate integration. J. Complex. **20**, 46–74 (2004)
101. Sloan, I.H., Woźniakowski, H.: When are quasi-Monte Carlo algorithms efficient for high-dimensional integrals? J. Complex. **14**, 1–33 (1998)
102. Sloan, I.H., Woźniakowski, H.: Tractability of integration in non-periodic and periodic weighted tensor product Hilbert spaces. J. Complex. **18**, 479–499 (2002)
103. Sobol', I.M.: The distribution of points in a cube and the approximate evaluation of integrals. U.S.S.R. Comput. Math. and Math. Phys. **7**, 86–112 (1967)
104. Traub, J.F., Wasilkowski, G.W., Woźniakowski, H.: Information-Based Complexity. Academic Press, New York (1988)
105. Tuffin, B.: On the use of low-discrepancy sequences in Monte Carlo methods. Monte Carlo Methods Appl. **2**, 295–320 (1996)
106. Wahba, G.: Spline Models for Observational Data. SIAM CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 59, SIAM, Philadelphia (1990)
107. Wang, X.: On the effects of dimension reduction techniques on some high-dimensional problems in finance. Oper. Res. **54**, 1063–1078 (2006)
108. Wang, X.: Constructing robust good lattice rules for computational finance. SIAM J. Sci. Comput. **29**, 598–621 (2007)
109. Wang, X., Hickernell, F.J.: Randomized Halton sequences. Math. Comput. Model. **32**, 887–899 (2000)
110. Wang, X., Sloan, I.H.: Why are high-dimensional finance problems often of low effective dimension?. SIAM J. Sci. Comput. **27**, 159–183 (2005)
111. Wang, X., Sloan, I.H.: Efficient weighted lattice rules with applications to finance. SIAM J. Sci. Comput. **28**, 728–750 (2006)
112. Wang, X., Sloan, I.H.: Brownian bridge and principal component analysis: toward removing the curse of dimensionality. IMA J. Numer. Anal. **27**, 631–654 (2007)
113. Wang, X., Sloan, I.H.: Low discrepancy sequences in high dimensions: How well are their projections distributed? J. Comput. Appl. Math. **213**, 366–386 (2008)
114. Wózniakowski, H.: Average case complexity of multivariate integration. Bull. Am. Math. Soc. **24**, 185–194 (1991)